77-FM-18
Vol. IX

JSC-12564

Flight Design System-1
System Design Document

Executive Logic Flow - Program Design Language

Mission Planning and Analysis Division

December 1979

NASA
National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston, Texas

77-FM-18
Vol. IX

JSC-12564

SHUTTLE PROGRAM

FLIGHT DESIGN SYSTEM-1
SYSTEM DESIGN DOCUMENT

EXECUTIVE LOGIC FLOW - PROGRAM DESIGN LANGUAGE

By Mission Analysis and Engineering
Federal Systems Division - Houston
IBM Corporation

JSC Task Monitor: Software Development Branch

Approved: Elric N. McHenry
Elric N. McHenry, Chief
Software Development Branch

Approved: Ronald L. Berry
Ronald L. Berry, Chief
Mission Planning and Analysis Division

Mission Planning and Analysis Division

National Aeronautics and Space Administration

Lyndon B. Johnson Space Center

Houston, Texas

December 1979

PREFACE

The Flight Design System-1 (FDS-1) is a pilot project to evaluate current concepts and to determine the hardware/software capability that will be required for the operational era to support Shuttle flight planning. This software system is being implemented on a Hewlett-Packard 21MX computer with a Daconics documentation system and will provide terminal-based interactive flight planning capability.

The System Design Document (SDD) for FDS-1 is the specification for and description of this hardware/software facility. The SDD is logically organized into 10 published volumes. This organization is presented in the accompanying table. The material in the early volumes is primarily presented from the user's point of view, whereas the latter material is software-developer oriented. The SDD will be published by volumes over a period of time, and various volumes will be updated and republished during the development of FDS-1.

FDS-1 SYSTEM DESIGN DOCUMENT

Volume I	Introduction, Overview, and User Interface
*Volume II	Utility Processor Library
*Volume III	Processor Library
Volume IV	System Architecture and Executive
Volume V	Data Management and Data Base Documentation Support System
Volume VI	Standards
Volume VII	Utility Support Software
Volume VIII	Build and Delivery Procedures, Software Development, Debug, and Sytem Build Aids
Volume IX	Executive Logic Flow - Program Design Language
Volume X	Document Change Request Procedure and Submittal Form

*Combined as one volume with title: Volume III FDS-1 Processor Library

CONTENTS

Section		Page
1.0	INTRODUCTION	1-1
2.0	COMMON FOR FDS EXECUTIVE	2-1
3.0	FDS EXECUTIVE MESSAGES	3-1
4.0	PDL LISTING PROGRAM	4-1
5.0	FDS EXECUTIVE DETAILED LOGIC FLOW	5-1
6.0	DETAILED LOGIC FLOW LISTING - PROGRAM EXECUTION	6-1

1.0 INTRODUCTION

The flight design system can be divided functionally into two major areas: the FDS Executive and the application processor. The requirements for and the architecture of the FDS Executive is presented in volume I and volume IV, respectively, of this document. Volume IX presents the detailed logic flow for the FDS Executive.

2.0 COMMON FOR FDS EXECUTIVE

Three commons (XE, XB, and XS) presented in this section are used throughout the FDS Executive.

```
      1 CD     COMMON XE(400), XB(1400), XS(200)                                                XECOMM
      1 CD                                                                                      XECOMM
      1 CD     XE      - EXECUTIVE FIXED COMMON (GLOBAL)                                        XECOMM
      1 CD     XB      - EXECUTIVE DYNAMIC BLOCK (SUBSTA LEVEL DEPENDENT)                       XECOMM
      1 CD     XS      - EXECUTIVE SCRATCH SPACE (VOLATILE ACCROSS ALL CALLS                    XECOMM
      1 CD               TO FDS ROUTINES EXCEPT XR___)                                        XECOMM
      1 CD                                                                                      XECOMM
      1 CD*********                                                                             XECOMM
      1 CD                                                                                      XECOMM
      1 CD     XE CONTENTS                                                                      XECOMM
      1 CD                                                                                      XECOMM
      1 CD     INTEGER                                                                          XECOMM
      1 CD     * CARTPG                                                                         XECOMM
      1 CD     *,CLASNO                        ,COMBUF(256)                                     XECOMM
      1 CD     *,COMPTR                        ,EXEND                                           XECOMM
      1 CD     *,FLAGS                         ,OLDIND                                          XECOMM
      1 CD     *,PRCNAM(3)                     ,QUAL                                            XECOMM
      1 CD     *,REQBUF(64)                    ,REQPTR                                          XECOMM
      1 CD     *,SEQEND                        ,SEQNAP(3)                                       XECOMM
      1 CD     *,SEQPTR                        ,SEQSTR                                          XECOMM
      1 CD     *,SUBSTA                        ,TABEND                                          XECOMM
      1 CD     *,TKNLNG                                                                         XECOMM
      1 CD     *,TOKENS(32)                    ,XB                                              XECOMM
      1 CD     *,XE                            ,XS                                              XECOMM
      1 CD                                                                                      XECOMM
      1 CD     DIMENSION                                                                        XECOMM
      1 CD     * INTNAM(3)                                                                      XECOMM
      1 CD                                                                                      XECOMM
      1 CD     EQUIVALENCE                                                                      XECOMM
      1 CD     * (XE(1)  ,LU)                  ,(XE(2)  ,CLASNO)                                XECOMM
      1 CD     *,(XE(3)  ,QUAL)                ,(XE(4)  ,FLAGS)                                 XECOMM
      1 CD     *,(XE(5)  ,MASSTA)              ,(XE(6)  ,SUBSTA)                                XECOMM
      1 CD     *,(XE(7)  ,SEQNAM(1))           ,(XE(10) ,SEQSTR)                                XECOMM
      1 CD     *,(XE(11) ,SEQEND)              ,(XE(12) ,SEQPTR)                                XECOMM
      1 CD     *,(XE(13) ,INTNAM(1))           ,(XE(16) ,PRCNAM(1))                             XECOMM
      1 CD     *,(XE(19) ,REQPTR)              ,(XE(20) ,REQBUF(1))                             XECOMM
      1 CD     *,(XE(84) ,TKNLNG)              ,(XE(85) ,TOKENS(1))                             XECOMM
      1 CD     *,(XE(139),EXEND)               ,(XE(140),TABEND)                                XECOMM
      1 CD     *,(XE(141),OLDIND)                                                               XECOMM
      1 CD     *,(XE(142),CARTRG)              ,(XE(143),NOPROC)                                XECOMM
      1 CD     *,(XE(144),COMPTR)              ,(XE(145),COMBUF(1))                             XECOMM
      1 CD                                                                                      XECOMM
      1 CD     CARTRG - NUMBER OF THE DISK CARTRIDGE CONTAINING EXECUTIVE MASTER                XECOMM
      1 CD              FILES                                                                   XECOMM
      1 CD     CLASNO - EXECUTIVE/MANAGER REQUEST BLOCK CLASS I/O NUMBER                        XECOMM
      1 CD     COMBUF - TERMINAL COMMUNICATIONS OUTPUT BUFFER                                   XECOMM
      1 CD              (1)     -  NUMBER OF TOKENS IN BUFFER                                   XECOMM
      1 CD              (2)     -  NUMBER OF USED WORDS IN BUFFER                               XECOMM
      1 CD              (3-256) -  TOKENS REPRESENTING USER'S RESPONSE                          XECOMM
      1 CD     COMPTR - POINTER TO TOKEN CURRENTLY BEING PROCESSED FROM COMBUF                  XECOMM
      1 CD     EXEND  - SEQUENCE # WHERE EXECUTION IS TO END ( RETURN TO X )                    XECOMM
      1 CD     FLAGS  - EXECUTIVE FLAG WORD (0-OFF, 1-ON)                                       XECOMM
      1 CD              BITS 0-10 UNUSED                                                        XECOMM
      1 CD                    11 PROCESSOR ON-LINE DEBUG                                        XECOMM
      1 CD                    12 MANAGER ON-LINE DEBUG                                          XECOMM
      1 CD                    13 EXECUTIVE ON-LINE DEBUG                                        XECOMM
      1 CD                    14 PRODUCE A DUMP ON ALL TERMINATIONS                             XECOMM
      1 CD                    15 - MANAGER REQUEST TRANSACTION TRACE FLAG                       XECOMM
      1 CD     INTNAM - NAME OF INTERFACE TABLE INPUT TO INTERFACE TABLE EDITOR                  XECOMM
```

```
61    1 CB                OR ASSOCIATED WITH PROCESSOR EXECUTED IN MANUAL, SEMI OR
62    1 CB                AUTO-WITH-TRACE MODE (FIRST WORD = 0 IF DEFAULT INTERFACE
63    1 CB                TABLE)
64    1 CB          LU    - LOGICAL UNIT NUMBER OF TERMINAL BEING SUPPORTED BY THIS
65    1 CB                EXECUTIVE
66    1 CB          MASSTA - EXECUTIVE MASTER STATE FLAG (LEVEL LAST PASSED CONTROL BY
67    1 CB                XEXEC).  RESET TO ZERO BY LEVELS RETURNING TO DIRECTIVE
68    1 CB                LEVEL.
69    1 CB                BITS  0- 9 - NOT USED
70    1 CB                BITS 10-13 - DIRECTIVE CONTROL MODE IF BITS 14-15 = 0
71    1 CB                             0 - LIST
72    1 CB                             1 - TOC
73    1 CB                             2 - SAVE
74    1 CB                             3 - RECALL
75    1 CB                             4 - DELETE
76    1 CB                             5 - RENAME
77    1 CB                             6 - COPY
78    1 CB                             7 - CLEAR
79    1 CB                             8 - OFF
80    1 CB                             9 - STORE
81    1 CB                            10 - RESTORE
82    1 CB                            11 - UNLOAD
83    1 CB                            12 - LOAD
84    1 CB                            13 - BATCH
85    1 CB                BIT 11      - EXECUTION CONTROL INITIALIZATION INDICATOR
86    1 CB                              IF BITS 14-15 = 1
87    1 CB                              0 - INITIALIZATION FROM DIRECTIVE
88    1 CB                              1 - INITIALIZATION FOR REENTRY
89    1 CB                BITS 12-13 - EXECUTION CONTROL MODE IF BITS 14-15 = 1
90    1 CB                             0 - MANUAL
91    1 CB                             1 - SEMI-AUTOMATIC
92    1 CB                             2 - AUTOMATIC-T
93    1 CB                             3 - AUTOMATIC
94    1 CB                BITS 14-15 - EXECUTIVE STATE
95    1 CB                             0 - DIRECTIVE LEVEL
96    1 CB                             1 - EXECUTION CONTROL LEVEL
97    1 CB                             2 - SEQUENCE TABLE EDIT LEVEL
98    1 CB                             3 - INTERFACE TABLE EDIT LEVEL
99    1 CB          NOPROC - NUMBER OF PROCESSORS IN LIBRARY
100   1 CB          OLDIND - OLD INDEX TO CURRENTLY EXECUTING ENTRY IN SEQUENCE TABLE
101   1 CB          PRCNAM - NAME OF PROCESSOR FOR WHICH INTERFACE TABLE EDITOR WAS
102   1 CB                INVOKED OR BEING EXECUTED IN MANUAL, SEMI OR AUTO-WITH-
103   1 CB                TRACE MODE
104   1 CB          QUAL  - USER UNIQUE FILE NAME QUALIFIER (SIXTH CHARACTER OF NAME)
105   1 CB          REQBUF - BUFFER FOR MANAGER WORK AREA REQUESTS (SEE SDD 6.2.7)
106   1 CB          REQPTR - POINTER TO END OF LAST COMPLETED 8 WORD ENTRY IN REQBUF
107   1 CB                (0 INDICATES REQBUF EMPTY) OR RETURN CODE FORM XREQ
108   1 CB                (HIGH BYTE = PARAM1, LOW BYTE = PARAM2, SEE SDD 6.2.6.3)
109   1 CB          SEQEND - TERMINATING SEQUENCE NUMBER OF SEQUENCE TABLE EXECUTED
110   1 CB                IN SEMI OR AUTO MODE AS PASSED TO THE MANAGER
111   1 CB          SEQNAM - NAME OF SEQUENCE TABLE INPUT TO SEQUENCE TABLE EDITOR OR
112   1 CB                EXECUTED IN SEMI OR AUTO MODE
113   1 CB          SEQPTR - POINTER TO LAST SEQUENCE TABLE ENTRY EXECUTED IN SEMI
114   1 CB                OR AUTO-WITH-TRACE MODE
115   1 CB          SEQSTR - INITIAL SEQUENCE NUMBER OF SEQUENCE TABLE EXECUTED IN
116   1 CB                SEMI OR AUTO MODE
117   1 CB          SUBSTA - EXECUTIVE SUB-STATE FLAG (LEVEL IN COMMUNICATION WITH
118   1 CB                USER TERMINAL).  SET TO LEVEL TO BE INITIALIZED NEXT OR
119   1 CB                ZERO IF LEVEL INITIALIZATION FAILS.
```

```
120    1 CD              0 - DIRECTIVE LEVEL                                                    XECOMM
121    1 CD              1 - EXECUTION CONTROL LEVEL                                            XECOMM
122    1 CD              2 - SEQUENCE TABLE EDIT LEVEL                                          XECOMM
123    1 CD              3 - INTERFACE TABLE EDIT LEVEL                                         XECOMM
124    1 CD      TABEND - SEQUENCE # OF LAST ENTRY IN SEQUENCE TABLE                            XECOMM
125    1 CD      TKNLNG - NUMBER OF CONSECUTIVE ENTRYS PRESENTLY ALLOCATED IN                   XECOMM
126    1 CD               TOKENS (29)                                                           XECOMM
127    1 CD      TOKENS - IDENTIFYING NUMBERS OF TOKENS RETURNABLE IN COMBUF                    XECOMM
128    1 CD         (1)= 0=END OF BUFFER          (2)= 1=INTEGER NUMBER                       XECOMM
129    1 CD         (3)= 2=REAL NUMBER            (4)= 3=DOUBLE PRECISION NUMBER              XECOMM
130    1 CD         (5)= 5=SIX CHARACTER NAME     (6)= 8=CHARACTER STRING                     XECOMM
131    1 CD         (7)=10= +                     (8)=11= -                                   XECOMM
132    1 CD         (9)=12= *                    (10)=13= /                                   XECOMM
133    1 CD        (11)=14= <                    (12)=15= >                                   XECOMM
134    1 CD        (13)=16= #                    (14)=17= @                                   XECOMM
135    1 CD        (15)=18= =                    (16)=19= &                                   XECOMM
136    1 CD        (17)=20= ?                    (18)=21= (                                   XECOMM
137    1 CD        (19)=22= )                    (20)=23= ' BEGIN SYMBOLIC STRING             XECOMM
138    1 CD        (21)=24= ' END SYMBOLIC STR   (22)=25= %                                   XECOMM
139    1 CD        (23)=26= /                    (24)=27= $                                   XECOMM
140    1 CD        (25)=28= .                    (26)=29= LEFT BRACKET                        XECOMM
141    1 CD        (27)=30= RIGHT BRACKET        (28)=31= REPEAT                              XECOMM
142    1 CD        (29)=32=,                     (30)=33=;                                    XECOMM
143    1 CD        (31) (UNUSED)                 (32) (UNUSED)                                XECOMM
144    1 CD                                                                                   XECOMM
145    1 CD********                                                                           XECOMM
```

```
147    1 C     XB CONTENTS (DIRECTIVE LEVEL)                                                    XBCOMM
148    1 CB                                                                                     XBCOMM
149    1 CB     INTEGER                                                                         XBCOMM
150    1 CB    * DIRECT(50)                                                                     XBCOMM
151    1 CB                                                                                     XBCOMM
152    1 CB     EQUIVALENCE                                                                     XBCOMM
153    1 CB    * (XB(1)   ,NUMDIR)              ,(XB(2)   ,DIRECT(1))                          XBCOMM
154    1 CB    *,(XB(151),BEGINNING OF DIRECTIVE DEPENDENT ALLOCATION)                          XBCOMM
155    1 CB                                                                                     XBCOMM
156    1 CB     DIRECT - FIXED ORDER LIST OF FDS DIRECTIVES (FOUR CHARACTERS PER               XBCOMM
157    1 CB              DIRECTIVE)                                                           XBCOMM
158    1 CB     NUMDIR - NUMBER OF DIRECTIVES ACTUALLY IN DIRECT                               XBCOMM
159    1 CB                                                                                     XBCOMM
160    1 CB********                                                                            XBCOMM
```

```
162   1 CD     XB CONTENTS (EXECUTION CONTROL LEVEL)                                                    XBCOMM
163   1 CD                                                                                              XBCOMM
164   1 CD      INTEGER                                                                                 XBCOMM
165   1 3D     * ASCENT                        ,CURIND                                                  XBCOMM
166   1 CD     *,RESIND                        ,SEQLEN                                                  XBCOMM
167   1 CD     *,SEQNO                         ,SEQTAB                                                  XBCOMM
168   1 CD                                                                                              XBCOMM
169   1 CD      DIMENSION                                                                               XBCOMM
170   1 CD     * ASCENT(10)                    ,LIBD(150)                                               XBCOMM
171   1 CD     *,SEQTAB(1150)                                                                           XBCOMM
172   1 CD                                                                                              XBCOMM
173   1 CD      EQUIVALENCE                                                                             XBCOMM
174   1 CD     * (XB(1)   ,NOPRC2)             ,(XB(2)   ,LIBD(1))                                      XBCOMM
175   1 CD     *,(XB(235),RESIND)              ,(XB(236),ASCENT(1))                                     XBCOMM
176   1 CD     *,(XB(246),CURIND)              ,(XB(249),SEQNO)                                         XBCOMM
177   1 CD     *,(XB(250),SEQLEN)              ,(XB(251),SEQTAB(1))                                     XBCOMM
178   1 CD                                                                                              XBCOMM
179   1 CD      ASCENT - SEQUENCE TABLE ENTRY IN ASCII TO PROMPT USER                                   XBCOMM
180   1 CD               IN SEMI MODE ONLY                                                              XBCOMM
181   1 CD      CURIND - CURRENT INDEX TO EXECUTING SEQUENCE ENTRY                                      XBCOMM
182   1 CD      LIBD   - LIBRARY DIRECTORY PROCESSOR NAME TABLE                                         XBCOMM
183   1 CD      NOPRC2 - NUMBER OF PROCESSORS IN LIBD (SAME AS XE(143))                                 XBCOMM
184   1 CD      RESIND - INDEX OF RESET ENTRY WHEN RESET SEQUENCE # IS REQUESTED                        XBCOMM
185   1 CD      SEQLEN - LENGTH OF SEQUENCE TABLE                                                       XBCOMM
186   1 CD      SEQNO  - NUMBER OF ENTRIES IN SEQUENCE TABLE                                            XBCOMM
187   1 CD      SEQTAB - SEQUENCE TABLE CURRENTLY BEING EXECUTED                                        XBCOMM
188   1 CD                                                                                              XBCOMM
189   1 CD*-------                                                                                      XBCOMM
```

```
191   1 C)    XB CONTENTS (SEQUENCE TABLE EDIT LEVEL)
192   1 C)
193   1 CD     INTEGER
194   1 CD    * BEGNO                     ,DEBUG
195   1 CD    *,ENDNO                     ,PRCNAM
196   1 CD    *,PRMLEN                    ,PRMTMD
197   1 CD    *,PROMPT                    ,OLDTAB
198   1 CD    *,SEQNO                     ,TABNDX
199   1 CD    *,TABSIZ                    ,WKBLNG
200   1 CD    *,WKBUF                     ,XLIBD
201   1 CD
202   1 CD     EQUIVALENCE
203   1 CD    * (XB(27), DEBUG)           ,(XB(28), OLDTAB)
204   1 CD    *,(XB(31), NEWTAB)          ,(XB(34), PRMTMD)
205   1 CD    *,(XB(35), NUMENT)          ,(XB(36), PRMLEN)
206   1 CD    *,(XB(37), PROMPT)          ,(XB(49), IRETC)
207   1 CD    *,(XB(50), SEQNO)           ,(XB(51), PRCNAM)
208   1 CD    *,(XB(54), INTNAM)          ,(XB(57), TABNDX)
209   1 CD    *,(XB(58), TABSIZ)          ,(XB(59), INSERT)
210   1 CD    *,(XB(60), LIBSIZ)          ,(XB(61), BEGNO)
211   1 CD    *,(XB(62), ENDNO)
212   1 CD    *,(XB(100),WKBLNG)          ,(XB(101),WKBUF)
213   1 CD    *,(XB(1209),XLIBD)
214   1 CD
215   1 CD     DIMENSION
216   1 CD    * INTNAM(3)                 ,NEWTAB(3)
217   1 CD    *,OLDTAB(3)                 ,PRCNAM(3)
218   1 CD    *,PROMPT(12)                ,WKBUF(1050)
219   1 CD    *,XLIBD(192)
220   1 CD
221   1 CD     DEBUG  = FLAG SET TO VALUE OF EXECUTIVE DEBUG OPTION (BITS 13 - 15)
222   1 CD              OF FLAGS (XE(4))
223   1 CD     BEGNO  = INDEX IN WKBUF TO FIRST TABLE ENTRY TO BE LISTED OR DELETED
224   1 CD     ENDNO  = INDEX IN WKBUF TO LAST TABLE ENTRY TO BE LISTED OR DELETED
225   1 CD     INSERT = FLAG INDICATING TO XSENT WHAT TYPE OF EDIT TO PERFORM :
226   1 CD                       = 0 => REPLACE ENTRY AT TABNDX
227   1 CD                       = 1 => INSERT A NEW ENTRY IN FRONT OF TABNDX
228   1 CD                       = 2 => ADD AN ENTRY AT THE BOTTOM OF THE TABLE
229   1 CD     INTNAM = INTERFACE TABLE NAME INPUT OR 0
230   1 CD     IRETC  = RETURN CODE SET TO -1 WHEN XSEQE IS TO TERMINATE
231   1 CD     LIBOSZ = SIZE IN WORDS OF PROCESSOR DIRECTORY, XLIBD
232   1 CD     NEWTAB = NAME OF SEQUENCE TABLE BEING CREATED
233   1 CD     NUMENT = NUMBER OF ENTRIES (INCLUDING THOSE MARKED FOR DELETION)
234   1 CD              CURRENTLY IN WKBUF
235   1 CD     OLDTAB = NAME OF EXISTING SEQUENCE TABLE INPUT TO THE EDITOR
236   1 CD     PRMLEN = LENGTH IN WORDS OF PROMPT CREATED BY XSPRM
237   1 CD     PRMTMD = CURRENT PROMPTING MODE :
238   1 CD                    = 1 => UPDATE MODE ( #: )
239   1 CD                    = 2 => CREATE OR MISSING MODE (# 200=: )
240   1 CD                    = 3 => ALL MODE ( # 200=PROC,TABLE: )
241   1 CD     PROMPT = PROMPT BUILT BY XSPRM
242   1 CD     SEQNO  = CURRENT SEQUENCE NUMBER BEING PROMPTED
243   1 CD     TABSIZ = SIZE IN WORDS OF TABLE IN WKBUF
244   1 CD     TABNDX = INDEX IN WKBUF TO CURRENT TABLE ENTRY
245   1 CD     WKBLNG = MAXIMUM SIZE IN WORDS OF WKBUF
246   1 CD     WKBUF  = WORKING BUFFER CONTAINING TABLE DURING EDIT SESSION
247   1 CD     XLIBD  = PROCESSOR LIBRARY DIRECTORY
248   1 CD
249   1 C*********
```

```
251   1 CD     XB CONTENTS (INTERFACE TABLE EDIT LEVEL)                                                                XBCOMM
252   1 CD                                                                                                             XBCOMM
253   1 CD      INTEGER                                                                                                XBCOMM
254   1 CD     * ARGNO                        ,BITNO                                                                   XBCOMM
255   1 CD     *,BITNUM                       ,CFLAG                                                                   XBCOMM
256   1 CD     *,DEBUG                        ,DIRECT(6)                                                               XBCOMM
257   1 CD     *,DFLAG                        ,EQULAT(3)                                                               XBCOMM
258   1 CD     *,PRMLEN                       ,PRMTMD                                                                  XBCOMM
259   1 CD     *,PROMPT(30)                   ,SFLAG                                                                   XBCOMM
260   1 CD     *,VERSON                                                                                                XBCOMM
261   1 CD     *,WKBLNG                       ,WKBUF(1300)                                                             XBCOMM
262   1 CD                                                                                                             XBCOMM
263   1 CD      DIMENSION                                                                                              XBCOMM
264   1 CD     * ISIZES(9)                    ,NEWTAB(3)                                                               XBCOMM
265   1 CD                                                                                                             XBCOMM
266   1 CD      EQUIVALENCE                                                                                            XBCOMM
267   1 CD     *,(XB(23), COMFLG)                                                                                      XBCOMM
268   1 CD     *,(XB(24), ICLASS)             ,(XB(25), LSTFLG)                                                        XBCOMM
269   1 CD     *,(XB(26), VERSON)             ,(XB(27), DEBUG)                                                         XBCOMM
270   1 CD     *,(XB(28), LISTLU)                                                                                      XBCOMM
271   1 CD     *,(XB(29), MODSAV)             ,(XB(30), IARG4)                                                         XBCOMM
272   1 CD     *,(XB(31), IDIM)               ,(XB(32), ISIZE)                                                         XBCOMM
273   1 CD     *,(XB(33), SFLAG)              ,(XB(34), JSUB)                                                          XBCOMM
274   1 CD     *,(XB(35), CFLAG)              ,(XB(36), DFLAG)                                                         XBCOMM
275   1 CD     *,(XB(37), ARGNO)              ,(XB(38), ARGPTR)                                                        XBCOMM
276   1 CD     *,(XB(39), ISUB)               ,(XB(40), IOFLAG)                                                        XBCOMM
277   1 CD     *,(XB(41), PRMTMD              ,(XB(42), PRMLEN)                                                        XBCOMM
278   1 CD     *,(XB(43), PROMPT(1))                                                                                   XBCOMM
279   1 CD     *,(XB(73), ISIZES)             ,(XB(82), LENEFF)                                                        XBCOMM
280   1 CD     *,(XB(83), ITYPE)              ,(XB(84), BITNO)                                                         XBCOMM
281   1 CD     *,(XB(85), BITNUM)             ,(XB(86), NDXBTM)                                                        XBCOMM
282   1 CD     *,(XB(87), NOBITM)             ,(XB(88), LITDSP)                                                        XBCOMM
283   1 CD     *,(XB(89), LITDWN)             ,(XB(90), LITPTR)                                                        XBCOMM
284   1 CD     *,(XB(91), LITLEN)             ,(XB(92), NARG)                                                          XBCOMM
285   1 CD     *,(XB(93), LITSIZ)             ,(XB(94), IRETC)                                                         XBCOMM
286   1 CD     *,(XB(95), IARG)               ,(XB(96), NUMARG)                                                        XBCOMM
287   1 CD     *,(XB(97), NEWTAB)             ,(XB(100),WKBLNG)                                                        XBCOMM
288   1 CD     *,(XB(101),WKBUF)                                                                                       XBCOMM
289   1 CD                                                                                                             XBCOMM
290   1 CD      ARGNO  - CURRENT ARGUMENT'S NO. (I.E. 1 TO 64)                                                         XBCOMM
291   1 CD      ARGPTR - INDEX IN WKBUF TO ARGUMENT PROMPT FOR CURRENT ARGUMENT                                        XBCOMM
292   1 CD      BITNO  - BIT NO. IN BIT MASK CURRENTLY BEING PROCESSED                                                 XBCOMM
293   1 CD      BITNUM - BIT NO. IN BIT MASK OF NEXT DIFFERING BIT                                                     XBCOMM
294   1 CD      CFLAG  - FLAG SET TO VALUE OF COMPLETE BIT FOR THIS ARGUMENT                                           XBCOMM
295   1 CD      COMFLG - FLAG SET TO 1 IF A CONTINUATION PROMPT OF CURRENT ARGUMENT                                    XBCOMM
296   1 CD               HAS BEEN EXPLICTITLY REQUESTED VIA A TRAILING COMMA OR IS                                     XBCOMM
297   1 CD               IMPLICITLY REQUIRED BECAUSE INCOMPLETE ELEMENTS REMAIN BEYOND                                 XBCOMM
298   1 CD               LAST ENTERED ELEMENT OF THIS ARGUMENT                                                         XBCOMM
299   1 CD      DEBUG  - FLAG SET TO VALUE OF DEBUG FLAG (BITS 13-15 OF XE(4))                                         XBCOMM
300   1 CD      DIRECT - ASCII ARRAY OF VALID INTERFACE TABLE EDITOR DIRECTIVES                                        XBCOMM
301   1 CD      DFLAG  - FLAG SET TO VALUE OF DATA BIT FOR THIS ARGUMENT                                               XBCOMM
302   1 CD      IARG   - INDEX IN WKBUF TO CURRENT ARG                                                                XBCOMM
303   1 CD      ICLASS - CLASS OF THIS ARGUMENT                                                                        XBCOMM
304   1 CD      IOFLAG - FLAG SET TO I/O BITS FOR THIS ARGUMENT                                                        XBCOMM
305   1 CD      IRETC  - GENERAL RETURN FLAG (0=OK, -1=ERROR, 5='EXIT')                                                XBCOMM
306   1 CD      ISIZE  - TOTAL SIZE (IN WORDS) FOR THIS ARGUMENT                                                       XBCOMM
307   1 CD      ISIZES - NO. OF WORDS/ELEMENT FOR EACH ALLOWED ITYPE VALUE                                             XBCOMM
308   1 CD      ISUB   - CURRENT EFFECTIVE SUBSCRIPT FOR THIS ARGUMENT                                                 XBCOMM
309   1 CD      ITYPE  - TYPE FLAG FOR THIS ARGUMENT                                                                   XBCOMM
```

```
310   1 CD     JSUB   - CURRENT SECOND SUBSCRIPT (OR 0) FOR THIS ARGUMENT                          XBCOMM
311   1 CD     LENEFF - NO. OF WORDS/ELEMENT FOR THIS ARGUMENT                                     XBCOMM
312   1 CD     LISTLU - LU TO WHICH PRINT SHOULD GO (USED WHEN                                     XBCOMM
313   1 CD              LIST DIRECTIVE CALLS XILSD OR XICHR)                                       XBCOMM
314   1 CD     LITDSP - DISPL. TO LITERAL DATA FOR THIS ARGUMENT                                   XBCOMM
315   1 CD     LITDWN - INDEX IN WKBUF TO END OF LITERAL DATA                                      XBCOMM
316   1 CD     LITLEN - LENGTH OF LITERAL DATA AREA OF WKBUF                                       XBCOMM
317   1 CD     LITPTR - INDEX IN WKBUF TO START OF LITERAL DATA                                    XBCOMM
318   1 CD     LITSIZ - NUMBER OF ARRAY ELEMENTS FOR THIS ARGUMENT (LOGICAL SIZE)                  XBCOMM
319   1 CD     LSTFLG - FLAG USED TO DETERMINE ORIGIN OF A                                         XBCOMM
320   1 CD              CALL TO XILSD OR XICHR.                                                    XBCOMM
321   1 CD                     = 0, CALLED FROM LIST DIRECTIVE (PER CENT PROMPT) TO LIST           XBCOMM
322   1 CD                          AN INTERFACE TABLE                                            XBCOMM
323   1 CD                     = 3, CALLED FROM INTERFACE TABLE EDITOR TO LIST ARGUMENT           XBCOMM
324   1 CD                          DATA, PARMETER OR INCOMPLETE INDICATORS                       XBCOMM
325   1 CD                     = 4, CALLED FROM LIST DIRECTIVE (PER CENT PROMPT) TO LIST          XBCOMM
326   1 CD                          A DATA ELEMENT'S VALUE(S)                                     XBCOMM
327   1 CD     MODSAV - PREVIOUS VALUE OF PRMTMD WHILE PRMTMD = 4 (CONTINUE)                       XBCOMM
328   1 CD     NARG   - INDEX IN WKBUF TO START OF SHORT PROMPTS                                   XBCOMM
329   1 CD     NDXBTM - INDEX IN WKBUF TO BIT MASK WORD(S) FOR THIS ARGUMENT                       XBCOMM
330   1 CD     NEWTAB - ASCII NAME OF TABLE BEING GENERATED                                        XBCOMM
331   1 CD     NOBITM - NO. OF BIT MASKS ASSOCIATED WITH THIS ARGUMENT                             XBCOMM
332   1 CD     NUMARG - NO. OF ARGUMENTS IN THIS INTE. TABLE                                       XBCOMM
333   1 CD     NUMDIR - NUMBER OF DIRECTIVES ACTUALLY IN DIRECT                                    XBCOMM
334   1 CD     PRMLEN - LENGTH (IN WORDS) OF PROMPT BUILT                                          XBCOMM
335   1 CD     PRMTMD - CURRENT PROMPTING MODE                                                     XBCOMM
336   1 CD                = 1, PROMPT FOR INCOMPLETE ARGUMENTS (MISSING)                         XBCOMM
337   1 CD                = 3, PROMPT WITH A LIST OF CURRENT VALUES                              XBCOMM
338   1 CD                = 4, CONTINUATION OF SAME ARGUMENT                                     XBCOMM
339   1 CD                = 5, PROMPT WITH "/" (CREATE MODE)                                     XBCOMM
340   1 CD     PROMPT - ASCII ARRAY FOR PROMPT BUILT                                               XBCOMM
341   1 CD     SFLAG  - FLAG SET TO VALUE OF SUBSCRIPT BIT FOR THIS ARGUMENT                       XBCOMM
342   1 CD     VERSON - VERSION NO. OF THIS INTERFACE TABLE                                        XBCOMM
343   1 CD     WKBLNG - LENGTH OF WKBUF                                                            XBCOMM
344   1 CD     WKBUF  - WORKING BUFFER FOR INTERFACE TABLE BEING EDITED                            XBCOMM
345   1 CD              ORGANIZED AS :                                                             XBCOMM
346   1 CD                                                                                         XBCOMM
347   1 CD                       HEADER (7 WORDS)                                                  XBCOMM
348   1 CD                       ARGUMENT CHARACTERISTICS (NUMARG*7 WORDS)                         XBCOMM
349   1 CD                       ARGUMENT PROMPTS (NUMARG*3 WORDS)                                 XBCOMM
350   1 CD                       LITERAL DATA AND PARAMETER SUBSCRIPTS (LITLEN WORDS)              XBCOMM
351   1 CD                                                                                         XBCOMM
352   1 CD********                                                                                 XBCOMM
```

3.0 FDS EXECUTIVE MESSAGES

The list of messages generated by the Executive are presented in this section.

```
                        FDS  ERROR  MESSAGES
*
*
*
* XA      ATTENTION FUNCTION
*** XA01 MANAGER HAS TERMINATED - REPLY TO CONTINUE TERMINATION
*** XA02 USER INITIATED INTERRUPT ENTER REQUEST - KILL(X),STATUS(S),RETURN( )
*** XA03 ERROR, LU IS NOT SIGNED ON TO FDS
*** XA04 FDS ATTENTION FUNCTION TERMINATING
*** XA05 FDS MANAGER SIGNALED TO TERMINATE BOTTOM ASSOC TASK
*** XA06 FDS EXECUTIVE ACTIVE - NO ACTION TAKEN
*** XA07 FDS PROCESSOR 'NAME' SCHEDULED TO ABORT
*** XA08 MANAGER WAITING FOR SYSTEM RESOURCES ... NO ACTION TAKEN
*
* XB      BATCH JOB CREATION
*
* XC      CONFIGURATION PROGRAMS
*** XC01 LU 'NN' SIGNED ON TO FDS
*** XC03 LU IS CURRENTLY USING ID 'ID' - SIGN ON REJECTED
*** XC04 -LU 'LU' IS AN INVALID LU
*** XC05 FDS CURRENTELY AT MAX USER'S. CANNOT SIGN ON
*** XC06 'LU' IS ALREADY SIGNED ON TO FDS
*** XC07 CANNOT FIND 'NAME' ID- SIGN ON TERMINATED
*** XC08 ENTER VALID USER ID (A-Z)
*
* XM      MANAGER
*** XM01 INVALID REQUEST II FROM 'NAME'
*** XM02 FDS SIGN OFF FOR LU 'LU'
*** XM03 INVALID BACK CHAIN FOR CURRENT PROCESSOR
*** XM04 'N' TRACKS NOT AVAILABLE FOR DWA
*** XM05 NO DWA SPACE FOR 'NAME'
*** XM06 NO AWA FOR DWA DIRECTORY
*** XM07 I/O ERROR FOR DWA, STATUS = 'NNNNNN'
*** XM08 TOC SEARCH ERROR, PHASE 3 COMPRESS
*
* XP      PROCESSOR SERVICES
*** XP01 TYPE OF RESPONSE DOES NOT MATCH TYPE REQUESTED
*** XP02 DATA AREA OVERFLOW
*** XP03 SUBSCRIPT OUT OF RANGE
*** XP04 INVALID ENTRY AFTER SUBSCRIPT
*** XP05 INVALID REPEAT ENTRY
*** XP06 TOO MANY NESTED REPEATS
*** XP07 INVALID SUBSCRIPT
*** XP08 RESPONSE IS TOO LONG FOR BUFFER
*** XP09 INVALID RESPONSE ON OR AFTER COLUMN NN
*** XP10 PROCESSOR INITIALIZATION ERROR
*** XP11 AWA ACCESS FAILURE FOR ...,...
*** XP12 PARAMETER I/O INCONSISTENCY
*** XP13 INVALID PARAMETER REQUEST
*** XP14 INVALID DRDE FILE ATTRIBUTE(S)
*** XP15 RETRIEVAL OF TOO MUCH DATA REQUESTED
*
* XV      SYSTEM SERVICES
*** XV01 'ID-NAME'  NAMED PROGRAM MADE A PAW REQUEST RESERVED FOR MANAGER
*** XV02 'ID-NAME'  NAMED PROGRAM NOT IN WAIT LIST FOR MANAGER REPLY
*** XV03 'ID-NAME'  NAMED PROGRAM HAS INVALID BACK CHAIN TO MANAGER
*
```

4.0 PDL LISTING PROGRAM

The detailed logic flow of the program that generates PDL listing is presented as follows.

```
      1 CD1        PDL STRUCTURED LISTING PROGRAM                                                    PDLIST
      1 CD1                                                                                          PDLIST
      1 C*********                                                                                   PDLIST
      1 CD2        INPUT                                                                             PDLIST
      1 CD2        80 COLUMN PDL IMAGES SUBJECT TO THE FOLLOWING CONVENTIONS:                        PDLIST
      1 CD2        . IN COLUMN 1 INDICATES PAGE EJECT AND THE FIRST TOKEN (6 OR LESS                 PDLIST
      1 CD2          CHARACTERS) IS REPRODUCED IN COLUMNS 127-132 OF OUTPUT LISTING                  PDLIST
      1 CD2          UNTIL NEXT . RECORD IS DETECTED.  IF RECORD CONTAINS ONLY THE . THE             PDLIST
      1 CD2          TOKEN FROM THE PREVIOUS . RECORD CONTINUES TO APPEAR IN THE                     PDLIST
      1 CD2          IDENTIFICATION COLUMNS OF THE OUTPUT                                            PDLIST
      1 CD2        * IN COLUMN 1 INDICATES A COMMENT TO BE COPIED TO THE OUTPUT LISTING.             PDLIST
      1 CD2     OTHER IN COLUMN 1 INDICATES PDL RECORD TO BE STRUCTURED BASED ON KEY                 PDLIST
      1 CD2          WORDS APPEARING AS FIRST NON-BLANK CHARACTERS (SEE LOGIC).                      PDLIST
      1 CD2                                                                                          PDLIST
      1 C*********                                                                                   PDLIST
      1 CD3        OUTPUT                                                                            PDLIST
      1 CD3        132 COLUMN LISTING AS FOLLOWS                                                     PDLIST
      1 CD3          2-6 SEQUENCE NUMBER                                                             PDLIST
      1 CD3         8-11 STRUCTURE LEVEL NUMBER                                                      PDLIST
      1 CD3       13-125 STRUCTURED LISTING OF 80 COLUMN INPUT RECORDS                               PDLIST
      1 CD3      127-132 IDENTIFICATION                                                              PDLIST
      1 CD3        FOLLOWING THE LISTING IS A SYMBOL DEFINITION TABLE INDICATING THE                 PDLIST
      1 CD3        SEQUENCE NUMBER OF THE LINE CONTAINING EACH 'BEGIN NAME' AND                      PDLIST
      1 CD3        ':LABEL:'.                                                                        PDLIST
      1 CD3                                                                                          PDLIST
      1 C*********                                                                                   PDLIST
      1 CD5        NOTES                                                                             PDLIST
      1 CD5        USES FSTWRD & SORT1                                                               PDLIST
      1 CD5                                                                                          PDLIST
      1 C*********                                                                                   PDLIST
```

```
33    1 BEGIN PDLIST                                                                                                    PDLIST
34    2    INITIALIZE SEQUENCE NUMBER, LEVEL AND DEFINITION TABLE COUNTER                                               PDLIST
35    2    DO UNTIL END-OF-FILE INPUT                                                                                   PDLIST
36    3       READ RECORD                                                                                               PDLIST
37    3       INCREMENT SEQUENCE NUMBER                                                                                 PDLIST
38    3       IF COLUMN 1 = .                                                                                           PDLIST
39    3       THEN                                                                                                      PDLIST
40    4          SET PAGE EJECT IN IMAGE                                                                                PDLIST
41    4          IF REMAINDER OF IMAGE IS NOT BLANK                                                                     PDLIST
42    4          THEN                                                                                                   PDLIST
43    5             SET ID TO CONTENTS OF FIRST NON-BLANK FIELD                                                         PDLIST
44    4          ENDIF                                                                                                  PDLIST
45    3       ELSE                                                                                                      PDLIST
46    4          CLEAR LEVEL INCREMENT                                                                                  PDLIST
47    4          IF COLUMN 1 NOT= *                                                                                     PDLIST
48    4          THEN                                                                                                   PDLIST
49    5             IF FIRST CHARACTER = :       - INDICATES :LABEL:                                                  PDLIST
50    5             THEN                                                                                               PDLIST
51    6                SET LINE SKIP IN IMAGE                                                                          PDLIST
52    6                GENERATE DEFINITION TABLE ENTRY FOR LABEL                                                       PDLIST
53    5             ELSE                                                                                               PDLIST
54    6                CALL FSTWRD TO GET FIRST WORD OF PDL                                                            PDLIST
55    6                LOOKUP FIRST WORD IN KEY WORD TABLE                                                             PDLIST
56    6 *                                                                                                              PDLIST
57    6 *                  KEY WORD TABLE CONTAINS                                                                     PDLIST
58    6 *                                                                                                              PDLIST
59    6 *                     1   BEGIN        - BEGIN SECTION INDICATOR                                               PDLIST
60    6 *                     2   IF           - SECTION INDICATOR                                                     PDLIST
61    6 *                     3   DO           - SECTION INDICATOR                                                     PDLIST
62    6 *                     4   DOFOR        - SECTION INDICATOR                                                     PDLIST
63    6 *                     5   DOUNTI       - SECTION INDICATOR                                                     PDLIST
64    6 *                     6   DOWHIL       - SECTION INDICATOR                                                     PDLIST
65    6 *                     7   CASE         - SECTION INDICATOR                                                     PDLIST
66    6 *                     8   START        - SECTION INDICATOR                                                     PDLIST
67    6 *                     9   STARTS       - SECTION INDICATOR                                                     PDLIST
68    6 *                    10   ELSE         - SECTION SEPARATOR                                                     PDLIST
69    6 *                    11   THEN         - SECTION SEPARATOR                                                     PDLIST
70    6 *                    12   EXIT         - SECTION SEPARATOR                                                     PDLIST
71    6 *                    13   EXITIF       - SECTION SEPARATOR                                                     PDLIST
72    6 *                    14   OR           - SECTION SEPARATOR                                                     PDLIST
73    6 *                    15   ORELSE       - SECTION SEPARATOR                                                     PDLIST
74    6 *                    16   ENDLOO       - SECTION SEPARATOR                                                     PDLIST
75    6 *                    17   END          - END OR END LOOP?                                                      PDLIST
76    6 *                    18   ENDIF        - SECTION TERMINATOR                                                    PDLIST
77    6 *                    19   ENDDO        - SECTION TERMINATOR                                                    PDLIST
78    6 *                    20   ENDCAS       - SECTION TERMINATOR                                                    PDLIST
79    6 *                    21   ENDSEA       - SECTION TERMINATOR                                                    PDLIST
80    6 *                                                                                                              PDLIST
81    6                   IF KEY WORD LOCATED                                                                          PDLIST
82    6                   THEN                                                                                         PDLIST
83    7                      CASE LOCATION (:BEGIN:, :SECIND:, :SECIND:, :SECIND:, :SECIND:, :SECIND:,                 PDLIST
84    8                         (:SECIND:, :SECIND:, :SECIND:, :SEP:, :SEP:, :SEP:, :SEP:, :SEP:, :SEP:, :SEP:,     PDLIST
85    8                         (:END?:, :TERM:, :TERM:, :TERM:, :TERM:)                                             PDLIST

86    8                            :BEGIN:  GENERATE DEFINITION TABLE ENTRY FOR SECTION NAME                         PDLIST
87    8                            SET LEVEL INCREMENT = 1                                                           PDLIST

88    8                            :SECIND: SET LEVEL INCREMENT = 1                                                  PDLIST
```

```
 89   8                    :SEP:  SET LEVEL INCREMENT = 1                                                    PDLIST
 90   8                    DECREMENT LEVEL                                                                   PDLIST

 91   8                    :END?:  CALL FSTWRD TO GET NEXT WORD OF PDL                                       PDLIST
 92   8                    DECREMENT LEVEL                                                                   PDLIST
 93   8                    IF WORD = LOOP                                                                    PDLIST
 94   8                    THEN                                                                              PDLIST
 95   9                       SET LEVEL INCREMENT = 1                                                        PDLIST
 96   8                    ENDIF                                                                             PDLIST

 97   8                    :TERM: DECREMENT LEVEL                                                            PDLIST
 98   7                 ENDCASE                                                                              PDLIST
 99   6              ENDIF                                                                                   PDLIST
100   5           ENDIF                                                                                      PDLIST
101   5           COMPUTE INDENTATION FACTOR = MINIMUM OF 3(LEVEL-1) AND 36                                  PDLIST
102   4        ELSE                                                                                          PDLIST
103   5           SET INDENTATION FACTOR = 1                                                                 PDLIST
104   4        ENDIF                                                                                         PDLIST
105   4        CONSTRUCT OUTPUT IMAGE FROM SEQUENCE NUMBER, LEVEL, INDENTATION FACTOR,                       PDLIST
106   4        INPUT RECORD AND ID FIELD                                                                     PDLIST
107   4        OUTPUT IMAGE                                                                                  PDLIST
108   4        APPLY LEVEL INCREMENT                                                                         PDLIST
109   4        CLEAR PAGE EJECT                                                                              PDLIST
110   3     ENDIF                                                                                            PDLIST
111   2  ENDDO                                                                                               PDLIST
112   2  CALL SORT1 TO ORDER DEFINITION TABLE                                                                PDLIST
113   2  OUTPUT DEFINITION TABLE                                                                             PDLIST
114   1 END PDLIST                                                                                           PDLIST
```

```
116   1 CD1       EXTRACT THE FIRST TOKEN FROM A PDL RECORD                                                    FSTWRD
117   1 CD1                                                                                                    FSTWRD
118   1 C**********                                                                                            FSTWRD
119   1 CD2       INPUT                                                                                        FSTWRD
120   1 CD2       SINGLE CHARACTER PER WORD RECORD AND LENGTH                                                  FSTWRD
121   1 CD2                                                                                                    FSTWRD
122   1 C**********                                                                                            FSTWRD
123   1 CD3       OUTPUT                                                                                       FSTWRD
124   1 CD3       FIRST (NEXT) TOKEN IN SIX CHARACTER WORD.  BLANKS AND : ARE                                  FSTWRD
125   1 CD3       DELIMETERS.  THE DISPLACEMENT OF THE NEXT CHARACTER IN THE RECORD IS                         FSTWRD
126   1 CD3       ALSO OUTPUT                                                                                  FSTWRD
127   1 CD3                                                                                                    FSTWRD
128   1 C**********                                                                                            FSTWRD
129   1 *                                                                                                      FSTWRD
130   1 *                                                                                                      FSTWRD
131   1 *                                                                                                      FSTWRD
132   1 *                                                                                                      FSTWRD
133   1 BEGIN FSTWRD                                                                                           FSTWRD
134   2    BLANK OUTPUT WORD                                                                                   FSTWRD
135   2    LOCATE FIRST NON-BLANK CHARACTER                                                                    FSTWRD
136   2    DO UNTIL SIX CHARACTERS STORED OR END-OF-RECORD                                                      FSTWRD
137   3       IF CHARACTER IS NON-BLANK AND NON-:                                                              FSTWRD
138   3       THEN                                                                                             FSTWRD
139   4          STORE CHARACTER                                                                               FSTWRD
140   3       ELSE                                                                                             FSTWRD
141   3       EXIT DO                                                                                          FSTWRD
142   3       ENDIF                                                                                            FSTWRD
143   2    ENDDO                                                                                               FSTWRD
144   2    RETURN LOCATION                                                                                     FSTWRD
145   1 END FSTWRD                                                                                             FSTWRD
```

```
147    1 CD1      ALGEBRAIC AND/OR ALPHABETIC ARRAY SORT                                SORT1
148    1 CD1                                                                            SORT1
149    1 CD1      ENTRY POINT INTO SUBROUTINE SORT2                                     SORT1
150    1 CD1                                                                            SORT1
151    1 CD1      THIS ROUTINE WAS EXTRACTED FORM THE MDAS SUBMONITOR PROGRAM FOR USE IN SORT1
152    1 CD1      PDLIST.  DOCUMENTATION MAY BE FOUND IN 'LEVEL II MDAS PROTOTYPE       SORT1
153    1 CD1      MONITOR PROGRAM DOCUMENT (PART II),' TRW NOTE NO. 74-FMT-937,         SORT1
154    1 CD1      14 JUNE 1974.                                                         SORT1
155    1 CD1                                                                            SORT1
156    1 C*********                                                                     SORT1
```

```
 SYMBOL DEFINITION TABLE
 :BEGIN ::      86
 :END?  ::      91
 :FSTWRD      133
  POLIST       33
 :SECIND:      88
 :SEP   ::     89
 :TERM  ::     97

 OXOT F.POLIST
```

5.0 FDS EXECUTIVE DETAILED LOGIC FLOW

A directory listing the major programs and subroutines in alphabetical order is presented initially. The detailed logic flow of each then follows in alphabetical order.

```
1 *                                                                 DIRECT
1 * NAME        DEFINITION                                          DIRECT
1 *                                                                 DIRECT
1 *                                                                 DIRECT
1 *                                                                 DIRECT
1 * XA      ATTENTION FUNCTION                                      DIRECT
1 * XATTN         FDS ATTENTION TASK                                DIRECT
1 *                                                                 DIRECT
1 * XB      BATCH JOB CREATION                                      DIRECT
1 *                                                                 DIRECT
1 * XC      CONFIGURATION PROGRAMS                                  DIRECT
1 * XCFNG         FDS INITIALIZATION                                DIRECT
1 *                                                                 DIRECT
1 * XD      DIRECTIVES                                              DIRECT
1 * XDCLD         INTERFACE ROUTINE FOR NON DATA BASE DIRECTIVES    DIRECT
1 * XDCLE         DIRECTIVE HANDLER FOR CLEAR                       DIRECT
1 * XDCLF         INTERFACE ROUTINE FOR DATA BASE ROUTINES          DIRECT
1 * XDCLL         INTERFACE ROUTINE FOR UTDB DIRECTIVES             DIRECT
1 * XDCOP         DIRECTIVE HANDLER FOR COPY                        DIRECT
1 * XDDBA         PDB/MDB ADDITION TO XPDB                          DIRECT
1 * XDDED         PDB/MDB DELETION FROM XPDB (XDDBA EP)             DIRECT
1 * XDDBV         PDB/MDB VERIFICATION IN XPDB (XDDBA EP)           DIRECT
1 * XDELE         DIRECTIVE HANDLER FOR DELETE                      DIRECT
1 * XDLIS         DECODES LIST TO BE STORED/RESTORED                DIRECT
1 * XDLST         DIRECTIVE HANDLER FOR LIST                        DIRECT
1 * XDOFF         DIRECTIVE HANDLER FOR OFF                         DIRECT
1 * XDRDE         CREATES AWA ELEMENTS FROM UTDB/MDB                DIRECT
1 * XDRDF         CREATES DRDE FILES FROM UTDB/MDB                  DIRECT
1 * XDREC         DIRECTIVE HANDLER FOR RECALL                      DIRECT
1 * XDREN         DIRECTIVE HANDLER FOR RENAME                      DIRECT
1 * XDREG         AWA MANAGEMENT REQUEST ROUTINE FOR RESTORE        DIRECT
1 * XDRES         DIRECTIVE HANDLER FOR RESTORE                     DIRECT
1 * XDSAV         DIRECTIVE HANDLER FOR SAVE                        DIRECT
1 * XDSTA         USAGE AND STATISTICS SUMMARY                      DIRECT
1 * XDSTO         DIRECTIVE HANDLER FOR STORE                       DIRECT
1 * XDTOC         DIRECTIVE HANDLER FOR TOC                         DIRECT
1 * XDUFT         CREATES UTDF FROM AWA                             DIRECT
1 *                                                                 DIRECT
1 * XE      EXECUTIVE PROGRAM MAIN LOGIC                            DIRECT
1 * XECAL         INITIALIZATION SEGMENT INTERFACE ROUTINE          DIRECT
1 * XEIND         DIRECTIVE INITIALIZATION                          DIRECT
1 * XEINE         EXECUTIVE INITIALIZATION                          DIRECT
1 * XEINI         INTERFACE TABLE EDITOR INITIALIZATION             DIRECT
1 * XEINS         SEQUENCE TABLE EDITOR INITIALIZATION              DIRECT
1 * XEINT         INTERFACE TABLE LITERAL AREA INIT ROUTINE         DIRECT
1 * XEINX         EXECUTION CONTROLLER INITIALIZATION               DIRECT
1 * XELDS         PROVIDES 'CALL' AND 'RETURN' FOR A SEGMENT        DIRECT
1 * XESCN         PARAMETRIC SCAN CLEAN UP                          DIRECT
1 * XEXEC         FDS EXECUTIVE MAIN PROGRAM                        DIRECT
1 *                                                                 DIRECT
1 * XI      INTERFACE TABLE EDITOR (ITE)                            DIRECT
1 * XICHR         ITE ARGUMENT CHARACTERISTICS LISTER               DIRECT
1 * XIDAT         ITE LITERAL DATA PROCESSOR                        DIRECT
1 * XIEXT         ITE ARGUMENT CHARACTERISTICS EXTRACTOR            DIRECT
1 * XILIT         ITE LITERAL DATA COMPACTER                        DIRECT
1 * XILSD         ITE LIST DATA ENTRIES                             DIRECT
1 * XILSS         ITE SYMBOLIC STRING LISTER                        DIRECT
1 * XILST         ITE LIST DIRECTIVE PROCESSOR                      DIRECT
1 * XIMIX         ITE MAIN LOGIC                                    DIRECT
```

```
61    1 * XIMPT        ITE USER RESPONSE PROCESSOR                                                    DIRECT
62    1 * XINTE        ITE OVERLAY INTERFACE ROUTINE                                                  DIRECT
63    1 * XIPAR        ITE PARAMETER DATA PROCESSOR                                                   DIRECT
64    1 * XIPMT        ITE PROMPT DIRECTIVE PROCESSOR                                                 DIRECT
65    1 * XIPRM        ITE PROMPT CONSTRUCTOR                                                         DIRECT
66    1 * XISUB        ITE SUBSCRIPT PROCESSOR                                                        DIRECT
67    1 *                                                                                             DIRECT
68    1 * XL      LIBRARY MAINTENANCE PROGRAM                                                       DIRECT
69    1 * XLCDB        CREATES A RDB/PDB DATA BASE FILE                                               DIRECT
70    1 * XLDBF        MAIN DATA BASE FILE SEGMENT                                                    DIRECT
71    1 * XLDEL        DELETES A PROCESSOR FROM LIBRARY DIRECTORY                                     DIRECT
72    1 * XLIFL        CREATES A DEFAULT INTERFACE TABLE                                              DIRECT
73    1 * XLINS        DISPLAYS PARAMETER SPECIFICATIONS INSTRUCTION                                  DIRECT
74    1 * XLINT        MAIN LOGIC FOR DEFAULT VALUES                                                  DIRECT
75    1 * XLMAN        MAIN MAINTENANCE PROGRAM                                                       DIRECT
76    1 * XLMOD        MODIFIES A PROCESSOR INTERFACE TABLE                                           DIRECT
77    1 * XLMSG        MAINTAINS FDS MESSAGE FILE                                                     DIRECT
78    1 * XLPCR        CREATES RDB/PDB LOG FILE                                                       DIRECT
79    1 * XLPFL        CREATES A PROCESSOR PROMPT TABLE                                               DIRECT
80    1 * XLPMO        MODIFIES RDB/PDB LOG FILE                                                      DIRECT
81    1 * XLPRM        CREATES SYSTEM PROMPT FILES                                                    DIRECT
82    1 * XLPRO        ADDS A PROCESSOR TO LIBRARY DIRECTORY AND ADDS DEFAULT                         DIRECT
83    1 *              INTERFACE TABLE AND PROMPT FILE                                                DIRECT
84    1 * XLSPS        ADDS SPECS FOR DEFAULT INTERFACE TABLE                                         DIRECT
85    1 *                                                                                             DIRECT
86    1 * XM      MANAGER                                                                           DIRECT
87    1 * XMAFR        AWA FREE SPACE                                                                 DIRECT
88    1 * XMAGT        AWA GET SPACE                                                                  DIRECT
89    1 * XMAMG        AWA MANAGEMENT PROCESSING (XMGR EP)                                            DIRECT
90    1 * XMAPK        GENERAL PACK INTERFACE (XMPCK EP)                                              DIRECT
91    1 * XMAWA        AWA AND CONTROL DATA MAPPING                                                   DIRECT
92    1 * XMDAL        DWA ALLOCATION (XMDWA EP)                                                      DIRECT
93    1 * XMDDA        DWA DEALLOCATION (XMDWA EP)                                                    DIRECT
94    1 * XMDIN        DWA INITIALIZATION (XMDWA EP)                                                  DIRECT
95    1 * XMDRT        DWA RETRIEVE (READ) (XMDWA EP)                                                 DIRECT
96    1 * XMDST        DWA STORE (WRITE) (XMDWA EP)                                                   DIRECT
97    1 * XMDWA        DWA MANAGEMENT                                                                 DIRECT
98    1 * XMGR         FDS MANAGER                                                                    DIRECT
99    1 * XMPAW        POST AND WAIT INTERFACE ROUTINE (XMGR EP)                                      DIRECT
100   1 * XMPCK        AWA COMPRESS                                                                   DIRECT
101   1 * XMPK1        PHASE 1 COMPRESS (PURGE TO DWA) (XMPCK EP)                                     DIRECT
102   1 * XMPK2        PHASE 2 COMPRESS (TOC COMPRESS) (XMPCK EP)                                     DIRECT
103   1 * XMPK3        PHASE 3 COMPRESS (SPACE REORDER) (XMPCK EP)                                    DIRECT
104   1 * XMTFN        TOC LOOK-UP ROUTINE                                                            DIRECT
105   1 * XMSST        SEQUENCE LOCATION COUNTER RESET                                                DIRECT
106   1 * XMXQT        SEQUENCE TABLE EXECUTION                                                       DIRECT
107   1 *                                                                                             DIRECT
108   1 * XP      PROCESSOR SERVICES                                                                DIRECT
109   1 * XPATR        PROCESSOR PARAMETER ATTRIBUTE RETRIEVAL                                        DIRECT
110   1 * XPGET        PROCESSOR PARAMETER RETRIEVAL                                                  DIRECT
111   1 * XPGTI        INDEXED PARAMETER RETRIEVAL                                                    DIRECT
112   1 * XPPUT        PROCESSOR PARAMETER STORAGE (XPGET EP)                                         DIRECT
113   1 * XPPTI        INDEXED PARAMETER STORAGE (XPGTI EP)                                           DIRECT
114   1 * XPRDM        PROMPTS USER,READS RESPONSE,RETURNS ENCODED BUFFER                             DIRECT
115   1 * XPRDS        CALLS XPRDM,DECODES BUFFER,RETURNS RESPONSE IN DATA AREA                       DIRECT
116   1 * XPRLQ        PROCESSOR SERVICE FOR AWA ACCESS                                               DIRECT
117   1 * XPXIT        PROCESSOR TERMINATION ROUTINE                                                  DIRECT
118   1 *                                                                                             DIRECT
119   1 * XR      EXECUTIVE SERVICES                                                                DIRECT
```

```
120   1 * XRBIT       MULTIPLE WORD BIT STRING BIT CLEAR/SET                                  DIRECT
121   1 * XRCPR       COMPARE ARRAYS                                                          DIRECT
122   1 * XRD18       DOUBLE PRECISION TO ASCII CONVERSION                                    DIRECT
123   1 * XREQ        AWA MANAGEMENT REQUEST ROUTINE                                          DIRECT
124   1 * XREXT       EXTRACTS A VARIABLE LENGTH FIELD FROM A WORD                            DIRECT
125   1 * XRE14       CONVERT A WORD TO ASCII IN 1PE14 FORMAT                                 DIRECT
126   1 * XRI6        BINARY INTEGER TO ASCII CONVERSION ROUTINE                              DIRECT
127   1 * XRLCK       XVSTB RN LOCK                                                           DIRECT
128   1 * XRLOC       RETURN 16-BIT ADDRESS OF ARGUMENT                                       DIRECT
129   1 * XRMOV       MOVES WORDS FROM ARRAY1 TO ARRAY2                                       DIRECT
130   1 * XRMSG       FDS EXECUTIVE MESSAGE ROUTINE                                           DIRECT
131   1 * XRNXB       MULTIPLE WORD BIT STRING BIT SEARCH                                     DIRECT
132   1 * XRO6        CONVERT A WORD TO ASCII IN O6 FORMAT                                    DIRECT
133   1 * XRPCK       PACKS CHARACTERS FROM R1 TO A2 FORMAT                                   DIRECT
134   1 * XRQFN       FILE NAME QUALIFICATION                                                 DIRECT
135   1 * XRSET       SETS A VARIABLE LENGTH FIELD INTO A WORD                                DIRECT
136   1 * XRSFL       SHIFT A WORD LEFT LOGICALLY                                             DIRECT
137   1 * XRSFR       SHIFT A WORD RIGHT LOGICALLY (XRSFL EP)                                 DIRECT
138   1 * XRULK       XVSTB RN UNLOCK (XRLCK EP)                                              DIRECT
139   1 * XRUNQ       FILE NAME DEQUALIFY                                                     DIRECT
140   1 * XRUPK       REMOVES BLANKS AND UNPACKS FROM A2 TO R1 FORMAT                         DIRECT
141   1 * XR1SP       REMOVE DUPLICATE BLANKS FROM A2 STRING                                  DIRECT
142   1 *                                                                                     DIRECT
143   1 * XS      SEQUENCE TABLE EDITOR                                                       DIRECT
144   1 * XSCAN       SEQUENCE TABLE EDITOR (STE) DIRECTIVE SCANNER                           DIRECT
145   1 * XSDEL       STE DELETE DIRECTIVE PROCESSOR                                          DIRECT
146   1 * XSENT       STE ENTRY PROCESSOR                                                     DIRECT
147   1 * XSEQE       SEQUENCE TABLE EDITOR MAIN ROUTINE                                      DIRECT
148   1 * XSLIS       STE LIST DIRECTIVE PROCESSOR                                            DIRECT
149   1 * XSLST       SEQUENCE TABLE LIST ROUTINE                                             DIRECT
150   1 * XSNPT       STE INPUT PROCESSOR                                                     DIRECT
151   1 * XSNUM       STE NUMBER DIRECTIVE PROCESSOR                                          DIRECT
152   1 * XSPCK       STE TABLE COMPACTER                                                     DIRECT
153   1 * XSPMT       STE PROMPT DIRECTIVE PROCESSOR                                          DIRECT
154   1 * XSPRM       STE PROMPT CONSTRUCTOR                                                  DIRECT
155   1 *                                                                                     DIRECT
156   1 * XT      TERMINAL COMMUNICATIONS                                                     DIRECT
157   1 * XTCOM        PROMPTS USER,READS RESPONSE,CALLS XTLAN AND XTPRM                       DIRECT
158   1 * XTLAN       CONVERTS ASCII USER'S RESPONSE TO TOKENS                                DIRECT
159   1 * XTPRM       HANDLES EXTENDED PROMPTING REQUESTS                                     DIRECT
160   1 *                                                                                     DIRECT
161   1 * XU      UTILITY (SOFTWARE AIDS)                                                     DIRECT
162   1 * XUDBG       ON-LINE SNAP AND MEMORY MODIFICATION ROUTINE                            DIRECT
163   1 * XUDMP       SYSTEM RESIDENT PARTITION DUMP (XVABN EP)                               DIRECT
164   1 * XUDPF       FILE MANAGER FILE DUMP PROGRAM                                          DIRECT
165   1 * XUDPL       OCTAL AND ASCII DUMP LINE FORMAT                                        DIRECT
166   1 * XUFMT       DUMP FORMATTER                                                          DIRECT
167   1 *                                                                                     DIRECT
168   1 * XV      SYSTEM SERVICES                                                             DIRECT
169   1 * XVABN        FDS ABEND (SEE XUDMP)                                                   DIRECT
170   1 * XVPAW       FDS COMMUNICATIONS SERVICES (POST AND WAIT)                             DIRECT
171   1 *                                                                                     DIRECT
172   1 * XX      EXECUTION CONTROL                                                           DIRECT
173   1 * XXAUT        AUTOMATIC MODE                                                          DIRECT
174   1 * XXCNT        EXECUTION CONTROL MAIN PROGRAM                                          DIRECT
175   1 * XXDEC       DECODES USER RESPONSE IN MANU AND SEMI                                  DIRECT
176   1 * XXDEF       READS IN DEFAULT INTERFACE TABLE IF NEEDED                              DIRECT
177   1 * XXEXE       EXECUTES &SEQTAB AND HANDLES ERROR CONDITIONS                           DIRECT
178   1 * XXMAN       MANUAL MODE                                                             DIRECT
```

```
179   1 *  XXSEM        SEMI - AUTOMATIC MODE                                                    DIRECT
180   1 *  XXSTO        STORE SEQUENCE TABLE IN &SEQTAB                                          DIRECT
181   1 *  XXTMP        TEMPORARY EXECUTION OF ONE ENTRY WITH &INTAB                             DIRECT
182   1 *                                                                                        DIRECT
183   1 *  XZ       UTILITY PROCESSORS                                                          DIRECT
184   1 *  ASSGN        ASSIGN PROCESSOR                                                         DIRECT
185   1 *  DBDSP        DATA BOX DISPLAY PROCESSOR                                               DIRECT
186   1 *  DEFIN        DEFINE PROCESSOR                                                         DIRECT
187   1 *  DO           CONDITIONAL LOOP IN SEQUENCE TABLE                                       DIRECT
188   1 *  ELSE         EXECUTION POINT FOR FALSE IF CONDITION                                   DIRECT
189   1 *  ENDIF        TERMINATES AN IF STRUCTURE                                               DIRECT
190   1 *  ENDDO        TERMINATES A DO LOOP STRUCTURE                                           DIRECT
191   1 *  ENDSC        END SCAN PROCESSOR                                                       DIRECT
192   1 *  IF           CONDITIONAL EXECUTION OF SEQUENCE TABLE ENTRIES                          DIRECT
193   1 *  SCAN         SCAN PROCESSOR                                                           DIRECT
194   1 *  XZCHR        CHARACTER OBJECT STORE FOR ASSGN                                         DIRECT
195   1 *  XZDFT        FIND ANY TOKEN IN A SYMBOLIC STRING                                      DIRECT
196   1 *  XZDIN        DATA BOX DISPLAY INPUT PROCESSOR                                         DIRECT
197   1 *  XZDMK        DATA BOX DISPLAY CONSTRAINT MASKER                                       DIRECT
198   1 *  XZDOT        DATA BOX DISPLAY OUTPUT ROUTINE                                          DIRECT
199   1 *  XZDP1        DATA BOX DISPLAY PASS 1 PROCESSOR                                        DIRECT
200   1 *  XZDP2        DATA BOX DISPLAY PASS 2 PROCESSOR                                        DIRECT
201   1 *  XZEVL        PERFORMS EVALUATION BETWEEN TWO REAL NUMBERS                             DIRECT
202   1 *  XZFCL        FIND PROCESSOR CLASS NUMBER                                              DIRECT
203   1 *  XZFNC        FUNCTIONAL OPERATIONS FOR ASSGN                                          DIRECT
204   1 *  XZFRE        FREE OBJECT STORE FOR ASSGN                                              DIRECT
205   1 *  XZFXD        FIXED OBJECT STORE FOR ASSGN                                             DIRECT
206   1 *  XZISP        REMOVE DUPL. BLANKS & BLANK FILL                                         DIRECT
207   1 *  XZLSS        SYMBOLIC STRING SYNTAX ERROR LISTER                                      DIRECT
208   1 *  XZMSG        FDS PROCESSOR MESSAGE ROUTINE                                            DIRECT
209   1 *  XZOPR        MATH OPERATIONS FOR ASSGN                                                DIRECT
210   1 *  XZPCS        DATA CONVERSION AND STORAGE FOR ASSGN                                    DIRECT
211   1 *  XZPS1        PASS 1 SUBROUTINE FOR ASSGN PROCESSOR                                    DIRECT
212   1 *  XZPS2        PASS 2 SUBROUTINE FOR ASSGN PROCESSOR                                    DIRECT
213   1 *  XZRET        DATA RETRIEVAL FOR ASSGN                                                 DIRECT
214   1 *  XZSCH        SEARCHES SEQUENCE TABLE FOR IF STRUCTURES                                DIRECT
215   1 *  XZSYM        SYMBOL TABLE INTERFACE FOR ASSGN                                         DIRECT
216   1 *  XZSYT        SYMBOL TABLE MAINTENANCE                                                 DIRECT
217   1 *                                                                                        DIRECT
```

```
2    1 BEGIN  XATTN                                                                  XATTN
3    2 *D1                FUNCTION                                                   XATTN
4    2 *D1         FDS ATTENTION TASK-                                               XATTN
5    2 *D1        USER MAY REQUEST THE CURRENT STATUS OF FDS                         XATTN
6    2 *D1        FOR THE TERMINAL, TO TERMINATE CURRENT PROCESSOR,                  XATTN
7    2 *D1        OR (IF FDS HAS TERMINATED) TO TEAR DOWN THE                        XATTN
8    2 *D1        FDS STRUCTURE FOR THIS TERMINAL.                                   XATTN
9    2 *D1                                                                           XATTN
10   2 *D5                NOTES                                                      XATTN
11   2 *D5         THE ID FOR XATTN IS CONNECTED TO THE USER'S                       XATTN
12   2 *D5        TERMINAL AT FDS SIGN-ON BY USING THE EQT                           XATTN
13   2 *D5        FOR THE DEVICE. THE FUNCTION IS DISCONNECTED                       XATTN
14   2 *D5        AT FDS SIGN-OFF.                                                   XATTN
15   2 *D5                                                                           XATTN
```

```
17  2     SAVE EQT ADDRESS(IN BREG ON ENTRY)                                                                                    XATTN
18  2     CALL EQLU(BREG)  GET LU IN ASCII & BINARY                                                                             XATTN
19  2     STARTSEARCH UNTIL LAST STATUS TABLE ENTRY                                                                             XATTN
20  2     EXITIF STBLU EQ LU                                                                                                    XATTN
21  3        SET STB ENTRY ADDRESS                                                                                              XATTN
22  2     ENDLOOP                                                                                                               XATTN
23  3        SET STB ENTRY TO ZERO                                                                                              XATTN
24  2     ENDSEARCH                                                                                                             XATTN
25  2     IF STB ENTRY FOUND, THEN                                                                                              XATTN
26  3        GET MANAGER'S ID ADDRESS(STBMG)                                                                                    XATTN
27  3        IF MANAGER IS DORMANT, THEN                                                                                        XATTN
28  4           WRITE '***XA01- MANAGER HAS TERMINATED;                                                                         XATTN
29  4           REPLY TO CONTINUE TERMINATION:'                                                                                 XATTN
30  4           READ(LU) ** WAIT FOR REPLY **                                                                                   XATTN
31  4           LOCK ON THE FDS TABLE RESOURCE                                                                                  XATTN
32  4           CALL $LIBR    DISABLE                                                                                           XATTN
33  4           IF STBEX(EXECUTIVE ADDRESS .NE. 0, THEN                                                                         XATTN
34  5              IF STBAT(CURRENT) .NE. STBEX, THEN                                                                           XATTN
35  6                 IF CURRENT AT IS NOT DORMANT AND BACK CHAIN POINTS TO OLD XMGR, THEN                                       XATTN
36  7                    FIND BOTTOM AT                                                                                         XATTN
37  7                    DO UNTIL NEXT-AT .EQ. STBMG(MANGER)                                                                    XATTN
38  8                       CALCULATE NEXT-AT FROM BOTTOM'S FATHER ID NUMBER                                                    XATTN
39  8                       CLEAR BOTTOM'S WAIT BIT & FATHER ID NUMBER                                                          XATTN
40  8                       CLEAR NEXT-AT'S PARM ONE(P1)                                                                        XATTN
41  8                       CALL $LIBX    ENABLE                                                                                XATTN
42  8                       CALL MESSS 'OFF,BOTTOM'                                                                             XATTN
43  8                       CALL $LIBR    DISABLE                                                                               XATTN
44  8                       SET BOTTOM TO NEXT-AT                                                                               XATTN
45  7                    ENDDO                                                                                                  XATTN
46  6                 ENDIF                                                                                                     XATTN
47  5              ENDIF                                                                                                        XATTN
48  5              CALL $LIST    MAKE EXEC DORMANT                                                                              XATTN
49  5              CLEAR EXEC'S ID & STBEX                                                                                      XATTN
50  4           ENDIF                                                                                                           XATTN
51  4           DECREMENT NUMBER ACTIVE(STBAC)                                                                                  XATTN
52  4           GET EQT ADDRESS                                                                                                 XATTN
53  4           RESTORE INTERRUPT HANDLER(FROM STBEQ)                                                                           XATTN
54  4           CLEAR STBEQ                                                                                                     XATTN
55  4           CLEAR MANAGER'S ID, STBMG, & STBLU                                                                              XATTN
56  4           ENABLE....(VIA A JMP TO $XEC(DISPATCHER))                                                                       XATTN
57  4           RELEASE EXEC'S AND PROCESSOR'S CLASS NUMBERS                                                                    XATTN
58  4           CLEAR LOCK ON FDS TABLE                                                                                         XATTN
59  3        ELSE      ** MANAGER IS STILL ALIVE **                                                                           XATTN
60  4           WRITE '***XA02- USER INITIATED INTERRUPT'                                                                       XATTN
61  4           WRITE 'ENTER REQUEST- KILL(K). STATUS(S),OR RETURN(BLANK)'                                                      XATTN
62  4           READ (LU) REQUEST                                                                                               XATTN
63  4           IF REQUEST IS KILL OR K, THEN                                                                                   XATTN
64  5              PERFORM XAKILL                                                                                               XATTN
65  4           ELSE                                                                                                            XATTN
66  5              IF REQUEST IS STATUS OR S, THEN                                                                              XATTN
67  6                 PERFORM XASTAT                                                                                            XATTN
68  5              ENDIF                                                                                                        XATTN
69  4           ENDIF                                                                                                           XATTN
70  3        ENDIF                                                                                                              XATTN
71  2     ELSE                                                                                                                  XATTN
72  3        WRITE '***XA03- ERROR LU IS NOT SIGNED-ON TO FDS'                                                                  XATTN
73  2     ENDIF                                                                                                                 XATTN
74  2     WRITE '***XA04 FDS ATTENTION FUNCTION TERMINATING'                                                                    XATTN
75  1 END    XATTN                                                                                                             XATTN
```

```
 77    1 BEGIN  XASTAT            PRODUCE A FDS STATUS REPORT                                                    XATTN
 78    2    CALL SLIBR     DISABLE                                                                               XATTN
 79    2    GET CURRENT-TIME FROM STIME                                                                          XATTN
 80    2    GET MANAGER'S ADDRESS FROM STBMG                                                                     XATTN
 81    2    MOVE NAME, STATUS, PARTITION, & PRIORITY                                                             XATTN
 82    2    GET EXECUTIVE'S ADDRESS FROM STBEX                                                                   XATTN
 83    2    MOVE NAME, STATUS, PARTITION, & PRIORITY                                                             XATTN
 84    2    GET CURRENT AT FROM STBAT                                                                            XATTN
 85    2    MOVE NAME, STATUS, PARTITION, & PRIORITY                                                             XATTN
 86    2    PERFORM XABTM(CURRENT)   FIND BOTTOM AT                                                              XATTN
 87    2    SET BOTTOM TO CURRENT                                                                                XATTN
 88    2    DO WHILE FATHER-ID NE ZERO                                                                           XATTN
 89    3       GET FATHER-ID FROM CURRENT                                                                        XATTN
 90    3       CALCULATE NEXT                                                                                    XATTN
 91    3       IF MAX ENTRIES HAVE NOT BEEN PROCESSED, THEN USE NEXT TO                                          XATTN
 92    4          MOVE NAME, STATUS, PARTITION, & PRIORITY                                                       XATTN
 93    3       ENDIF                                                                                             XATTN
 94    3       IF NEXT IS THE MANAGER, THEN                                                                      XATTN
 95    4          SET CURRENT AS TOP                                                                             XATTN
 96    3       ENDIF                                                                                             XATTN
 97    3       SET CURRENT TO NEXT                                                                               XATTN
 98    2    ENDDO                                                                                                XATTN
 99    2    IF TOP EQ ZERO, THEN                                                                                 XATTN
100    3       SET TOP TO CURRENT                                                                                XATTN
101    3       USE TOP TO MOVE NAME, PARTITION, & PRIORITY                                                       XATTN
102    3       SET STATUS TO 'IN USE' OCTAL 17                                                                   XATTN
103    2    ENDIF                                                                                                XATTN
104    2    CALL SLIBX     ENABLE                                                                                XATTN
105    2    WRITE FIRST SET OF HEADERS                                                                           XATTN
106    2    SET TOP AS REPORT DATA                                                                               XATTN
107    2    WRITE REPORT LINE                                                                                    XATTN
108    2    SET MANAGER AS REPORT DATA                                                                           XATTN
109    2    WRITE REPORT LINE                                                                                    XATTN
110    2    SET EXECUTIVE AS REPORT DATA                                                                         XATTN
111    2    WRITE REPORT LINE                                                                                    XATTN
112    2    SET CURRENT AS REPORT DATA                                                                           XATTN
113    2    WRITE REPORT LINE                                                                                    XATTN
114    2    WRITE INTERMEDIATE HEADERS                                                                           XATTN
115    2    DO UNTIL MAX ENTRIES OR NO MORE DATA                                                                 XATTN
116    3       WRITE REPORT LINE                                                                                 XATTN
117    3       SET NEXT REPORT DATA                                                                              XATTN
118    2    ENDDO                                                                                                XATTN
119    1 END    XASTAT                                                                                           XATTN
120    1 SAMPLE REPORT * * * * * * * * * * * * * * * * * * * * * *                                               XATTN
121    1 #                                                                                                       XATTN
122    1 #        FDS STATUS FOR LU 10 HH:MM:SS  360                                                             XATTN
123    1 #            NAME  PRIOR PART#   STATUS                                                                 XATTN
124    1 #                                                                                                       XATTN
125    1 TOP AT-     PROCA   922    3  GENERAL WAIT                                                             XATTN
126    1 MANAGER-    XMGNN    40    5  GENERAL WAIT                                                             XATTN
127    1 EXECUTIVE-  XEXNN    80    3  GENERAL WAIT                                                             XATTN
128    1 CURRENT AT- PROCD 11311    4  GENERAL WAIT                                                             XATTN
129    1 #                                                                                                       XATTN
130    1 #BACK CHAIN (UP TO 8) FROM BOTTOM VIA FATHER-ID                                                         XATTN
131    1 #                                                                                                       XATTN
132    1 PROCE 32767    6  DISC ALLOCATE SUSPEND                                                                XATTN
133    1 PROCD 11311    4  GENERAL WAIT                                                                         XATTN
134    1 PROCC  2060    3  GENERAL WAIT                                                                         XATTN
135    1 PROCB   845    6  GENERAL WAIT                                                                         XATTN
```

```
136    1 PROCA    922    3  GENERAL WAIT                                                  XATTN
137    1 XMGNN     40    5  GENERAL WAIT                                                  XATTN
138    1 SAMPLE REPORT * * * * * * * * * * * * * * * * * * *                              XATTN
```

```
140   1 BEGIN  XAKILL                                                                                                    XATTN
141   2 *   TERMINATE CURRENT FDS FUNCTION                                                                               XATTN
142   2 *   IF MANAGER IS ACTIVE- SET FLAG FOR SEQUENCE TERMINATION                                                      XATTN
143   2 *  ON NEXT RETURN VIA A PAW.                                                                                     XATTN
144   2 *   IF THE EXEC IS ACTIVE-DO NOTHING                                                                            XATTN
145   2 *   IF A PROCESSOR IS ACTIVE- USE RTE MESSS TO OFF THE PROCESSOR                                                 XATTN
146   2    GET MANAGER'S ID ADDRESS(STBMG)                                                                             XATTN
147   2    IF STATUS OF MANAGER IS NOT WAIT, THEN                                                                      XATTN
148   3       SET TERMINATE FLAG IN STB-ENTRY                                                                          XATTN
149   3       WRITE '***XA05 FDS MANAGER SIGNALED TO TERMINATE SEQUENCE'                                               XATTN
150   2    ELSE                                                                                                        XATTN
151   3       IF CURRENT(STBAT) EQ EXEC(STBEX), THEN                                                                   XATTN
152   4          WRITE '***XA06 FDS EXECUTIVE ACTIVE; NO ACTION TAKEN'                                                 XATTN
153   3       ELSE                                                                                                     XATTN
154   4          PERFORM XABTM(CURRENT)  FIND BOTTOM AT                                                                XATTN
155   4          IF BOTTOM AT IS D.RTR OR SMP THEN                                                                     XATTN
156   5             WRITE '**XA08 MANAGER IS WAITING FOR SYSTEM RESOURCES...NO ACTION TAKEN.'                           XATTN
157   4          EXIT XAKIL                                                                                            XATTN
158   4          ELSE                                                                                                  XATTN
159   5             IF RETURNED BOTTOM IS MANAGER THEN                                                                 XATTN
160   6                IF MANAGER IS NOT WAITING ON A PROGRAM THEN                                                     XATTN
161   7                   WRITE '**XA08 MANAGER WAITING FOR SYSTEM RESOURCES...NO ACTION TAKEN.'                       XATTN
162   6                EXIT XAKILL                                                                                     XATTN
163   6                ELSE                                                                                            XATTN
164   7                   SET RETURN PARAMETER TO PROCESSOR ABENDED                                                    XATTN
165   7                   INCREMENT MANAGER SUSPEND ADDRESS PAST SCHEDULE OF PROCESSOR                                 XATTN
166   7                   CALL SLIST TO REACTIVATE MANAGER                                                             XATTN
167   6                ENDIF                                                                                           XATTN
168   5             ENDIF                                                                                              XATTN
169   5             WRITE '**XA07 FDS PROCESSOR 'NAME' SCHEDULED TO ABORT.'                                            XATTN
170   5             IF RETURNED BOTTOM WAS NOT MANAGER THEN                                                            XATTN
171   6                SET NAME IN 'OFF' COMMAND                                                                       XATTN
172   6                CALL MESSS TO 'OFF' THE PROCESSOR                                                               XATTN
173   5             ENDIF                                                                                              XATTN
174   4          ENDIF                                                                                                 XATTN
175   3       ENDIF                                                                                                    XATTN
176   2    ENDIF                                                                                                       XATTN
177   1 END    XAKILL                                                                                                  XATTN
```

```
179    1 BEGIN  XABTM        FIND BOTTOM AT                                  XATTN
180    2    DO WHILE CURRENT IS IN GENERAL WAIT,                             XATTN
181    3       AND WAIT POINTER(P1) HAS A SON ADDRESS,                       XATTN
182    3       AND SONS FATHER ID POINTS TO CURRENT                          XATTN
183    3       SET SON AS CURRENT                                            XATTN
184    2    ENDDO                                                            XATTN
185    2    SET BOTTOM AS CURRENT                                            XATTN
186    1 END    XABTM                                                        XATTN
```

```
  SYMBOL DEFINITION TABLE
  XABTM      179
  XAKILL     140
  XASTAT      77
  XATTN        2

@XQT F.PDLIST
```

```
1 BEGIN XCNFG                                                          XCNFG
2 *                                                                    XCNFG
2 *D1  FDS CONFIGURATION MANAGER                                       XCNFG
2 *D1  INITIATES AN FDS SYSTEM FOR A REQUESTED TERMINAL OR             XCNFG
2 *D1  TERMINATES AN FDS SYSTEM FOR A REQUESTED TERMINAL               XCNFG
2 *D1  INITIATED VIA                                                   XCNFG
2 *D1     RU,FDS,LU,ID,CVA SIZE,PARMS,OPTIONS                         XCNFG
2 *D2  INPUTS                                                          XCNFG
2 *D2     LOGICAL UNIT(LU) FOR THE REQUESTED TERMINAL,                 XCNFG
2 *D2        A PARM TO DENOTE INITIATION OR TERMINATION                XCNFG
2 *D2        A ONE CHARACTER USERID                                    XCNFG
2 *D2        A DEBUG OPTION INDICATOR                                  XCNFG
2 *D2        THE NUMBER OF DVA TRACKS                                  XCNFG
2 *D3  OUTPUTS                                                         XCNFG
2 *D3     INITIATION-                                                  XCNFG
2 *D3        A BLANK ID-SEGMENT WILL BE CONSTRUCTED FOR THE FDS MANAGER,XCNFG
2 *D3        AND FDS EXECUTIVE                                         XCNFG
2 *D3        THE EQT FOR THE REQUESTED LU WILL BE CONNECTED            XCNFG
2 *D3        TO THE FDS ATTENTION TASK                                 XCNFG
2 *D3        THE FDS TABLE(FDSTAB) IN RESIDENT LIBRARY ROUTINE         XCNFG
2 *D3        WILL BE UPDATED TO REFLECT THE INITIATION                  XCNFG
2 *D3     TERMINATION-                                                 XCNFG
2 *D3        THE FDS ID SEGMENTS WILL BE RETURNED TO BLANK STATUS      XCNFG
2 *D3        THE EQT FOR THE LU WILL BE REINSTATED                     XCNFG
2 *D3        THE FDS TABLE(FDSTAB) IN RESIDENT LIBRARY ROUTINE         XCNFG
2 *D3        WILL BE UPDATED FOR THE TERMINATION                       XCNFG
```

```
29    2 *  PARMS=LU,ID,DMA SIZE,PARM(ON OR OFF),OPTIONS                                  XCNFG
30    2    SWITCH INPUT PARMS AROUND SO THAT                                            XCNFG
31    2 *  NOW PARMS=LU,P2(ON OR OFF),ID,OPTS,DMA SIZE                                  XCNFG
32    2 *  FOR COMPATABILITY TO BUILD I.                                                XCNFG
33    2    CALL RMPAR(PARMS)                                                            XCNFG
34    2    IF LU IS .LT. 0, OR                                                          XCNFG
35    3       .GT. LUMAX(1653), OR                                                      XCNFG
36    3       .EQ. 6(PRINTER), OR                                                       XCNFG
37    3       THE DRIVER IS .NE. 0 OR 5, THEN                                           XCNFG
38    3       ISSUE MESSAGE "***XC04 'LU' IS AN INVALID LU"                             XCNFG
39    2    ELSE                                                                         XCNFG
40    3       IF PARM P2 IS OFF THEN                                                    XCNFG
41    4          PERFORM XCOFF   SIGN OFF                                               XCNFG
42    3       ELSE                                                                      XCNFG
43    4          PERFORM XCON    SIGN ON                                                XCNFG
44    3       ENDIF                                                                     XCNFG
45    2    ENDIF                                                                        XCNFG

46    2    :XCEXT                                                                       XCNFG
47    2    CALL  EXEC  PROGRAM TERMINATION                                              XCNFG
48    1 END XCNFG                                                                       XCNFG
```

```
1 BEGIN XCON                                                                    XCON
2 * S:GN ON A USER TO FDS                                                       XCON
2     DO UNTIL VALID USER ID (P3)                                               XCON
3         IF ID NOT A - Z, THEN                                                 XCON
4             WRITE 'XC08 ENTER VALID ID (A - Z)'                               XCON
4             READ RESPONSE                                                     XCON
3         ENDIF                                                                 XCON
2     ENDDO                                                                     XCON
2     IF USER ID IS BEING USED, THEN                                            XCON
3         WRITE "***XC03 LU 'LU' IS CURRENTLY USING ID 'ID'- SIGN ON REJECTED"  XCON
2     EXIT :XCETA                                                               XCON
2     ENDIF                                                                     XCON
2     IF FDS RESOURCE NUMBER NOT DEFINED, THEN                                  XCON
3         CALL  RNRQ (GLOBAL ALLOCATE, LOCAL SET)                               XCON
2     ELSE                                                                      XCON
3         CALL  RNRQ (LOCAL SET)                                                XCON
2     ENDIF                                                                     XCON
2     IF NUMBER SIGNED ON(STBAC) .EQ. MAXIMUM USERS(STBNM), THEN                XCON
3         ISSUE MESSAGE '***XC05 FDS CURRENTLY AT MAX USER'S.                   XCON
2     ELSE                                                                      XCON
3         DO FOR STBNM(NUMBER OF FDS ENTRIES)                                   XCON
4             IF ENTRY'S LU(STBLU) .EQ. REQUESTING LU(P1) THEN                  XCON
5                 ISSUE MESSAGE "***XC06 'LU' IS ALREADY SIGNED ON TO FDS"      XCON
4             EXIT TO :XCETA                                                    XCON
4             ELSE                                                              XCON
5                 IF THIS ENTRY IS AVAILABLE, THEN                              XCON
6                     SET AS CURRENT-ENTRY-ADDRESS                              XCON
5                 ENDIF                                                         XCON
4             ENDIF                                                             XCON
3         ENDDO                                                                 XCON
3 *           BECOME PRIVLEDGED & DISABLED                                      XCON
3         CALL  SLIBR                                                           XCON
3         STARTSEARCH WHILE NUMBER-FOUND .LT. NUMBER-NEEDED                     XCON
4 *     SEARCH ID-SEGMENTS USING KEYWD(1657)                                    XCON
4             IF XEXEC NOT FOUND AND THIS ID .EQ. XEXEC, THEN                   XCON
5                 SET ID ADDRESS OF XEXEC                                       XCON
5                 INCREMENT NUMBER-FOUND                                        XCON
4             ELSE                                                              XCON
5                 IF XMGR NOT FOUND AND THIS ID .EQ. XMGR, THEN                 XCON
6                     SET ID ADDRESS OF XMGR                                    XCON
6                     INCREMENT NUMBER-FOUND                                    XCON
5                 ELSE                                                          XCON
6                     IF XATTN NOT FOUND AND THIS ID .EQ. XATTN, THEN           XCON
7                         SET ID ADDRESS OF XATTN                               XCON
7                         INCREMENT NUMBER-FOUND                                XCON
6                     ELSE                                                      XCON
7                         IF FIRST-BLANK NOT FOUND AND THIS IS A BLANK ID, THEN XCON
8                             SET ID ADDRESS OF FIRST-BLANK                     XCON
8                             INCREMENT NUMBER-FOUND                            XCON
7                         ELSE                                                  XCON
8                             IF SECONU-BLANK NOT FOUND AND THIS IS A BLANK ID, THEN XCON
9                                 SET ID ADDRESS OF SECOND-BLANK                XCON
9                                 INCREMENT COUNT                               XCON
8                             ENDIF                                             XCON
7                         ENDIF                                                 XCON
6                     ENDIF                                                     XCON
5                 ENDIF                                                         XCON
4             ENDIF                                                             XCON
3         EXITIF  THERE ARE NO MORE ID'S                                        XCON
```

```
109   4          CALL  SLIBX     ENABLE                                                    XCON
110   4          ISSUE MESSAGES "***XC07 CANNOT FIND 'NAME' ID-SIGNON TERMINATED"          XCON
111   3       ENDLOOP                                                                      XCON
112   4 *        BUILD ENTRY IN XVSTB                                                      XCON
113   4          SET LU INFO STBLU                                                         XCON
114   4          SET LU IN ASCII INTO STBLA                                                XCON
115   4          SET USER'S ID INTO STBID                                                  XCON
116   4          SET ADDRESS OF FIRST-BLANK INTO STBMG                                     XCON
117   4          SET ADDRESS OF SECOND-BLANK INTO STBEX                                    XCON
118   4          INCREMENT ACTIVE COUNT(STBAC)                                             XCON
119   4 *        BUILD XMGNN & XEXNN                                                       XCON
120   4          MOVE PRIORITY THRU DISC ADDRESS FROM XMGR TO FIRST-BLANK                  XCON
121   4          TURN ON TM BIT                                                            XCON
122   4          SET NAME TO XMGNN                                                         XCON
123   4          MOVE PRIORITY THRU DISC ADDRESS FROM XEXEC TO SECOND-BLANK                XCON
124   4          TURN ON TM BIT                                                            XCON
125   4          SET NAME TO XEXNN                                                         XCON
126   4 *        LINK ATTENTION FUNCTION TO THE USER                                       XCON
127   4          IF LU .NE. 1, THEN                                                        XCON
128   5 *        DO FOR ANY TERMINAL EXCEPT SYSTEM'S CONSOLE                               XCON
129   5             CALCULATE EQT OVERLAY                                                  XCON
130   5             SAVE EQT VALUE IN STBEQ                                                XCON
131   5             SET ID ADDRESS OF XATTN INTO EQT                                       XCON
132   4          ENDIF                                                                     XCON
133   4          SET INPUT PARMS INTO ID OF XMGNN                                          XCON
134   4 *        SCHEDULE XMGNN VIA SLIST                                                  XCON
135   4          CALL  SLIST                                                               XCON
136   4          CALL  SLIBX     ENABLE                                                    XCON
137   4          IF FDS HAS A FATHER, THEN                                                 XCON
138   5             CALL MESSS 'OFF,FATHER'                                                XCON
139   4          ENDIF                                                                     XCON
140   4          ISSUE MESSAGE '***XC01 LU "NN" SIGNED ON TO FDS'                          XCON
141   3       ENDSEARCH                                                                    XCON

142   3       :XCETA                                                                       XCON
143   3       CALL  RNRQ (LOCAL CLEAR)                                                     XCON
144   2    ENDIF                                                                           XCON
145   2    CALL  EXEC TERMINATE                                                            XCON
146   1 END XCON                                                                           XCON
```

```
XCOFF
XCOFF
XCOFF
XCOFF
XCOFF
```

```
148     1 BEGIN XCOFF
149     2 * SIGN OFF A USER ON FDS
150     2 *
151     2 *    WRITE 'FDS SIGN OFF FUNCTION NOT SUPPORTED'
152     1 END XCOFF
```

```
 SYMBOL DEFINITION TABLE
:XCETA :    142
:XCEXT :     46
 XCNFG        2
 XCOFF      148
 XCON        50

@XQT F.PDLIST
```

```
 2    1 *DO       FORTRAN CALLING PROCEDURE                                                    XDCLD
 3    1 *DO    CALL XELDS (XDCLD)                                                              XDCLD
 4    1 *DO                                                                                    XDCLD
 5    1 **********                                                                             XDCLD
 6    1 *D1                                                                                    XDCLD
 7    1 *D1       XDCLD DETERMINES WHICH OF THE FOLLOWING DIRECTIVES WAS                       XDCLD
 8    1 *D1       REQUESTED AND CALLS THE APPROPRIATE HANDLER SUBROUTINE                       XDCLD
 9    1 *D1          LIST                                                                      XDCLD
10    1 *D1                                                                                    XDCLD
11    1 **********                                                                             XDCLD
12    1 *D2       INPUT                                                                        XDCLD
13    1 *D2    XE COMMON - MASSTA (BITS 10-13 CONTAIN A 0 INDEX INTO A LIST OF                 XDCLD
14    1 *D2                  DIRECTIVES)                                                       XDCLD
15    1 *D2                                                                                    XDCLD
16    1 **********                                                                             XDCLD
17    1 *D4       INTERNAL VARIABLES                                                           XDCLD
18    1 *D4    LIST   - ORDERED LIST OF APPROPRIATE HANDLER ADDRESSES                          XDCLD
19    1 *D4                                                                                    XDCLD
20    1 **********                                                                             XDCLD
21    1 *D5       NOTES                                                                        XDCLD
22    1 *D5    USES .ENTR, XDLST, XERTN                                                        XDCLD
23    1 *D5                                                                                    XDCLD
24    1 *D5    XDCLD IS DESIGNED TO BE THE MAIN ROUTINE FOR THE OVERLAY SEGMENT              XDCLD
25    1 *D5    CONTAINING THE REFERENCED DIRECTIVES                                            XDCLD
26    1 *D5                                                                                    XDCLD
27    1 **********                                                                             XDCLD
28    1 *                                                                                      XDCLD
29    1 *                                                                                      XDCLD
30    1 *                                                                                      XDCLD
31    1 *                                                                                      XDCLD
32    1 BEGIN XDCLD                                                                            XDCLD
33    2    EXTRACT DIRECTIVE INDEX FROM MASSTA                                                 XDCLD
34    2    CASE (:LIST:) INDEX                                                                 XDCLD

35    3       :LIST: CALL XDLST                                                                XDCLD
36    2    ENDCASE                                                                             XDCLD
37    2    CALL XERTN TO RETURN FROM SEGMENT                                                   XDCLD
38    1 END XDCLD                                                                              XDCLD
```

```
40   1 *D0      FORTRAN CALLING PROCEDURE                                                                    XDCLF
41   1 *D0   CALL XELDS (XDCLF)                                                                              XDCLF
42   1 *D0                                                                                                   XDCLF
43   1 **********                                                                                            XDCLF
44   1 *D1                                                                                                   XDCLF
45   1 *D1      XDCLF DETERMINES WHICH OF THE FOLLOWING DIRECTIVES WAS                                       XDCLF
46   1 *D1      REQUESTED AND CALLS THE APPROPRIATE HANDLER SUBROUTINE                                       XDCLF
47   1 *D1         TOC                                                                                       XDCLF
48   1 *D1         SAVE                                                                                      XDCLF
49   1 *D1         RECALL                                                                                    XDCLF
50   1 *D1         DELETE                                                                                    XDCLF
51   1 *D1         RENAME                                                                                    XDCLF
52   1 *D1         COPY                                                                                      XDCLF
53   1 *D1         CLEAR                                                                                     XDCLF
54   1 *D1         OFF                                                                                       XDCLF
55   1 *D1                                                                                                   XDCLF
56   1 **********                                                                                            XDCLF
57   1 *D2      INPUT                                                                                        XDCLF
58   1 *D2   XE COMMON - MASSTA (BITS 10-13 CONTAIN A 1-8 INDEX INTO A LIST OF                               XDCLF
59   1 *D2               DIRECTIVES)                                                                         XDCLF
60   1 *D2                                                                                                   XDCLF
61   1 **********                                                                                            XDCLF
62   1 *D4      INTERNAL VARIABLES                                                                           XDCLF
63   1 *D4   LIST   - ORDERED LIST OF APPROPRIATE HANDLER ADDRESSES                                          XDCLF
64   1 *D4                                                                                                   XDCLF
65   1 **********                                                                                            XDCLF
66   1 *D5      NOTES                                                                                        XDCLF
67   1 *D5   USES .ENTR, XDCLE, XDCOP, XDELE, XDOFF, XDREC, XDREN, XDSAV,                                    XDCLF
68   1 *D5        XDTOC, XERTN                                                                               XDCLF
69   1 *D5                                                                                                   XDCLF
70   1 *D5   XDCLF IS DESIGNED TO BE THE MAIN ROUTINE FOR THE OVERLAY SEGMENT                                XDCLF
71   1 *D5   CONTAINING THE REFERENCED DIRECTIVES                                                            XDCLF
72   1 *D5                                                                                                   XDCLF
73   1 **********                                                                                            XDCLF
74   1 *                                                                                                     XDCLF
75   1 *                                                                                                     XDCLF
76   1 *                                                                                                     XDCLF
77   1 BEGIN XDCLF                                                                                           XDCLF
78   2    EXTRACT DIRECTIVE INDEX FROM MASSTA AND DECREMENT BY 1                                             XDCLF
79   2    CASE (:TOC:, :SAVE:, :RECA:, :DELE:, :RENA:, :COPY:, :CLEA:, :OFF:) INDEX                          XDCLF

80   3       :TOC:  CALL XDTOC                                                                               XDCLF

81   3       :SAVE: CALL XDSAV                                                                               XDCLF

82   3       :RECA: CALL XDREC                                                                               XDCLF

83   3       :DELE: CALL XDELE                                                                               XDCLF

84   3       :RENA: CALL XDREN                                                                               XDCLF

85   3       :COPY: CALL XDCOP                                                                               XDCLF

86   3       :CLEA: CALL XDCLE                                                                               XDCLF

87   3       :OFF:  CALL XDOFF                                                                               XDCLF
88   2    ENDCASE                                                                                            XDCLF
89   2    CALL XERTN TO RETURN FROM SEGMENT                                                                  XDCLF
90   1 END XDCLF                                                                                             XDCLF
```

```
 92    1 *D0      FORTRAN CALLING PROCEDURE                                                                    XDCLU
 93    1 *D0   CALL XELDS (XDCLU)                                                                              XDCLU
 94    1 *D0                                                                                                   XDCLU
 95    1 **********                                                                                            XDCLU
 96    1 *D1                                                                                                   XDCLU
 97    1 *D1      XDCLU DETERMINES WHICH OF THE FOLLOWING DIRECTIVES WAS                                       XDCLU
 98    1 *D1      REQUESTED AND CALLS THE APPROPRIATE HANDLER SUBROUTINE                                       XDCLU
 99    1 *D1         STORE                                                                                    XDCLU
100    1 *D1         RESTORE                                                                                  XDCLU
101    1 *D1         UNLOAD                                                                                   XDCLU
102    1 *D1         LOAD                                                                                     XDCLU
103    1 *D1         BATCH                                                                                    XDCLU
104    1 *D1                                                                                                  XDCLU
105    1 **********                                                                                           XDCLU
106    1 *D2      INPUT                                                                                       XDCLU
107    1 *D2   XE COMMON - MASSTA (BITS 10-13 CONTAIN A 9-13 INDEX INTO A LIST OF                             XDCLU
108    1 *D2                  DIRECTIVES)                                                                     XDCLU
109    1 *D2                                                                                                  XDCLU
110    1 **********                                                                                           XDCLU
111    1 *D4      INTERNAL VARIABLES                                                                          XDCLU
112    1 *D4   LIST   - ORDERED LIST OF APPROPRIATE HANDLER ADDRESSES                                         XDCLU
113    1 *D4                                                                                                  XDCLU
114    1 **********                                                                                           XDCLU
115    1 *D5      NOTES                                                                                       XDCLU
116    1 *D5   USES .ENTR, XDSTO, XDRES, XDUNL, XDLOA, XDBAT, XERTN                                           XDCLU
117    1 *D5                                                                                                  XDCLU
118    1 *D5   XDCLU IS DESIGNED TO BE THE MAIN ROUTINE FOR THE OVERLAY SEGMENT                               XDCLU
119    1 *D5   CONTAINING THE REFERENCED DIRECTIVES                                                           XDCLU
120    1 *D5                                                                                                  XDCLU
121    1 **********                                                                                           XDCLU
122    1 *                                                                                                    XDCLU
123    1 *                                                                                                    XDCLU
124    1 *                                                                                                    XDCLU
125    1 *                                                                                                    XDCLU
126    1 BEGIN XDCLU                                                                                          XDCLU
127    2    EXTRACT DIRECTIVE INDEX FROM MASSTA AND DECREMENT BY 7                                            XDCLU
128    2    CASE (:STOR:, :REST:, :UNLO:, :LOAD:, :BATC:) INDEX                                               XDCLU

129    3       :STOR: CALL XDSTO                                                                              XDCLU

130    3       :REST: CALL XDRES                                                                              XDCLU

131    3       :UNLO: CALL XDUNL                                                                              XDCLU

132    3       :LOAD: CALL XDLOA                                                                              XDCLU

133    3       :BATC: CALL XDBAT                                                                              XDCLU
134    2    ENDCASE                                                                                           XDCLU
135    2    CALL XERTN TO RETURN FROM SEGMENT                                                                 XDCLU
136    1 END XDCLU                                                                                            XDCLU
```

```
138   1 CD0       FORTRAN CALLING PROCEDURE                                                    XDCLE
139   1 CD0                                                                                    XDCLE
140   1 CD0    CALL XDCLE                                                                      XDCLE
141   1 CD0                                                                                    XDCLE
142   1 C*********                                                                             XDCLE
143   1 CD1                                                                                    XDCLE
144   1 CD1       XDCLE PURGES ALL DATA FROM THE AWA.  THE ONLY ELEMENTS REMAIN-               XDCLE
145   1 CD1       ING ARE PERMANENT SYSTEM TABLES AND DATA BASE FILES (CHAINS 1                XDCLE
146   1 CD1       & 8)                                                                         XDCLE
147   1 CD1                                                                                    XDCLE
148   1 C*********                                                                             XDCLE
149   1 CD2       INPUT                                                                        XDCLE
150   1 CD2    XE COMMON - CARTRG, FLAGS, LU                                                   XDCLE
151   1 CD2    MANAGER   - AWA HEADER AND TOC (SEE XMAWA)                                      XDCLE
152   1 CD2                                                                                    XDCLE
153   1 C*********                                                                             XDCLE
154   1 CD3       OUTPUT                                                                       XDCLE
155   1 CD3    XE COMMON - REQBUF, REQPTR, COMBUF-SCRATCH                                      XDCLE
156   1 CD3    XB COMMON - SCRATCH BEYOND XB(200)                                              XDCLE
157   1 CD3    XS COMMON - SCRATCH                                                             XDCLE
158   1 CD3    MANAGER   - COMMANDS TO PURGE AWA AND RESTORE CHAINS 0 AND 8                    XDCLE
159   1 CD3                                                                                    XDCLE
160   1 C*********                                                                             XDCLE
161   1 CD4       LOCAL VARIABLES                                                              XDCLE
162   1 CD4    AWA     - AWA HEADER AND TOC RECEIVED FROM MANAGER                              XDCLE
163   1 CD4    HD1     - TOC CHAIN 1 HEAD                                                      XDCLE
164   1 CD4    HD3     - TOC CHAIN 3 HEAD                                                      XDCLE
165   1 CD4    HD8     - TOC CHAIN 8 HEAD                                                      XDCLE
166   1 CD4    NEXT    - INDEX TO NEXT ALLOCATABLE EIGHT WORD ENTRY IN QUEUE                   XDCLE
167   1 CD4    NMAX    - MAXIMUM SIZE OF QUEUE                                                 XDCLE
168   1 CD4    ORG     - BASE REFERENCE ADDRESS FOR AWA ADDRESSES                              XDCLE
169   1 CD4    QUEUE   - BUFFER FOR CONSTRUCTING AWA RESTORE REQUESTS FOR NON-                 XDCLE
170   1 CD4              DELETED ITEMS                                                        XDCLE
171   1 CD4    TOCMAX  - MAXIMUM SIZE OF TOC ACCOMODATABLE BY XDCLE                            XDCLE
172   1 CD4                                                                                    XDCLE
173   1 C*********                                                                             XDCLE
174   1 CD5       NOTES                                                                        XDCLE
175   1 CD5    USES EXEC, IAND, PRTN, PURGE, XREQ, XRI6, XRMOV, XRMSG, XRO6,                   XDCLE
176   1 CD5         XRQFN, XUDRG, XVABN                                                        XDCLE
177   1 CD5                                                                                    XDCLE
178   1 CD5    IN THE EVENT THAT A COMPACTED AWA TOC WILL NOT FIT IN THE ALLOCAT-             XDCLE
179   1 CD5    ED BUFFER SPACE AND ELEMENTS OF CHAIN 1 OR 8 EXTEND BEYOND IT, THE            XDCLE
180   1 CD5    CLEAR FUNCTION WILL NOT BE PERFORMED.                                         XDCLE
181   1 CD5                                                                                  XDCLE
182   1 CD5    SHOULD THE RESTORATION OF CHAIN 1 OR 8 FAIL AFTER THE CLEAR                   XDCLE
183   1 CD5    REQUEST HAS BEEN COMPLETED BY THE MANAGER, FDS WILL BE TERMINATED             XDCLE
184   1 CD5    INORDER TO PROTECT USER DISK FILES.                                           XDCLE
185   1 CD5                                                                                  XDCLE
186   1 C*********                                                                           XDCLE
```

```
188   1 BEGIN XDCLE                                                                  XDCLE
189   2    RETRIEVE TOC                                                              XDCLE
190   2    BUILD REQUEST TO CLEAR AWA                                                XDCLE
191   2    DO UNTIL END OF PERMANENT SYSTEM TABLES CHAIN (CHAIN 1)                   XDCLE
192   2    EXIT TO :ERR24: IF CHAIN POINTS BEYOND TOC BUFFER                         XDCLE
193   2    EXIT TO :ERR48: IF RESTORATION REQUEST QUEUE IS FULL                      XDCLE
194   3       BUILD REQUEST TO REALLOCATE TABLE                                      XDCLE
195   3       REQUEST MANAGER TO RETRIEVE TABLE (HOLD IN SAM)                        XDCLE
196   3       BUILD REQUEST TO STORE TABLE INTO AWA FROM SAM                         XDCLE
197   2    ENDDO                                                                     XDCLE
198   2    DO UNTIL END OF DATA BASE FILES CHAIN (CHAIN 8)                           XDCLE
199   2    EXIT TO :ERR24: IF CHAIN POINTS BEYOND TOC BUFFER                         XDCLE
200   2    EXIT TO :ERR48: IF RESTORATION REQUEST QUEUE IS FULL                      XDCLE
201   3       BUILD REQUEST TO REALLOCATE TOC ENTRY                                  XDCLE
202   2    ENDDO                                                                     XDCLE
203   2    BUILD REQUEST TO TERMINATE LIST                                           XDCLE
204   2    DO UNTIL END OF DRDE CHAIN (CHAIN 3)                                      XDCLE
205   3       IF CHAIN POINTS BEYOND TOC BUFFER                                      XDCLE
206   3       THEN                                                                   XDCLE
207   4          OUTPUT XD13 'TOC TOO LARGE, DRDE PURGE INCOMPLETE'                  XDCLE
208   3       EXIT PURGE LOOP                                                        XDCLE
209   3       ENDIF                                                                  XDCLE
210   3       PURGE FILE                                                             XDCLE
211   2    ENDDO                                                                     XDCLE
212   2    DO FOR EACH BLOCK OF EIGHT REQUESTS                                       XDCLE
213   3       TRANSMIT BLOCK TO MANAGER                                              XDCLE
214   2    EXIT TO :ERR23: IF REQUESTS FAILED                                        XDCLE
215   2    ENDDO                                                                     XDCLE
216   1 EXIT XDCLE                                                                   XDCLE

217   2    :ERR23:                                                                   XDCLE
218   2    DO FROM FAILING REQUEST TO END OF LIST                                    XDCLE
219   3       IF REQUEST TO STORE                                                    XDCLE
220   3       THEN                                                                   XDCLE
221   4          READ SAM TO FREE BUFFER AND CLASS NUMBER                            XDCLE
222   3       ENDIF                                                                  XDCLE
223   2    ENDDO                                                                     XDCLE
224   2    DO UNTIL END OF DATA BASE FILE CHAIN (CHAIN 8)                            XDCLE
225   3       IF FILE IS UTDB (TYPE 1)                                               XDCLE
226   3       THEN                                                                   XDCLE
227   4          CALL PURGE TO DELETE FILE                                           XDCLE
228   3       ENDIF                                                                  XDCLE
229   2    ENDDO                                                                     XDCLE
230   2    PURGE ALL UTDB FILES                                                      XDCLE
231   2    TERMINATE FDS WITH CLEAR FAILURE MESSAGE                                  XDCLE

232   2    :ERR48:                                                                   XDCLE

233   2    :ERR24:                                                                   XDCLE
234   2    DO FOR ALL STORE REQUESTS BUILT                                           XDCLE
235   3       READ SAM TO FREE BUFFER AND CLASS NUMBER                               XDCLE
236   2    ENDDO                                                                     XDCLE
237   1 EXIT XDCLE WITH CLEAR FAILURE MESSAGE                                        XDCLE
238   1 END XDCLE                                                                    XDCLE
```

```
240   1 C*********                                                                                                    XDCOP
241   1 CD0                                                                                                          XDCOP
242   1 CD0     FORTRAN CALLING PROCEDURE                                                                            XDCOP
243   1 CD0                                                                                                          XDCOP
244   1 CD0     CALL XDCOP                                                                                           XDCOP
245   1 CD0                                                                                                          XDCOP
246   1 C*********                                                                                                   XDCOP
247   1 CD1                                                                                                          XDCOP
248   1 CD1     XDCOP PROCESSES A USER REQUEST TO COPY A SEQUENCE TABLE,                                              XDCOP
249   1 CD1     INTERFACE TABLE, DATA ELEMENT, DRDE, UTDB, OR PDB.  ONLY                                             XDCOP
250   1 CD1     THOSE TABLES OR ELEMENTS LOGGED IN THE USER'S AWA CAN BE COPIED.                                     XDCOP
251   1 CD1                                                                                                          XDCOP
252   1 C*********                                                                                                   XDCOP
253   1 CD2                                                                                                          XDCOP
254   1 CD2     INPUT                                                                                                XDCOP
255   1 CD2                                                                                                          XDCOP
256   1 CD2     COMMON XE - CARTRG, COMBUF, QUAL, REQPTR, TOKENS                                                     XDCOP
257   1 CD2                                                                                                          XDCOP
258   1 C*********                                                                                                   XDCOP
259   1 CD3                                                                                                          XDCOP
260   1 CD3     OUTPUT                                                                                               XDCOP
261   1 CD3                                                                                                          XDCOP
262   1 CD3     COMMON XE - COMPTR, REQBUF                                                                           XDCOP
263   1 CD3                                                                                                          XDCOP
264   1 C*********                                                                                                   XDCOP
265   1 CD4                                                                                                          XDCOP
266   1 CD4     INTERNAL VARIABLES                                                                                   XDCOP
267   1 CD4                                                                                                          XDCOP
268   1 CD4    COMMON XS - (1) DATCLS: DATA CLASS CODE (STORED IN LEFT BYTE)                                         XDCOP
269   1 CD4                (2) DBTYPE: TYPE CODE FOR DATA BASE FILES                                               XDCOP
270   1 CD4                (3) I: INDEX                                                                            XDCOP
271   1 CD4                (4) IDCB: EXEC BUFFER                                                                   XDCOP
272   1 CD4              (148) IERR: FILE MANAGER ERROR RETURN                                                     XDCOP
273   1 CD4              (149) IMSG: ERROR MESSAGE NUMBERS                                                         XDCOP
274   1 CD4              (150) NFNAME: NEW FMGR FILE NAME                                                          XDCOP
275   1 CD4              (153) NNMPTR: POINTER TO NEW NAME IN COMBUF                                               XDCOP
276   1 CD4              (154) OFNAME: OLD FMGR FILE NAME                                                          XDCOP
277   1 CD4              (157) ONMPTR: POINTER TO OLD NAME IN COMBUF                                               XDCOP
278   1 CD4              (158) TOCENT: TOC ENTRY FOR DATA BASE RENAME                                              XDCOP
279   1 CD4              (166) EOF: EOF RETURN FROM FMGR                                                           XDCOP
280   1 CD4                                                                                                        XDCOP
281   1 CD4     COMMON XB - (204) IBUFF: USER BUFFER FOR EXEC I/O  (1024)                                            XDCOP
282   1 CD4                 (1244) FMTYPE: FMGR FILE TYPE                                                          XDCOP
283   1 CD4                 (1245) PARM1: FIRST BYTE OF REQPTR                                                     XDCOP
284   1 CD4                 (1246) PARM2: SECOND BYTE OF REQPTR                                                    XDCOP
285   1 CD4                 (1247) PREFIX: PREFIX FOR FILE NAME                                                    XDCOP
286   1 CD4                 (1248) SCRTY: FILE SECURITY CODE (0 OR 88)                                             XDCOP
287   1 CD4                 (1249) SIZE: FILE SIZE IN BLOCKS                                                       XDCOP
288   1 CD4                  (1251) IDCB1: EXEC BUFFER                                                             XDCOP
289   1 CD4                                                                                                        XDCOP
290   1 C*********                                                                                                 XDCOP
291   1 CD5                                                                                                        XDCOP
292   1 CD5     ROUTINES USED -                                                                                    XDCOP
293   1 CD5                                                                                                        XDCOP
294   1 CD5     CLOSE, CREAT, EXEC, OPEN, PURGE, READF, WRITF, XDDBA, XDDBD,                                       XDCOP
295   1 CD5     XREQ, XRMOV, XRMSG, XRQFN, XRSFR                                                                   XDCOP
296   1 CD5                                                                                                        XDCOP
297   1 C*********                                                                                                 XDCOP
```

```
299    1 BEGIN XDCOP                                                                                    XDCOP
300    1 EXIT TO :SYNTAX: IF FIRST TOKENS ARE NOT ",NAME"                                               XDCOP
301    2    SAVE POINTER TO CURRENT NAME                                                                XDCOP
302    2    IF NEXT TOKEN IS HYPHEN, THEN                                                               XDCOP
303    3       DECODE CLASS NAME                                                                        XDCOP
304    2    EXIT TO :CLASER: IF CLASS SPECIFIED IS NOT VALID (S,I,D,F,B)                                XDCOP
305    2    ELSE                                                                                        XDCOP
306    3       SET CLASS TO DATA ELEMENT                                                                XDCOP
307    2    ENDIF                                                                                       XDCOP
308    1 EXIT TO :SYNTAX: IF NEXT TOKENS ARE NOT ",NAME"                                                XDCOP
309    2    SAVE POINTER TO NEW NAME                                                                    XDCOP
310    1 EXIT TO :SYNTAX: IF NEXT TOKEN IS NOT END-OF-MESSAGE                                           XDCOP
311    2    IF CLASS IS DATA BASE OR DRDE, THEN                                                         XDCOP
312    2    EXIT IF NEW NAME IS MORE THAN 4 CHARACTERS   :NAMERR:                                       XDCOP
313    3       IF CLASS IS DATA BASE, THEN                                                              XDCOP
314    3       EXIT TO :NAMERR: IF NEW NAME STARTS WITH DOUBLE EXCLAMATION                              XDCOP
315    3       ENDIF                                                                                    XDCOP
316    3       BUILD AWA MANAGER REQUEST FOR TOC ENTRY FOR CURRENT NAME                                 XDCOP
317    3       BUILD AWA MANAGER REQUEST TO VERIFY NEW NAME                                             XDCOP
318    3       CALL XREQ                                                                                XDCOP
319    2    EXIT TO :TOCERR: IF RETURN INDICATES ERROR ON FIRST REQUEST                                 XDCOP
320    2    EXIT TO :TOCERR: IF NO ERROR RETURNED ON SECOND REQUEST                                     XDCOP
321    3       CALL EXEC TO GET TOC ENTRY                                                               XDCOP
322    3       IF CLASS IS DATA BASE  THEN                                                              XDCOP
323    4          IF TYPE IS PDB, THEN                                                                  XDCOP
324    5             CALL XDDBA TO ADD NEW PDB TO LOG FILE ( XPDB)                                      XDCOP
325    4          EXIT TO :FILERR: IF FMGR ERROR RETURNED                                               XDCOP
326    4          EXIT TO :TOCERR: IF NEW NAME IS DUPLICATE                                             XDCOP
327    4          EXIT TO :MAXERR: IF PDB MAX IS EXCEEDED                                               XDCOP
328    5             SET FILE PREFIX TO  RIGHT BRACKET                                                  XDCOP
329    4          ELSE                                                                                  XDCOP
330    4          EXIT TO :INVLD: IF FILE IS MDB                                                        XDCOP
331    5             SET FILE PREFIX TO *                                                               XDCOP
332    4          ENDIF                                                                                 XDCOP
333    4          SET FMGR FILE TYPE TO 1                                                               XDCOP
334    4          SET SECURITY CODE TO 88                                                               XDCOP
335    3       ELSE                                                                                     XDCOP
336    4          SET FILE PREFIX TO /                                                                  XDCOP
337    4          SET SECURITY CODE = 0                                                                 XDCOP
338    4          SET FMGR FILE TYPE FROM TOC ENTRY                                                     XDCOP
339    4          IF FILE IS TYPE 2, THEN                                                               XDCOP
340    5             STORE RECORD LENGTH FROM TOC ENTRY                                                 XDCOP
341    4          ENDIF                                                                                 XDCOP
342    3       ENDIF                                                                                    XDCOP
343    3       CALL XRQFN TO FORMAT NEW FILE NAME                                                       XDCOP
344    3       CALL CREATE FOR NEW FILE                                                                 XDCOP
345    2    EXIT TO :FILERR: IF FMGR ERROR RETURNED                                                     XDCOP
346    3       CALL XRQFN TO FORMAT OLD FILE NAME                                                       XDCOP
347    3       CALL OPEN FOR OLD FILE                                                                   XDCOP
348    2    EXIT TO :FILERR: IF FMGR ERROR RETURNED                                                     XDCOP
349    2    EXIT TO :TYPERR: IF TYPE IS NOT SAME AS IN TOC                                              XDCOP
350    3       DO UNTIL END-OF-FILE IS READ ON OLD FILE                                                 XDCOP
351    4          CALL READF TO READ RECORD FROM OLD FILE                                               XDCOP
352    3       EXIT IF ERROR DETECTED TO :FILERR:                                                       XDCOP
353    4          CALL WRITE TO WRITE RECORD TO NEW FILE                                                XDCOP
354    3       EXIT IF ERROR DETECTED TO :FILERR:                                                       XDCOP
355    3       END DO                                                                                   XDCOP
356    3       CALL CLOSE FOR OLD FILE                                                                  XDCOP
357    3       CALL CLOSE FOR NEW FILE                                                                  XDCOP
```

```
358  3       BUILD AND ISSUE AWA MANAGER REQUEST TO ALLOCATE TOC ENTRY                    XDCOP
359  2    EXIT TO :TOCERR: IF ERROR IS INDICATED                                         XDCOP
360  2    ELSE                                                                           XDCOP
361  3       BUILD MANAGER REQUEST FOR CURRENT TOC ENTRY                                 XDCOP
362  3       BUILD MANAGER REQUEST FOR DATA RETRIEVAL                                    XDCOP
363  3       CALL XREQ                                                                   XDCOP
364  2    EXIT TO :TOCERR: IF ERROR IS INDICATED                                         XDCOP
365  3       CALL EXEC TO GET TOC ENTRY                                                  XDCOP
366  3       BUILD MANAGER REQUEST TO ALLOCATE NEW TABLE                                 XDCOP
367  3       BUILD MANAGER REQUEST TO STORE TABLE                                        XDCOP
368  3       IF CLASS IS INTERFACE TABLE, THEN                                           XDCOP
369  4          CALL EXEC TO WRITE NEW TABLE NAME TO SAM                                 XDCOP
370  4          BUILD MANAGER REQUEST TO STORE NEW NAME IN TABLE                         XDCOP
371  3       ENDIF                                                                       XDCOP
372  3       CALL XREQ                                                                   XDCOP
373  2    EXIT TO :TOCERR: IF ERROR IS INDICATED                                         XDCOP
374  2    ENDIF                                                                          XDCOP
375  1 EXIT XDCOP                                                                        XDCOP

376  2    :SYNTAX: CALL XRMSG -"SYNTAX ERROR ..." AND EXIT                               XDCOP

377  2    :CLASER: CALL XRMSG -"INVALID CLASS DESIGNATOR ..." AND EXIT                   XDCOP

378  2    :NAMERR: CALL XRMSG -"NEW NAME IS INVALID ..." AND EXIT                        XDCOP

379  2    :MAXERR: CALL XRMSG -"AUTHORIZED LIMIT ..." AND EXIT                           XDCOP

380  2    :INVALD: CALL XRMSG -"MDB CANNOT BE ..." AND EXIT                              XDCOP

381  2    :FILERR: CALL XRMSG -"FILE ACCESS ERROR #.. ON ....." AND EXIT TO :END:        XDCOP

382  2    :TOCERR: CALL XRMSG TO OUTPUT APPROPRIATE MESSAGE AND EXIT TO :END:            XDCOP

383  2    :TYPERR: CALL XRMSG -"INCONSISTENT FILE TYPE ..."                              XDCOP

384  2    :END:                                                                          XDCOP
385  2    IF PDB HAS BEEN LOGGED IN XPDB, THEN                                           XDCOP
386  3       CALL XDDBD TO DELETE PDB FROM XPDB                                          XDCOP
387  2    ENDIF                                                                          XDCOP
388  2    IF A NEW FILE HAS BEEN BUILT, THEN                                             XDCOP
389  3       PURGE NEW FILE                                                              XDCOP
390  3       CLOSE OLD FILE                                                              XDCOP
391  2    ENDIF                                                                          XDCOP
392  1 END XDCOP                                                                         XDCOP
```

```
1 *D0                                                                    XDDBA
1 *D0     FORTRAN CALLING PROCEDURES                                     XDDBA
1 *D0                                                                    XDDBA
1 *D0   CALL XDDBA(NAME,IERR,ISIZE)                                      XDDBA
1 *D0   CALL XDDBV(NAME,IERR,ISIZE)                                      XDDBA
1 *D0   CALL XDDBD(NAME,IERR)                                            XDDBA
1 *D0                                                                    XDDBA
1 **********                                                             XDDBA
1 *D1                                                                    XDDBA
1 *D1   XDDBA, XDDBV, AND XDDBD PROVIDE THE CAPABILITY TO ADD AN MDB OR  XDDBA
1 *D1   PDB TO MDB/PDB DIRECTORY (XPDB), TO VERIFY THAT AN MDB OR PDB IS XDDBA
1 *D1   LOGGED IN XPDB AND RETRIEVE THE SIZE OF THAT MDB/PDB, AND TO     XDDBA
1 *D1   DELETE AN MDB OR PDB FROM XPDB, RESPECTIVELY                     XDDBA
1 *D1                                                                    XDDBA
1 **********                                                             XDDBA
1 *D2                                                                    XDDBA
1 *D2     INPUT                                                          XDDBA
1 *D2   NAME - 2 WORD INTEGER ARRAY CONTAINING THE UNQUALIFIED ASCII     XDDBA
1 *D2          MDB/PDB NAME                                              XDDBA
1 *D2   ISIZE - INTEGER WORD CONTAINING THE MDB/PDB SIZE IN BLOCKS (FOR  XDDBA
1 *D2          XDDBA ONLY)                                               XDDBA
1 *D2                                                                    XDDBA
1 **********                                                             XDDBA
1 *D3                                                                    XDDBA
1 *D3     OUTPUT                                                         XDDBA
1 *D3   IERR - INTEGER WORD CONTAINING RETURN CODE                       XDDBA
1 *D3            <0  FMGR ERROR CODE RETURNED FOR  XPDB ACCESS           XDDBA
1 *D3            =0  NO ERRORS                                           XDDBA
1 *D3            =1  DUPLICATE NAME OR NAME NOT FOUND, AS APPROPRIATE    XDDBA
1 *D3            =2  MDB/PDB LIMIT EXCEEDED                              XDDBA
1 *D3   ISIZE - INTEGER WORD CONTAINING THE MDB/PDB SIZE IN BLOCKS (FOR  XDDBA
1 *D3          XDDBV ONLY)                                               XDDBA
1 *D3                                                                    XDDBA
1 **********                                                             XDDBA
1 *D4                                                                    XDDBA
1 *D4     COMMON USED                                                    XDDBA
1 *D4                                                                    XDDBA
1 *D4   XE  - QUAL                                                       XDDBA
1 *D4         CARTRG                                                     XDDBA
1 *D4   XS  - DCB (WORDS 1-144)                                          XDDBA
1 *D4                                                                    XDDBA
1 **********                                                             XDDBA
1 *D5                                                                    XDDBA
1 *D5     NOTES                                                          XDDBA
1 *D5                                                                    XDDBA
1 *D5   XRULK, XRLCK, .ENTR, OPEN, CLOSE, READF, AND WRITF ARE USED      XDDBA
1 *D5                                                                    XDDBA
1 *D5   WHEN WORKING WITH MDB'S, QUAL SHOULD BE SET TO 77 OCTAL          XDDBA
1 *D5                                                                    XDDBA
1 **********                                                             XDDBA
```

```
445   1 BEGIN XDDBA                                                                                                    XDDBA
446   2    STORE RETURN ADDRESS                                                                                        XDDBA
447   2    CALL .ENTR TO SET UP CALLING ARGUMENTS                                                                      XDDBA
448   2    CALL XRLCK FOR EXCLUSIVE USE OF XPDB                                                                        XDDBA
449   2    CALL OPEN FOR EXCLUSIVE USE OF XPDB                                                                         XDDBA
450   1 EXIT TO :FILERR: IF ERROR RETURNED                                                                             XDDBA
451   2    COMPUTE RECORD NUMBER FOR USER'S DIRECTORY   (QUAL-77B)/2+1                                                 XDDBA
452   2    CALL READF FOR RECORD COMPUTED                                                                              XDDBA
453   1 EXIT TO :FILERR: IF ERROR RETURNED                                                                             XDDBA
454   2    DETERMINE PART OF RECORD TO BE USED                                                                         XDDBA
455   2    IF REQUEST IS FOR ADD, THEN                                                                                 XDDBA
456   2    EXIT TO :MAXERR: IF CURRENT # OF ENTRIES + 1 > MAX ENTRIES                                                  XDDBA
457   3       START SEARCH UNTIL ALL CURRENT ENTRIES ARE TESTED                                                        XDDBA
458   3       EXIT TO :NAMERR: IF ENTRY NAME MATCHES PARAMETER NAME                                                    XDDBA
459   3       ENDLOOP                                                                                                  XDDBA
460   3       ENDSEARCH                                                                                                XDDBA
461   3       STORE NEW NAME AND SIZE IN ENTRY FOLLOWING LAST ENTRY                                                    XDDBA
462   3       INCREMENT # OF CURRENT ENTRIES                                                                           XDDBA
463   2    ELSE                                                                                                        XDDBA
464   3       START SEARCH UNTIL ALL CURRENT ENTRIES, IF ANY, ARE TESTED                                               XDDBA
465   3       EXIT IF ENTRY NAME MATCHES PARAMETER NAME                                                                XDDBA
466   3       ENDLOOP                                                                                                  XDDBA
467   3       EXIT TO :NAMERR:                                                                                         XDDBA
468   3       ENDSEARCH                                                                                                XDDBA
469   3       IF REQUEST IS FOR VERIFY, THEN                                                                           XDDBA
470   4          STORE WORD 3 OF ENTRY IN ISIZE                                                                        XDDBA
471   3       ELSE                                                                                                     XDDBA
472   4          REPLACE ENTRY WITH LAST ENTRY                                                                         XDDBA
473   4          STORE ZEROES IN LAST ENTRY                                                                            XDDBA
474   4          DECREMENT # OF CURRENT ENTRIES                                                                        XDDBA
475   3       ENDIF                                                                                                    XDDBA
476   2    ENDIF                                                                                                       XDDBA
477   2    CALL WRITF TO WRITE RECORD TO  XPDB                                                                         XDDBA
478   1 EXIT TO :RETURN:                                                                                               XDDBA

479   2    :NAMERR: SET IERR = 1 AND EXIT TO :RETURN:                                                                  XDDBA

480   2    :MAXERR: SET IERR = 2 AND EXIT TO :RETURN:                                                                  XDDBA

481   2    :FILERR: SET IERR = FMGR ERROR CODE                                                                         XDDBA

482   2    :RETURN: CALL CLOSE FOR  XPDB                                                                               XDDBA
483   2    CALL XRULK TO RETURN RESOURCE #                                                                             XDDBA
484   1 END XDDBA                                                                                                      XDDBA
```

```
1 C*********                                                  XDELE
1 CD0                                                         XDELE
1 CD0       FORTRAN CALLING PROCEEDURE                        XDELE
1 CD0                                                         XDELE
1 CD0   CALL XDELE                                            XDELE
1 CD0                                                         XDELE
1 C*********                                                  XDELE
1 CD1                                                         XDELE
1 CD1   XDELE PROCESSES THE DELETE DIRECTIVE. EACH ELEMENT    XDELE
1 CD1   SPECIFIED ON THE DIRECTIVE IS DELETED FROM THE AWA.   XDELE
1 CD1   IF THE ELEMENT IS A DATA BASE, THE ASSOCIATED FILE    XDELE
1 CD1   MANAGER FILE IS PURGED AND FOR A PDB THE PDB DIRECTORY XDELE
1 CD1   IS UPDATED                                            XDELE
1 CD1                                                         XDELE
1 C*********                                                  XDELE
1 CD2                                                         XDELE
1 CD2       INPUT                                             XDELE
1 CD2                                                         XDELE
1 CD2   COMMON XE - COMBUF, COMPTR, LU, QUAL, TOKENS          XDELE
1 CD2                                                         XDELE
1 C*********                                                  XDELE
1 CD3                                                         XDELE
1 CD3       OUTPUT                                            XDELE
1 CD3                                                         XDELE
1 CD3   COMMON XE - REQBUF                                    XDELE
1 CD3                                                         XDELE
1 C*********                                                  XDELE
1 CD5                                                         XDELE
1 CD5       NOTES                                             XDELE
1 CD5                                                         XDELE
1 CD5   ROUTINES USED                                         XDELE
1 CD5                                                         XDELE
1 CD5       EXEC                                              XDELE
1 CD5       IAND                                              XDELE
1 CD5       PURGE                                             XDELE
1 CD5       XDDBD                                             XDELE
1 CD5       XREQ                                              XDELE
1 CD5       XREXT                                             XDELE
1 CD5       XRI6                                              XDELE
1 CD5       XRMOV                                             XDELE
1 CD5       XRMSG                                             XDELE
1 CD5       XRPCK                                             XDELE
1 CD5       XRQFN                                             XDELE
1 CD5       XRUPK                                             XDELE
1 CD5       XUDBG                                             XDELE
1 CD5                                                         XDELE
1 C*********                                                  XDELE
```

```
534   1 BEGIN XDELE                                                                                                    XDELE
535   2    DO WHILE END-OF-STATEMENT NOT REACHED PROCESSING EACH ELEMENT SPECIFIED                                      XDELE
536   3       ERREXIT IF COMMA IS NOT NEXT LEXICAL ELEMENT :ERR09:                                                     XDELE
537   3       IF CLASS DESIGNATOR IS SPECIFIED, THEN                                                                   XDELE
538   4          SET REQUESTED CLASS APPROPRIATELY (B, S, I, D, OR F)                                                  XDELE
539   3       ELSE                                                                                                     XDELE
540   4          SET REQUESTED CLASS TO BE (B)                                                                         XDELE
541   3       ENDIF                                                                                                    XDELE
542   3       IF DATA BASE TO BE DELETED, THEN                                                                         XDELE
543   4          BUILD AND ISSUE AWA MANAGER REQUEST FOR TOC ENTRY                                                     XDELE
544   3       ENDIF                                                                                                    XDELE
545   3       IF ELEMENT IS NOT A MASTER DATA BASE, THEN                                                               XDELE
546   4          IF ELEMENT IS A PERSONAL DATA BASE,THEN                                                               XDELE
547   5             CALL XDDBD TO DELETE THIS PDB FROM  XPDB                                                           XDELE
548   5             IF ERROR IS RETURNED, THEN                                                                         XDELE
549   6                CALL XRMSG - "FILE ACCESS ERROR #-..  XPDB"                                                     XDELE
550   6                SET ERROR FLAG                                                                                  XDELE
551   5             ENDIF                                                                                              XDELE
552   4          ENDIF                                                                                                 XDELE
553   4          BUILD AND ISSUE AWA MANAGER REQUEST TO DELETE ELEMENT SPECIFIED                                       XDELE
554   4          IF RETURN CODE INDICATES ELEMENT DOES NOT EXIST, THEN                                                 XDELE
555   5             CALL XRMSG - 'XXXXXX NOT FOUND'                                                                    XDELE
556   5             SET ERROR FLAG                                                                                     XDELE
557   4          ENDIF                                                                                                 XDELE
558   4          IF CLASS IS DATA BASE (D), OR                                                                         XDELE
559   5             CLASS IS DRDE (F), THEN                                                                            XDELE
560   5             IF ERROR FLAG IS NOT SET, THEN                                                                     XDELE
561   6                CALL XRGFN TO CONSTRUCT FILE NAME                                                               XDELE
562   6                ISSUE RTE PURGE FOR THE FILE                                                                    XDELE
563   6                IF RETURN CODE FROM PURGE, THEN                                                                 XDELE
564   7                   CALL XRMSG - "FMGR ERROR NNN XXXXXX"                                                         XDELE
565   6                ENDIF                                                                                           XDELE
566   5             ENDIF                                                                                              XDELE
567   4          ENDIF                                                                                                 XDELE
568   3       ELSE                                                                                                     XDELE
569   4          CALL XRMSG - "... IS A MDB.  NOT DELETED."                                                            XDELE
570   3       ENDIF                                                                                                    XDELE
571   2    ENDDO                                                                                                       XDELE
572   1 EXIT TO :RETURN:                                                                                               XDELE

573   2    :ERR09: CALL XRMSG - "SYNTAX ERROR"                                                                         XDELE

574   2    :RETURN:                                                                                                    XDELE
575   1 END XDELE                                                                                                      XDELE
```

```
577     1 CD**************                                                                    XDLIS
578     1 CDO                                                                               XDLIS
579     1 CDO             CALL XDLIS (TOCLST)                                               XDLIS
580     1 CDO                                                                               XDLIS
581     1 CD**************                                                                    XDLIS
582     1 CD1                                                                               XDLIS
583     1 CD1       XDLIS PROCESSES A LIST OF ELEMENTS TO BE STORED/RESTORED BY              XDLIS
584     1 CD1       DECODING EACH ELEMENT AND FLAGGING IT IN TOCLST AS AN ELEMENT           XDLIS
585     1 CD1       TO BE STORED/RESTORED.                                                  XDLIS
586     1 CD1                                                                               XDLIS
587     1 CD**************                                                                    XDLIS
588     1 CD2                                                                               XDLIS
589     1 CD2       INPUTS IN CALLING SEQUENCE:                                             XDLIS
590     1 CD2                                                                               XDLIS
591     1 CD2             TOCLST - (INTEGER, 1200 WORDS) ADDRESS OF FIRST TOC ENTRY         XDLIS
592     1 CD2                      OF A LIST OF POSSIBLE ELEMENTS' TOC ENTRIES              XDLIS
593     1 CD2                                                                               XDLIS
594     1 CD2       INPUTS IN COMMON:                                                       XDLIS
595     1 CD2                                                                               XDLIS
596     1 CD2             XB(151) ABFLG, XB(201) NOTOC                                      XDLIS
597     1 CD2                                                                               XDLIS
598     1 CD**************                                                                    XDLIS
599     1 CD3                                                                               XDLIS
600     1 CD3       OUTPUTS IN CALLING SEQUENCE:                                            XDLIS
601     1 CD3                                                                               XDLIS
602     1 CD3             TOCLST - (INTEGER, 1200 WORDS) LIST OF TOC ENTRIES; THOSE         XDLIS
603     1 CD3                      THAT ARE TO BE STORED/ RESTORE: ARE FLAGGED.            XDLIS
604     1 CD3                                                                               XDLIS
605     1 CD3       OUTPUTS IN COMMON:                                                      XDLIS
606     1 CD3                                                                               XDLIS
607     1 CD3             XB(151) ABFLG, XB(157) TOTSIZ, XB(158) TOTWRD                     XDLIS
608     1 CD3                                                                               XDLIS
609     1 CD**************                                                                    XDLIS
610     1 CD4                                                                               XDLIS
611     1 CD4       INTERNAL XB COMMON USED:                                                XDLIS
612     1 CD4                                                                               XDLIS
613     1 CD4             XB(151) ABFLG - (INTEGER, 1 WORD) ABORT FLAG                      XDLIS
614     1 CD4             XB(152) ERFLG - (INTEGER, 1 WORD) ERROR MESSAGE FLAG              XDLIS
615     1 CD4             XB(153) MSGNO - (INTEGER, 1 WORD) MESSAGE NUMBER                  XDLIS
616     1 CD4             XB(157) TOTSIZ- (INTEGER, 1 WORD) TOTAL # BLOCKS OF DATA TO       XDLIS
617     1 CD4                             BE STORED/RESTORED                                XDLIS
618     1 CD4             XB(158) TOTWRD- (INTEGER, 1 WORD) TOTAL # WORDS OF DATA TO        XDLIS
619     1 CD4                             BE STORED/RESTORED                                XDLIS
620     1 CD4             XB(198) RETC  - (INTEGER, 1 WORD) XTCOM RETURN CODE               XDLIS
621     1 CD4             XB(199) DATCLS- (INTEGER, 1 WORD) CLASS OF DATA BEING SEARCHED    XDLIS
622     1 CD4                             FOR IN LIST OF TOC ENTRIES                        XDLIS
623     1 CD4             XB(200) NAMPTR- (INTEGER, 1 WORD) POINTER TO NAME IN COMBUF       XDLIS
624     1 CD4             XB(201) NOTOC - (INTEGER, 1 WORD) NUMBER ENTRIES IN TOCLST        XDLIS
625     1 CD4                                                                               XDLIS
626     1 CD**************                                                                    XDLIS
627     1 CD5                                                                               XDLIS
628     1 CD5       COMMON USED:                                                            XDLIS
629     1 CD5                                                                               XDLIS
630     1 CD5             EQUIVALENCE          (XE(1),   LU    ),                         XDLIS
631     1 CD5           + (XE(5),   MASSTA),   (XE(85),  EOS   ),                         XDLIS
632     1 CD5           + (XE(89),  NAME  ),   (XE(92),  MTPH  ),                         XDLIS
633     1 CD5           + (XE(113), COMMA ),   (XE(144), COMPTR),                         XDLIS
634     1 CD5           + (XE(145), COMBUF)                                               XDLIS
635     1 CD5                                                                               XDLIS
```

```
636    1 CDS      RTE ROUTINES USED:                                                    XDLIS
637    1 CDS                                                                            XDLIS
638    1 CDS           EXEC                                                             XDLIS
639    1 CDS                                                                            XDLIS
640    1 CDS      FDS ROUTINES USED:                                                    XDLIS
641    1 CDS                                                                            XDLIS
642    1 CDS           XRCPR, XREXT, XRMOV, XRMSG,                                      XDLIS
643    1 CDS           XRSET, XRSFL, XRSFR, XTCOM                                       XDLIS
644    1 CDS                                                                            XDLIS
645    1 CD**************                                                               XDLIS
```

```
647   1 BEGIN XDLIS                                                                    XDLIS
648   2    DO WHILE ERROR FLAG IS ON OR UNTIL RESPONSE IS CR                           XDLIS
649   3       TURN ERFLG OFF                                                           XDLIS

650   3       :RTN1:                                                                   XDLIS
651   3       DO UNTIL EOS IS SENSED IN COMBUF                                         XDLIS
652   4          ERREXIT IF TOKEN IS NOT ",NAME" TO :ERR1:                             XDLIS
653   4          SAVE INDEX TO NAME FIELD                                              XDLIS
654   4          INCREMENT TO NEXT TOKEN                                               XDLIS
655   4          IF TOKEN IS A HYPHEN THEN                                             XDLIS
656   5             ERREXIT IF NEXT TOKEN IS NOT "NAME" TO :ERR1:                      XDLIS
657   5             INCREMENT TO NEXT TOKEN                                            XDLIS
658   5             DECODE CLASS NAME  (I, S, D, F)                                    XDLIS
659   5             ERREXIT IF CLASS SPECIFIED IS NOT VALID TO :ERR1:                  XDLIS
660   5             SET CLASS TO CLASS SPECIFIED                                       XDLIS
661   4          ELSE                                                                  XDLIS
662   5             SET CLASS TO DATA ELEMENT                                          XDLIS
663   4          ENDIF                                                                 XDLIS
664   4          IF XDLIS CALLED FROM STORE THEN                                       XDLIS
665   5             ERREXIT IF PREFIX IS DOUBLE EXCLAMATION TO :ERR2:                  XDLIS
666   4          ENDIF                                                                 XDLIS
667   4          ERREXIT IF NAME/CLASS ENTRY NOT FOUND IN TOC TO :ERR2:                XDLIS
668   4          CALL XRSET TO TURN  STORE/RESTORE BIT ON                              XDLIS
669   4          INCREMENT TOTAL SIZE BY SIZE OF THIS ELEMENT                          XDLIS
670   3       ENDDO                                                                    XDLIS

671   3       :RTN2:                                                                   XDLIS
672   3       IF ERROR FLAG IS ON THEN                                                 XDLIS
673   4          CALL XTCOM TO REPROMPT USER TO CONTINUE                               XDLIS
674   4          ERREXIT IF RESPONSE IS  X TO :ERR3:                                   XDLIS
675   3       ENDIF                                                                    XDLIS
676   2    ENDDO                                                                       XDLIS
677   1 EXIT XDLIS                                                                     XDLIS

678   2    :ERR1:                                                                      XDLIS
679   2    SET ERROR FLAG ON                                                           XDLIS
680   2    CALL XRMSG TO DISPLAY SYNTAX ERROR                                          XDLIS
681   2    GO TO :RTN2:                                                                XDLIS

682   2    :ERR2:                                                                      XDLIS
683   2    IF ERROR FLAG IS OFF THEN                                                   XDLIS
684   3       TURN ERROR FLAG ON                                                       XDLIS
685   3       CALL XRMSG TO DISPLAY NOT STORED/RESTORED MESSAGE                        XDLIS
686   2    ENDIF                                                                       XDLIS
687   2    CALL EXEC TO DISPLAY ELEMENT NAME                                           XDLIS
688   2    GO TO :RTN1:                                                                XDLIS

689   2    :ERR3:                                                                      XDLIS
690   2    SET ABFLG TO ABORT STORE/RESTORE OPERATION                                  XDLIS
691   1 END XDLIS                                                                      XDLIS
```

```
693    1 CD0                                                                    XDLST
694    1 CD0       FORTRAN CALLING PROCEEDURE                                   XDLST
695    1 CD0                                                                    XDLST
696    1 CD0    CALL XDLST                                                      XDLST
697    1 CD0                                                                    XDLST
698    1 C*********                                                             XDLST
699    1 CD1                                                                    XDLST
700    1 CD1    PROCESS THE LIST DIRECTIVE TO LIST ALL TABLES AND/OR            XDLST
701    1 CD1    DATA ELEMENTS SPECIFIED.                                        XDLST
702    1 CD1                                                                    XDLST
703    1 C*********                                                             XDLST
704    1 CD2                                                                    XDLST
705    1 CD2       INPUT                                                        XDLST
706    1 CD2                                                                    XDLST
707    1 CD2    COMMON XE - COMBUF, COMPTR, FLAGS, LU, TOKENS                   XDLST
708    1 CD2                                                                    XDLST
709    1 C*********                                                             XDLST
710    1 CD3                                                                    XDLST
711    1 CD3       OUTPUT                                                       XDLST
712    1 CD3                                                                    XDLST
713    1 CD3    PRINT TO LOGICAL UNIT 6 OR TO LOGICAL UNIT 'LU'                 XDLST
714    1 CD3                                                                    XDLST
715    1 C*********                                                             XDLST
716    1 CD4                                                                    XDLST
717    1 CD4       INTERNAL VARIABLES                                           XDLST
718    1 CD4                                                                    XDLST
719    1 CD4    COMMON XS - BUFFER = PRINT LINE TO BE OUTPUT                    XDLST
720    1 CD4                BUFPTR = INDEX INTO BUFFER FOR NEXT ASCII DATA      XDLST
721    1 CD4                                                                    XDLST
722    1 CD4                                                                    XDLST
723    1 CD4    COMMON XE - POINTR = CHAIN POINTER TO NEXT (OR 1ST) TOC         XDLST
724    1 CD4                                                                    XDLST
725    1 C*********                                                             XDLST
726    1 CD5                                                                    XDLST
727    1 CD5       NOTES                                                        XDLST
728    1 CD5                                                                    XDLST
729    1 CD5    USES ROUTINES - EXEC,  XEINT, XILSD, XILST,                     XDLST
730    1 CD5                    XREQ,  XREXT, XRMOV, XRMSG,                     XDLST
731    1 CD5                    XRPCK, XRUPK, XSLST, OPEN,                      XDLST
732    1 CD5                    CLOSE, READF                                    XDLST
733    1 CD5                                                                    XDLST
734    1 C*********                                                             XDLST
```

```
736   1 BEGIN XDLST                                                                    XDLST
737   2    IF DEVICE ID FIELD SPECIFIED, THEN                                          XDLST
738   3       SET LU FOR LISTING AS INDICATED ON  IRECTIVE                             XDLST
739   2    ELSE                                                                        XDLST
740   3       SET LU FOR LISTING TO BE TERMINAL LU                                     XDLST
741   2    ENDIF                                                                       XDLST
742   2    DO UNTIL END-OF-STATEMENT IS REACHED                                        XDLST
743   3       ERREXIT IF COMMA IS NOT SPECIFIED  :ERR09:                               XDLST
744   3       ERREXIT IF A NAME DOES NOT FOLLOW THE COMMA  :ERR09:                     XDLST
745   3       RETAIN NAME FOR XREQ CALL                                                XDLST
746   3       IF A CLASS DESIGNATOR IS SPECIFIED, THEN                                 XDLST
747   4          SET CLASS (I, S, OR D) FOR XREQ CALL                                  XDLST
748   3       ELSE                                                                     XDLST
749   4          USE DATA (D) CLASS IN XREQ CALL                                       XDLST
750   3       ENDIF                                                                    XDLST
751   3       CALL XREQ TO RETRIEVE THIS TABLE OR DATA ELEMENT                         XDLST
752   3       ERREXIT IF NOT FOUND  :ERR10:                                            XDLST
753   3       ERREXIT IF AWA SPACE NOT AVAILABLE FOR TABLE IN DWA  :ERR11:             XDLST
754   3       CALL EXEC TO PERFORM CLASS READ OF DATA OR TABLE INTO                    XDLST
755   3       BOTTOM OF WORKING BUFFER                                                 XDLST
756   3       IF INTERFACE TABLE TO BE LISTED, THEN                                    XDLST
757   4          CALL XRMOV TO MOVE CHARACTERISTICS TO TOP OF WORKING BUFFER           XDLST
758   4          READ SHORT PROMPTS FOR THIS PROCESSOR INTO WORKING BUFFER             XDLST
759   4          CALL XEINT TO INITIALIZE LITERAL ENTRIES                              XDLST
760   4          INITIALIZE INTERFACE TABLE EDITOR COMMON TO USE ITS LIST RTN.         XDLST
761   4          INITIALIZE 'LSTFLG' TO INDICATE ENTIRE TABLE TO BE LISTED             XDLST
762   4          CALL XILST TO LIST THE INTERFACE TABLE                                XDLST
763   3       ELSE                                                                     XDLST
764   4          IF SEQUENCE TABLE TO BE LISTED, THEN                                  XDLST
765   5             CALL XRMOV TO MOVE TABLE TO TOP OF WORKING BUFFER                  XDLST
766   5             CALL XSLST TO LIST SEQUENCE TABLE                                  XDLST
767   4          ELSE                                                                  XDLST
768   5             INITIALIZE INTERFACE TABLE EDITOR COMMON FOR USE OF ITS LIST RTN.  XDLST
769   5             SET 'LSTFLG' TO INDICATE ONLY 1 DATA ELEMENT BEING LISTED          XDLST
770   5             INITIALIZE PRINT BUFFER WITH NAME OF ELEMENT                       XDLST
771   5             CALL XILSD TO LIST THE DATA                                        XDLST
772   4          ENDIF                                                                 XDLST
773   3       ENDIF                                                                    XDLST
774   3       INCREMENT TO NEXT TOKEN IN OPERAND LIST OF THE DIRECTIVE IMAGE           XDLST

775   3       :ERR10: CALL XRMSG -- '... NOT FOUND'                                    XDLST

776   3       :ERR11: CALL XRMSG -- '... CANNOT BE MOVE FROM DWA TO AWA '              XDLST
777   2    ENDDO                                                                       XDLST
778   1 EXIT TO :RETURN:                                                               XDLST

779   2    :ERR09: CALL XRMSG -- 'SYNTAX ERROR'                                        XDLST

780   2    :RETURN:                                                                    XDLST
781   1 END XDLST                                                                      XDLST
```

```
783   1 CD**************                                                    XDOFF
784   1 CD0                                                                XDOFF
785   1 CD0       FORTRAN CALLING PROCEEDURE                               XDOFF
786   1 CD0                                                                XDOFF
787   1 CD0    CALL XDOFF                                                  XDOFF
788   1 CD0                                                                XDOFF
789   1 C*********                                                         XDOFF
790   1 CD1                                                                XDOFF
791   1 CD1    XDOFF CONFIRMS THE USER'S REQUEST FOR TERMINATION,          XDOFF
792   1 CD1    DELETES ALL DRDE AND UTDB FILES LOGGED IN THE AWA,          XDOFF
793   1 CD1    PERFORMS ABNORMAL TERMINATION, IF INDICATED, OR             XDOFF
794   1 CD1    RETURNS NORMAL PARMS TO THE FDS MANAGER AND TERMINATES      XDOFF
795   1 CD1    NORMALLY VIA RTE.                                           XDOFF
796   1 CD1                                                                XDOFF
797   1 C*********                                                         XDOFF
798   1 CD2                                                                XDOFF
799   1 CD2       INPUT                                                    XDOFF
800   1 CD2                                                                XDOFF
801   1 CD2    COMMON XE - LU, FLAGS, QUAL, REQBUF                         XDOFF
802   1 CD2                                                                XDOFF
803   1 CD2    COMMON XB - ORG = ORIGIN ADDRESS OF AWA. USED TO CALCULATE  XDOFF
804   1 CD2                      INDICES INTO 'AWA' FROM ADDRESS POINTERS  XDOFF
805   1 CD2                      OF TOC ENTRIES                            XDOFF
806   1 CD2                AWA = IMAGE OF AWA HEADER, CHAIN HEADS, AND     XDOFF
807   1 CD2                      TOC RETRIEVED VIA XREQ                    XDOFF
808   1 CD2                                                                XDOFF
809   1 C*********                                                         XDOFF
810   1 CD4                                                                XDOFF
811   1 CD4       INTERNAL VARIABLES                                       XDOFF
812   1 CD4                                                                XDOFF
813   1 CD4    COMMON X5 - POINTER = CHAIN POINTER TO NEXT (OR 1ST) TOC    XDOFF
814   1 CD4                          ENTRY. MOST SIGNIFICANT (BIT 15) BIT  XDOFF
815   1 CD4                          SET TO INDICATE END-OF-CHAIN          XDOFF
816   1 CD4                INDEX   = VALUE COMPUTED FROM POINTER TO BE     XDOFF
817   1 CD4                          FORTRAN INDEX INTO 'AWA' FOR NEXT TOC XDOFF
818   1 CD4                          ENTRY                                 XDOFF
819   1 C*********                                                         XDOFF
820   1 CD5                                                                XDOFF
821   1 CD5       NOTES                                                    XDOFF
822   1 CD5                                                                XDOFF
823   1 CD5    ROUTINE USED - EXEC,  PURGE, XDSTA, XPXIT, XREQ,  XREXT, XRI6,  XDOFF
824   1 CD5                   XRMOV, XRMSG, XRQFN, XRPCK, XRSET, XRUPK, XTCOM, XDOFF
825   1 CD5                   XUDBG, XVABN                                 XDOFF
826   1 CD5                                                                XDOFF
827   1 C*********                                                         XDOFF
```

```
829   1 BEGIN XDOFF                                                                                                    XDOFF
830   2    PROMPT USER FOR TERMINATION CONFIRMATION                                                                    XDOFF
831   2    IF USER RESPONDS GO AHEAD WITH TERMINATION THEN                                                             XDOFF
832   3       CALL XREQ TO REQUEST TOC AND CHAIN HEADS                                                                 XDOFF
833   3       CALL EXEC TO READ IN TOC AND CHAIN HEADS                                                                 XDOFF
834   3       IF CHAIN HEAD FOR DRDE FILES IS NOT NEGATIVE THEN                                                        XDOFF
835   4          DO UNTIL DRDE CHAIN HEAD IS NEGATIVE                                                                  XDOFF
836   5             IF CHAIN POINTS BEYOND END OF TOC BUFFER THEN                                                      XDOFF
837   6                OUTPUT 'XD13 TOC TOO LARGE, PURGE INCOMPLETE'                                                   XDOFF
838   5             EXIT DO                                                                                           XDOFF
839   5             ENDIF                                                                                             XDOFF
840   5             CALL XRQFN TO CREATE FILE NAME '/XXXXQ'                                                           XDOFF
841   5             CALL PURGE TO SCRATCH FILE                                                                        XDOFF
842   5             SET DRDE CHAIN HEAD TO TOC ENTRY CHAIN POINTER                                                    XDOFF
843   4          ENDDO                                                                                                XDOFF
844   3       ENDIF                                                                                                    XDOFF
845   3       IF CHAIN HEAD FOR DATA BASES IS NOT NEGATIVE, THEN                                                       XDOFF
846   4          DO UNTIL DATA BASE CHAIN HEAD IS NEGATIVE                                                             XDOFF
847   5             IF CHAIN POINTS BEYOND END OF TOC BUFFER THEN                                                      XDOFF
848   6                OUTPUT 'XD13 TOC TOO LARGE, PURGE INCOMPLETE'                                                   XDOFF
849   5             EXIT DO                                                                                           XDOFF
850   5             ENDIF                                                                                             XDOFF
851   5             IF TYPE OF DATA BASE IS UTDB, THEN                                                                XDOFF
852   6                CALL XRQFN TO CREATE FILE NAME  '*XXXXQ'                                                       XDOFF
853   6                CALL PURGE TO SCRATCH FILE                                                                     XDOFF
854   5             ENDIF                                                                                             XDOFF
855   5             SET DATA BASE CHAIN HEAD TO TOC ENTRY CHAIN POINTER                                               XDOFF
856   4          ENDDO                                                                                                XDOFF
857   3       ENDIF                                                                                                    XDOFF
858   3       CALL XDSTA TO OUTPUT USAGE STATISTICS                                                                    XDOFF
859   3       IF USER REQUESTED DEBUG SNAP THEN                                                                        XDOFF
860   4          CALL XUDBG                                                                                            XDOFF
861   3       ENDIF                                                                                                    XDOFF
862   3       IF USER REQUESTED ABEND DUMP THEN                                                                        XDOFF
863   4          CALL XVABN -  NO RETURN FROM THIS CALL                                                                XDOFF
864   3       ENDIF                                                                                                    XDOFF
865   3       SET PARAMETER 1 TO INDICATE TERMINATE EXEC                                                               XDOFF
866   3       CALL XPXIT TO WAIT ON I/O COMPLETION, RETURN PARAMS AND TERMINATE EXEC                                   XDOFF
867   2    ENDIF                                                                                                       XDOFF
868   2    RETURN                                                                                                      XDOFF
869   1 END XDOFF                                                                                                      XDOFF
```

```
871   1 CD**************                                                                                    XDRDE
872   1 CD0                                                                                                XDRDE
873   1 CD0       FORTRAN CALLING SEQUENCE:                                                                XDRDE
874   1 CD0                                                                                                XDRDE
875   1 CD0             CALL XDRDE (DATBUF, DBDCB)                                                         XDRDE
876   1 CD0                                                                                                XDRDE
877   1 CD**************                                                                                    XDRDE
878   1 CD1                                                                                                XDRDE
879   1 CD1       XDRDE HANDLES THE RESTORING OF DATA ELEMENTS, SEQUENCE TABLES                            XDRDE
880   1 CD1       AND INTERFACE TABLES                                                                     XDRDE
881   1 CD1                                                                                                XDRDE
882   1 CD**************                                                                                    XDRDE
883   1 CD2                                                                                                XDRDE
884   1 CD2       INPUTS FROM CALLING SEQUENCE:                                                            XDRDE
885   1 CD2                                                                                                XDRDE
886   1 CD2             DATBUF - (INTEGER, 1408 WORDS) BUFFER USED TO READ IN RECORDS                      XDRDE
887   1 CD2                      OF DATA FROM DATA BASE FILE                                               XDRDE
888   1 CD2             DBDCB  - (INTEGER, 144 WORDS) OPEN DATA BASE FILE DCB                              XDRDE
889   1 CD2                                                                                                XDRDE
890   1 CD2       INPUTS FROM XB COMMON:                                                                   XDRDE
891   1 CD2                                                                                                XDRDE
892   1 CD2             XB(154) ALLFLG, XB(175) ENDBLK, XB(182) TOTSIZ,                                    XDRDE
893   1 CD2             XB(201) NOTOC                                                                      XDRDE
894   1 CD2                                                                                                XDRDE
895   1 CD**************                                                                                    XDRDE
896   1 CD3                                                                                                XDRDE
897   1 CD3       OUTPUTS IN CALLING SEQUENCE:                                                             XDRDE
898   1 CD3                                                                                                XDRDE
899   1 CD3             DATBUF, DBDCB                                                                      XDRDE
900   1 CD3                                                                                                XDRDE
901   1 CD3       OUTPUTS IN XB COMMON                                                                     XDRDE
902   1 CD3                                                                                                XDRDE
903   1 CD3             XB(151) ABFLG, XB(152) ERFLG, XB(165) TOCWDS,                                      XDRDE
904   1 CD3             XB(169) TOPBLK, XB(170) ENDBLK, XB(171) TOCNDX,                                    XDRDE
905   1 CD3             XB(201) NOTOC                                                                      XDRDE
906   1 CD3                                                                                                XDRDE
907   1 CD**************                                                                                    XDRDE
908   1 CD4                                                                                                XDRDE
909   1 CD4       INTERNAL XB COMMON USED:                                                                 XDRDE
910   1 CD4                                                                                                XDRDE
911   1 CD4             XB(151) ABFLG - ABORT FLAG                                                         XDRDE
912   1 CD4             XB(152) ERFLG - ERROR MESSAGE FLAG                                                 XDRDE
913   1 CD4             XB(154) ALLFLG- RESTORE ALL UTDB FLAG                                              XDRDE
914   1 CD4             XB(155) DEBUG - DEBUG FLAG                                                         XDRDE
915   1 CD4             XB(159) FILNAM- DB FILE NAME                                                       XDRDE
916   1 CD4             XB(162) DATBLK- BLOCK # WHERE DATA ITEM BEGINS                                     XDRDE
917   1 CD4             XB(163) DATNDX- WORD INDEX INTO DATBUF WHERE DATA BEGINS                           XDRDE
918   1 CD4             XB(164) IERR  - ERROR FLAG FOR FMGR CALLS                                          XDRDE
919   1 CD4             XB(165) TOCWDS- # WORDS OF TOC IN TOCBUF (NOTOC *8) + 8                            XDRDE
920   1 CD4             XB(166) NOBLKS- # BLOCKS TO READ                                                   XDRDE
921   1 CD4             XB(167) LEN   - # WORDS OF DATA TO MOVE                                            XDRDE
922   1 CD4             XB(168) DATEND- BLOCK # WHERE DATA ITEM ENDS                                       XDRDE
923   1 CD4             XB(169) TOPBLK- BLOCK # OF FIRST BLOCK IN DATBUF                                   XDRDE
924   1 CD4             XB(170) ENDBLK- BLOCK # OF LAST BLOCK READ                                         XDRDE
925   1 CD4             XB(171) TOCNDX- INDEX INTO TOC OF CURRENT DATA ITEM                                XDRDE
926   1 CD4             XB(175) IL    - # WORDS OF DATA TO READ                                            XDRDE
927   1 CD4             XB(176) TOCENT- DATA BASE TOC ENTRY                                                XDRDE
928   1 CD4             XB(201) NOTOC - # TOC ENTRIES IN TOCBUF                                            XDRDE
929   1 CD4             XB(201) TOCBUF- TOC BUFFER                                                         XDRDE
```

```
930     1 CD4                                                                   XDRDE
931     1 CD*****************                                                   XDRDE
932     1 CD5                                                                   XDRDE
933     1 CD5       COMMON USED:                                                XDRDE
934     1 CD5                                                                   XDRDE
935     1 CD5         EQUIVALENCE (XE(1),  LU      ),                           XDRDE
936     1 CD5           + (XE(19), REQPTR), (XE(20), REQBUF)'                   XDRDE
937     1 CD5                                                                   XDRDE
938     1 CD5       FDS ROUTINES USED:                                          XDRDE
939     1 CD5                                                                   XDRDE
940     1 CD5         XDREQ, XRI6, XRMOV, XRMS6, XUDB6                          XDRDE
941     1 CD5                                                                   XDRDE
942     1 CD5       RTE ROUTINES USED:                                          XDRDE
943     1 CD5                                                                   XDRDE
944     1 CD5         EXEC, IAND, READF                                         XDRDE
945     1 CD4                                                                   XDRDE
946     1 CD*****************                                                   XDRDE
```

```
948   1 BEGIN XDRDE                                                                                                    XDRDE
949   2    INITIALIZE FILE INDICES TO INDICATE NO DATA IN DATBUF                                                       XDRDE
950   2    INITIALIZE REQUEST BUFFER TO SAY NO REQUESTS                                                                XDRDE
951   2    DO WHILE THERE ARE NON-ORDE FILES TO PROCESS                                                                XDRDE
952   3       IF ALLFLG IS ZERO OR IF STORE/RESTORE BIT IS ON THEN                                                     XDRDE
953   4          IF DATA IS NOT CURRENTLY IN DATBUF THEN                                                               XDRDE
954   5             COMPUTE SIZE OF DATA TO BE READ                                                                    XDRDE
955   5             CALL READF TO READ 1 BUFFER BEGINNING WITH DATBLK FOR THIS ELEMENT                                 XDRDE
956   5             ERREXIT IF READF ERROR TO :ERR1:                                                                   XDRDE
957   5             SET FILE INDICES INDICATING WHICH DATA IS IN DATBUF                                                XDRDE
958   5             ELSE, DATA BEGINS IN DATBUF                                                                        XDRDE
959   5             IF DATA DOES NOT END IN DATBUF THEN                                                                XDRDE
960   6                CALL XRMOV TO MOVE PARTIAL DATA TO TOP OF DATBUF                                                XDRDE
961   6                COMPUTE SIZE AND LOCATION OF DATA TO BE READ                                                    XDRDE
962   6                CALL READF TO READ ENOUGH TO FILL DATBUF                                                        XDRDE
963   6                ERREXIT IF READF ERROR TO :ERR1:                                                                XDRDE
964   6                SET FILE INDICES INDICATING WHICH DATA IS IN DATBUF                                             XDRDE
965   5             ENDIF                                                                                              XDRDE
966   4          ENDIF                                                                                                 XDRDE
967   4          BUILD AWA REQUEST TO ALLOCATE AND STORE DATA                                                          XDRDE
968   4          CALL EXEC TO WRITE DATA TO SAM                                                                        XDRDE
969   4          ERREXIT IF ERROR FROM EXEC TO  .. 2:                                                                  XDRDE
970   4          IF AWA REQUEST BUFFER IS FULL THEN                                                                    XDRDE
971   5             CALL XDREQ TO ISSUE REQUEST                                                                        XDRDE
972   4          EXIT XDRDE IF ERROR IN XDREQ                                                                          XDRDE
973   4          ENDIF                                                                                                 XDRDE
974   3       ENDIF                                                                                                    XDRDE
975   2    ENDDO                                                                                                       XDRDE
976   1 EXIT XDRDE                                                                                                     XDRDE

977   2    :ERR1:                                                                                                      XDRDE
978   2    CALL XRI6 TO CONVERT ERROR CODE TO ASKII                                                                    XDRDE
979   2    CALL XRMSG TO DISPLAY ERROR MESSAGE (208)                                                                   XDRDE
980   2    GO TO :ERR3:                                                                                                XDRDE

981   2    :ERR2:                                                                                                      XDRDE
982   2    CALL XRMSG TO DISPLAY ERROR MESSAGE (212)                                                                   XDRDE

983   2    :ERR3:                                                                                                      XDRDE
984   2    SET ABFLG TO SAY ABORT RESTORE                                                                              XDRDE
985   1 END XDRDE                                                                                                      XDRDE
```

```
 987    1 CD**************                                                                                    XDRDF
 988    1 CD0                                                                                               XDRDF
 989    1 CD0       FORTRAN CALLING SEQUENCE:                                                               XDRDF
 990    1 CD0                                                                                               XDRDF
 991    1 CD0            CALL XDRDF (DATBUF, DBDCB)                                                         XDRDF
 992    1 CD0                                                                                               XDRDF
 993    1 CD**************                                                                                  XDRDF
 994    1 CD1                                                                                               XDRDF
 995    1 CD1      XDRDF HANDLES THE RESTORING OF DRDE FILES FROM A MDB/UTDB FILE                           XDRDF
 996    1 CD1                                                                                               XDRDF
 997    1 CD**************                                                                                  XDRDF
 998    1 CD2                                                                                               XDRDF
 999    1 CD2      INPUTS IN CALLING SEQUENCE:                                                              XDRDF
1000    1 CD2                                                                                               XDRDF
1001    1 CD2            DATBUF  - (INTEGER, 1408 WORDS) BUFFER USED TO READ IN                             XDRDF
1002    1 CD2                      RECORDS OF DRDE FILES FROM DATA BASE FILE                                XDRDF
1003    1 CD2                                                                                               XDRDF
1004    1 CD2            DBDCB   - (INTEGER, 144 WORDS)  OPEN DATA BASE FILE DCB                            XDRDF
1005    1 CD2                                                                                               XDRDF
1006    1 CD2      INPUTS IN XB COMMON:                                                                     XDRDF
1007    1 CD2                                                                                               XDRDF
1008    1 CD2            XB(152) ERFLG, XB(154) ALLFLG, XB(165) TOCWDS,                                     XDRDF
1009    1 CD2            XB(169) TOPBLK, XB(170) ENDBLK, XB(171) TOCNDX,                                    XDRDF
1010    1 CD2            XB(182) TOTSIZ                                                                     XDRDF
1011    1 CD2                                                                                               XDRDF
1012    1 CD**************                                                                                  XDRDF
1013    1 CD3                                                                                               XDRDF
1014    1 CD3      OUTPUTS IN CALLING SEQUENCE:                                                             XDRDF
1015    1 CD3                                                                                               XDRDF
1016    1 CD3            DBDCB                                                                              XDRDF
1017    1 CD3                                                                                               XDRDF
1018    1 CD3      OUTPUTS IN XB COMMON:                                                                    XDRDF
1019    1 CD3                                                                                               XDRDF
1020    1 CD3            XB(151) ABFLG                                                                      XDRDF
1021    1 CD3                                                                                               XDRDF
1022    1 CD**************                                                                                  XDRDF
1023    1 CD4                                                                                               XDRDF
1024    1 CD4      INTERNAL XB COMMON USED:                                                                 XDRDF
1025    1 CD4            XB(151) ABFLG - ABORT FLAG                                                         XDRDF
1026    1 CD4            XB(152) ERFLG - ERROR MESSAGE FLAG                                                 XDRDF
1027    1 CD4            XB(154) ALLFLG- RESTORE ALL UTDB FLAG                                              XDRDF
1028    1 CD4            XB(155) DEBUG - DEBUG FLAG                                                         XDRDF
1029    1 CD4            XB(159) FILNAM- DB FILE NAME                                                       XDRDF
1030    1 CD4            XB(162) DATBLK- BLOCK # WHERE DRDE BEGINS                                          XDRDF
1031    1 CD4            XB(163) DATNDX- WORD INDEX INTO DATBUF WHERE DRDE BEGINS                           XDRDF
1032    1 CD4            XB(164) IERR  - ERROR FLAG FOR FNGR CALLS                                          XDRDF
1033    1 CD4            XB(165) TOCWDS- # WORDS OF TOC IN TOCBUF (NOTOC *8) * 8                            XDRDF
1034    1 CD4            XB(166) NOBLKS- # BLOCKS OF UTDB/DRDE TO READ/WRITE                                XDRDF
1035    1 CD4            XB(167) LEN   - # WORDS OF DATA TO MOVE                                            XDRDF
1036    1 CD4            XB(169) TOPBLK- BLOCK # OF FIRST BLOCK IN DATBUF                                   XDRDF
1037    1 CD4            XB(170) ENDBLK- BLOCK # OF LAST BLOCK READ                                         XDRDF
1038    1 CD4            XB(171) TOCNDX- INDEX TO CURRENT TOC ENTRY                                         XDRDF
1039    1 CD4            XB(172) DRDFIL- DRDE FILE NAME                                                     XDRDF
1040    1 CD4            XB(175) IL    - # WORDS OF DATA TO READ/WRITE                                      XDRDF
1041    1 CD4            XB(201) TOCBUF- TOC BUFFER                                                         XDRDF
1042    1 CD4                                                                                               XDRDF
1043    1 CD**************                                                                                  XDRDF
1044    1 CD5                                                                                               XDRDF
1045    1 CD5      COMMON USED:                                                                             XDRDF
```

```
1046    1 CDS           EQUIVALENCE          (XE(1),  LU     ),
1047    1 CDS          + (XE(19),  REQPTR),  (XE(20),  REQBUF)
1048    1 CDS          + (XE(142), CARTRG)
1049    1 CDS
1050    1 CDS
1051    1 CDS      FDS ROUTINES USED:
1052    1 CDS
1053    1 CDS            XDREQ, XRI6, XRMOV, XRMSG, XRQFM, XUDBG
1054    1 CDS
1055    1 CDS      RTE ROUTINES USED:
1056    1 CDS
1057    1 CDS            CLOSE, CREAT, EXEC, OPEN, PURGE, WRITF
1058    1 CDS
1059    1 CD**************
```

XDRDF

```
1061    1 BEGIN XDRDF                                                                   XDRDF
1062    2    DO WHILE THERE ARE TOC ENTRIES TO PROCESS                                  XDRDF
1063    3       IF ALLFLG IS ZERO OR THE STORE/RESTORE BIT IS ON THEN                   XDRDF
1064    4          CALL XRGFN TO CREATE THE DRDE FILE NAME                              XDRDF
1065    4          CALL CREAT TO CREATE THE DRDE FILE                                   XDRDF
1066    4          IF THERE WAS A CREATE ERROR THEN                                     XDRDF
1067    5             IF ERROR FLAG IS OFF THEN                                         XDRDF
1068    6                SET ERROR FLAG ON                                              XDRDF
1069    6                CALL XRMSG TO DISPLAY MAIN MESSAGE                             XDRDF
1070    5             ENDIF                                                             XDRDF
1071    5             CALL XRI6 TO CONVERT ERROR CODE TO ASKII                          XDRDF
1072    5             CALL EXEC TO DISPLAY DRDE NAME AND ERROR CODE                     XDRDF
1073    4          ELSE                                                                 XDRDF
1074    5             CALL CLOSE TO CLOSE FILE                                          XDRDF
1075    5             ERREXIT IF CLOSE ERROR TO :ERR1:                                  XDRDF
1076    5             CALL OPEN TO OPEN DRDE FILE AS TYPE 1                             XDRDF
1077    5             ERREXIT IF OPEN ERROR TO :ERR1:                                   XDRDF
1078    5             DO UNTIL ALL BLOCKS OF DRDE HAVE BEEN PROCESSED                   XDRDF
1079    6                IF DATA FOR DRDE IS IN DATBUF THEN                             XDRDF
1080    7                   CALL WRITF DATA TO DRDE                                     XDRDF
1081    7                   ERREXIT IF WRITF ERROR TO :ERR1:                            XDRDF
1082    6                ELSE                                                           XDRDF
1083    7                   CALL READF TO READ NEXT BUFFER OF DATA                      XDRDF
1084    7                   ERREXIT IF READF ERROR TO :ERR2:                            XDRDF
1085    7                   SET INDICES INDICATING DATA IN BUFFER                       XDRDF
1086    6                ENDIF                                                          XDRDF
1087    5             ENDDO                                                             XDRDF
1088    5             CALL CLOSE TO CLOSE DRDE FILE                                     XDRDF
1089    5             ERREXIT IF CLOSE FAILED TO :ERR1:                                 XDRDF
1090    5             BUILD AWA REQUEST TO ALLOCATE DRDE IN AWA                         XDRDF
1091    5             IF AWA REQUEST BUFFER IS FULL THEN                                XDRDF
1092    6                CALL XDREQ TO MAKE REQUEST                                     XDRDF
1093    5             EXIT XDRDF IF XDREQ ERROR                                         XDRDF
1094    5             ENDIF                                                             XDRDF
1095    4          ENDIF                                                                XDRDF
1096    3       ENDIF                                                                   XDRDF
1097    2    ENDDO                                                                      XDRDF
1098    1 EXIT XDRDF                                                                    XDRDF

1099    2    :ERR1:                                                                     XDRDF
1100    2    CALL PURGE TO PURGE DRDE FILE                                              XDRDF

1101    2    :ERR2:                                                                     XDRDF
1102    2    CALL XRMSG TO DISPLAY ERROR MESSAGE WITH FILE NAME                         XDRDF
1103    2    SET ABFLG TO SAY ABORT RESTORE                                             XDRDF
1104    1 END XDRDF                                                                     XDRDF
```

```
1106    1 C**********                                                                  XDREC
1107    1 CD0                                                                          XDREC
1108    1 CD0       FORTRAN CALLING PROCEEDURE                                         XDREC
1109    1 CD0                                                                          XDREC
1110    1 CD0    CALL XDREC                                                            XDREC
1111    1 CD0                                                                          XDREC
1112    1 C**********                                                                  XDREC
1113    1 CD1                                                                          XDREC
1114    1 CD1    XDREC PROCESSES THE RECALL DIRECTIVE. A UTDB IS CREATED AND            XDREC
1115    1 CD1    THE CONTENTS OF THE SPECIFIED PDB ARE COPIED TO IT.                   XDREC
1116    1 CD1                                                                          XDREC
1117    1 C**********                                                                  XDREC
1118    1 CD2                                                                          XDREC
1119    1 CD2       INPUT                                                              XDREC
1120    1 CD2                                                                          XDREC
1121    1 CD2    COMMON XE - CARTRG, COMBUF, COMPTR, FLAGS, LU, TOKENS                 XDREC
1122    1 CD2                                                                          XDREC
1123    1 CD2    FILES     - )XXXXO (PDB FILE SPECIFIED)                               XDREC
1124    1 CD2                                                                          XDREC
1125    1 C**********                                                                  XDREC
1126    1 CD3                                                                          XDREC
1127    1 CD3       OUTPUT                                                             XDREC
1128    1 CD3                                                                          XDREC
1129    1 CD3    COMMON XE - REQBUF, REQPTR                                            XDREC
1130    1 CD3                                                                          XDREC
1131    1 CD3    FILES     - *XXXXO (UTDB FILE SPECIFIED)                              XDREC
1132    1 CD3                                                                          XDREC
1133    1 C**********                                                                  XDREC
1134    1 CD4                                                                          XDREC
1135    1 CD4       INTERNAL VARIABLES                                                 XDREC
1136    1 CD4                                                                          XDREC
1137    1 CD4    DCBPDB - DCB FOR THE PDB FILE; ALLOCATED IN XB COMMON;                XDREC
1138    1 CD4             CONTAINS 1152 WORD BUFFER USED TO READ THE PDB               XDREC
1139    1 CD4             AND TO WRITE THE UTDB                                        XDREC
1140    1 CD4    DCBUTD - DCB FOR THE UTDB FILE; ALLOCATED IS XS COMMON                XDREC
1141    1 CD4                                                                          XDREC
1142    1 C**********                                                                  XDREC
```

```
1144    1 BEGIN XDREC                                                                                                   XDREC
1145    2    SET STATUS FLAG INDICATING PDB & UTDB FILES NOT OPEN                                                       XDREC
1146    2    ERREXIT IF ',' IS NOT NEXT TOKEN :ERR09:                                                                   XDREC
1147    2    INCREMENT TO NEXT TOKEN                                                                                    XDREC
1148    2    ERREXIT IF NEXT TOKEN IS NOT 'NAME' :ERR09:                                                                XDREC
1149    2    ERREXIT IF THIS NAME IS > 4 CHARACTERS :ERR16:                                                             XDREC
1150    2    ERREXIT IF THIS NAME BEGINS WITH DOUBLE EXCLAMATION :ERR16:                                                XDREC
1151    2    RETAIN THIS NAME AS PDB                                                                                    XDREC
1152    2    RETAIN CURRENT USER ID (QUALIFIER)                                                                         XDREC
1153    2    INCREMENT TO NEXT TOKEN                                                                                    XDREC
1154    2    IF NEXT TOKEN IS '-', THEN                                                                                 XDREC
1155    3       INCREMENT TO NEXT TOKEN                                                                                 XDREC
1156    3       ERREXIT IF NEXT TOKEN IS NOT 'NAME' :ERR09:                                                             XDREC
1157    3       ERREXIT IF 'NAME' IS GREATER THAN 1 CHARACTER :ERR19:                                                   XDREC
1158    3       ERREXIT IF QUALIFIER < 'A' OR > 'Z' :ERR19:                                                             XDREC
1159    3       SAVE IN R1 (ZERO FILL, RIGHT-JUSTIFIED) FORMAT AS QUALIFIER                                             XDREC
1160    3       INCREMENT TO NEXT TOKEN                                                                                 XDREC
1161    2    ENDIF                                                                                                      XDREC
1162    2    ERREXIT IF NEXT TOKEN IS NOT A 'NAME' :ERR09:                                                              XDREC
1163    2    INCREMENT TO NEXT TOKEN                                                                                    XDREC
1164    2    ERREXIT IF NEXT TOKEN IS NOT A 'NAME' :ERR09:                                                              XDREC
1165    2    ERREXIT IF THIS NAME IS > 4 CHARACTERS :ERR06:                                                             XDREC
1166    2    ERREXIT IF THIS NAME BEGINS WITH DOUBLE EXCLAMATION :ERR06:                                                XDREC
1167    2    RETAIN THIS NAME AS UTDB                                                                                   XDREC
1168    2    INCREMENT TO NEXT TOKEN                                                                                    XDREC
1169    2    ERREXIT IF NEXT TOKEN IS NOT EOS :ERR04:                                                                   XDREC
1170    2    BUILD AWA REQUEST TO VERIFY EXISTENCE OF UTDB                                                              XDREC
1171    2    CALL XREQ TO PROCESS AWA REQUEST                                                                           XDREC
1172    2    ERREXIT IF UTDB DOES ALREADY EXIST :ERR22:                                                                 XDREC
1173    2    CALL XDDBV TO VERIFY PDB AND RETRIEVE SIZE                                                                 XDREC
1174    2    ERREXIT IF PDB DOES NOT EXIST  :ERR16:                                                                     XDREC
1175    2    ERREXIT IF FMGR ERROR RETURNED  :ERR44:                                                                    XDREC
1176    2    CALL XRQFM TO BUILD PDB FILE NAME                                                                          XDREC
1177    2    CALL OPEN TO OPEN PDB FILE                                                                                 XDREC
1178    2    ERREXIT IF OPEN FAILED :ERR18:                                                                             XDREC
1179    2    SET STATUS FLAG INDICATING PDB FILE OPEN                                                                   XDREC
1180    2    BUILD AWA REQUEST TO ALLOCATE UTDB                                                                         XDREC
1181    2    CALL XREQ TO PROCESS AWA REQUEST                                                                           XDREC
1182    2    ERREXIT IF ALLOCATE REQUEST FAILS :ERR21:                                                                  XDREC
1183    2    SET STATUS FLAG TO INDICATE UTDB FILE ALLOCATED IN AWA                                                     XDREC
1184    2    RESTORE CURRENT USER'S ID (QUALIFIER)                                                                      XDREC
1185    2    CALL XRQFM TO BUILD UTDB FILE NAME                                                                         XDREC
1186    2    CALL CREAT TO CREATE TYPE 1 UTDB FILE                                                                      XDREC
1187    2    ERREXIT IF CREATE FAILED :ERR07:                                                                           XDREC
1188    2    SET STATUS FLAG INDICATING UTDB FILE NOW OPEN                                                              XDREC
1189    2    DO FOR EACH BUFFER OF DATA IN PDB FILE                                                                     XDREC
1190    3       CALL READF TO READ 1 BUFFER FROM PDB FILE                                                               XDREC
1191    3       ERREXIT IF READ ERROR :ERR18:                                                                           XDREC
1192    3       CALL WRITF  TO WRITE 1 BUFFER TO UTDB FILE                                                              XDREC
1193    3       ERREXIT IF WRITE ERROR :ERR07:                                                                          XDREC
1194    2    ENDDO                                                                                                      XDREC
1195    2    CALL CLOSE FOR PDB                                                                                         XDREC
1196    2    CALL CLOSE FOR UTDB                                                                                        XDREC
1197    1 EXIT XDREC                                                                                                    XDREC

1198    2    :ERR04:  ISSUE MESSAGE - "SYNTAX ERROR. EXTRANEOUS DATA"                                                   XDREC

1199    2    :ERR06:  ISSUE MESSAGE - "ILLEGAL UTDB NAME (NOT FOUND OR TOO LONG)"                                       XDREC
```

```
1200    2     :ERR07:  ISSUE MESSAGE - "UTDB FILE ACCESS ERROR ..."                                  XDREC
1201    2     :ERR09:  ISSUE MESSAGE - "SYNTAX ERROR - ILLEGAL OR MISSING FIELD"                     XDREC
1202    2     :ERR16:  ISSUE MESSAGE - "INVALID PDB FILE NAME..."                                    XDREC
1203    2     :ERR18:  ISSUE MESSAGE - "PDB FILE ACCESS ERROR ..."                                   XDREC
1204    2     :ERR19:  ISSUE MESSAGE - "USER ID IS INVALID FOR PDB/UTDB LOGGING"                     XDREC
1205    2     :ERR21:  ISSUE MESSAGE - "AWA OVERFLOW - XXXX NOT LOGGED"                              XDREC
1206    2     :ERR22:  ISSUE MESSAGE - "XXXX ALREADY EXISTS"                                         XDREC
1207    2     :ERR44:  ISSUE MESSAGE - "FILE ACCESS ERROR #-.. XPDB"                                 XDREC

1208    2     :RETURN:                                                                               XDREC
1209    2     IF STATUS FLAG INDICATES UTDB FILE IS OPEN, THEN                                       XDREC
1210    3        PURGE UTDB                                                                          XDREC
1211    2     ENDIF                                                                                  XDREC
1212    2     IF STATUS FLAG INDICATES UTDB IS LOGGED IN AWA, THEN                                   XDREC
1213    3        CALL XREQ TO DELETE UTDB FROM AWA                                                   XDREC
1214    2     ENDIF                                                                                  XDREC
1215    2     IF STATUS FLAG INDICATES PDB FILE IS OPEN, THEN                                        XDREC
1216    3        CALL CLOSE FOR PDB FILE                                                             XDREC
1217    2     ENDIF                                                                                  XDREC
1218    1  END XDREC                                                                                 XDREC
```

```
1220    1 C*******                                                                      XDREN
1221    1 CD0                                                                           XDREN
1222    1 CD0    FORTRAN CALLING PROCEDURE                                              XDREN
1223    1 CD0                                                                           XDREN
1224    1 CD0    CALL XDREN                                                             XDREN
1225    1 CD0                                                                           XDREN
1226    1 C*******                                                                      XDREN
1227    1 CD1                                                                           XDREN
1228    1 CD1    XDREN PROCESSES A USER REQUEST TO RENAME A SEQUENCE TABLE,              XDREN
1229    1 CD1    INTERFACE TABLE, DATA ELEMENT, DRDE, UTDB, OR PDB. ONLY                XDREN
1230    1 CD1    THOSE TABLES OR ELEMENTS LOGGED IN THE USERS AWA ARE RENAMED.          XDREN
1231    1 CD1                                                                           XDREN
1232    1 C*******                                                                      XDREN
1233    1 CD2                                                                           XDREN
1234    1 CD2    INPUT                                                                  XDREN
1235    1 CD2                                                                           XDREN
1236    1 CD2    COMMON XE - CARTRG, COMBUF, COMPTR, QUAL, REQPTR, TOKENS               XDREN
1237    1 CD2                                                                           XDREN
1238    1 C*******                                                                      XDREN
1239    1 CD3                                                                           XDREN
1240    1 CD3    OUTPUT                                                                 XDREN
1241    1 CD3                                                                           XDREN
1242    1 CD3    COMMON XE - REQBUF                                                     XDREN
1243    1 CD3                                                                           XDREN
1244    1 C*******                                                                      XDREN
1245    1 CD4    INTERNAL VARIABLES                                                     XDREN
1246    1 CD4                                                                           XDREN
1247    1 CD4    COMMON XS - (2) DBTYPE: TYPE CODE FOR DATA BASE FILES                  XDREN
1248    1 CD4                (3) I: INDEX                                               XDREN
1249    1 CD4            (148) IERR: FILE MANAGER ERROR RETURN                          XDREN
1250    1 CD4            (149) IMSG: ERROR MESSAGE NUMBERS                              XDREN
1251    1 CD4            (150) NFNAME: NEW FMGR FILE NAME                               XDREN
1252    1 CD4            (153) NNMPTR: POINTER TO NEW NAME IN COMBUF                    XDREN
1253    1 CD4            (154) OFNAME: OLD FMGR FILE NAME                               XDREN
1254    1 CD4            (157) ONMPTR: POINTER TO OLD NAME IN COMBUF                    XDREN
1255    1 CD4            (158) TOCENT: TOC ENTRY FOR DATA BASE RENAME                   XDREN
1256    1 CD4            (166) DATCLS: DATA CLASS CODE (STORED IN LEFT BYTE)            XDREN
1257    1 CD4                                                                           XDREN
1258    1 CD4    COMMON XB - (201) IOCB: EXEC BUFFER                                    XDREN
1259    1 CD4                                                                           XDREN
1260    1 C*******                                                                      XDREN
1261    1 CD5                                                                           XDREN
1262    1 CD5    EXTERNAL REFERENCES                                                    XDREN
1263    1 CD5                                                                           XDREN
1264    1 CD5    ROUTINES USED -                                                        XDREN
1265    1 CD5                                                                           XDREN
1266    1 CD5       EXEC                                                                XDREN
1267    1 CD5       IAND                                                                XDREN
1268    1 CD5       KCVT                                                                XDREN
1269    1 CD5       NAMF                                                                XDREN
1270    1 CD5       XDDBA                                                               XDREN
1271    1 CD5       XDUBD                                                               XDREN
1272    1 CD5       XREQ                                                                XDREN
1273    1 CD5       XRMOV                                                               XDREN
1274    1 CD5       XRMSG                                                               XDREN
1275    1 CD5       XRQFN                                                               XDREN
1276    1 CD5       XRSFR                                                               XDREN
1277    1 CD5                                                                           XDREN
1278    1 C*******                                                                      XDREN
```

```
1280    1 BEGIN XDREN                                                                                                    XDREN
1281    1 EXIT TO :SYNTAX: IF FIRST TOKENS ARE NOT ",NAME"                                                               XDREN
1282    2    SAVE POINTER TO CURRENT NAME                                                                                XDREN
1283    2    INCREMENT TO NEXT TOKEN                                                                                     XDREN
1284    2    IF TOKEN IS A HYPHEN, THEN                                                                                  XDREN
1285    2    EXIT TO :SYNTAX: IF NEXT TOKEN IS NOT NAME                                                                  XDREN
1286    3       DECODE CLASS NAME                                                                                        XDREN
1287    2    EXIT TO :CLASER: IF CLASS SPECIFIED IS NOT VALID (S,I,D,F,B)                                                XDREN
1288    2    ELSE                                                                                                        XDREN
1289    3       SET CLASS TO DE                                                                                          XDREN
1290    2    ENDIF                                                                                                       XDREN
1291    1 EXIT TO :SYNTAX: IF NEXT TOKENS ARE NOT ",NAME"                                                                XDREN
1292    2    SAVE POINTER TO NEW NAME                                                                                    XDREN
1293    2    IF CLASS IS DATA BASE OR DRDE, THEN                                                                         XDREN
1294    2    EXIT TO :TOOLNG: IF NEW NAME IS MORE THAN 4 CHARACTERS                                                      XDREN
1295    2    ENDIF                                                                                                       XDREN
1296    2    IF CLASS IS DATA BASE, THEN                                                                                 XDREN
1297    2    EXIT TO :TOOLNG: IF NEW NAME STARTS WITH DOUBLE EXCLAMATION                                                 XDREN
1298    2    ENDIF                                                                                                       XDREN
1299    2    IF CLASS IS INTERFACE TABLE, THEN                                                                           XDREN
1300    3       CALL EXEC TO WRITE/READ NEW NAME                                                                         XDREN
1301    3       BUILD AWA MANAGER REQUEST TO CHANGE NAME IN TOC                                                          XDREN
1302    3       BUILD AWA MANAGER REQUEST TO STORE NEW NAME IN TABLE                                                     XDREN
1303    3       CALL XREQ TO ISSUE REQUESTS                                                                              XDREN
1304    2    EXIT TO :CALGET: IF RETURN CODE INDICATES ERROR                                                             XDREN
1305    2    ELSE                                                                                                        XDREN
1306    3       IF CLASS IS DATA BASE, THEN                                                                              XDREN
1307    4          BUILD AND ISSUE AWA MANAGER REQUEST FOR TOC ENTRY                                                     XDREN
1308    3       EXIT TO :NAMERR: IF RETURN INDICATES ERROR                                                               XDREN
1309    3       EXIT TO :INVALD: IF DATA BASE IS AN MDB                                                                  XDREN
1310    3       ENDIF                                                                                                    XDREN
1311    3       BUILD AND ISSUE MANAGER REQUEST FOR NAME CHANGE                                                          XDREN
1312    2    EXIT TO :NAMERR: IF RETURN CODE INDICATES ERROR                                                             XDREN
1313    3       IF CLASS IS DATA BASE OR DRDE, THEN                                                                      XDREN
1314    4          CALL XRQFN TO FORMAT FILE NAMES                                                                       XDREN
1315    4          CALL FILE MANAGER TO CHANGE DISC FILE NAME                                                            XDREN
1316    3       EXIT TO :UNDO: IF FILE MANAGER RETURNS ERROR                                                             XDREN
1317    4          IF FILE IS A PDB, THEN                                                                                XDREN
1318    5             CALL XDDBD TO DELETE OLD PDB FROM  XPDB                                                            XDREN
1319    4          EXIT TO :NAMAGN: IF ERROR RETURNED                                                                    XDREN
1320    5             CALL XDDBA TO ADD NEW PDB NAME TO  XPDB                                                            XDREN
1321    4          EXIT TO :TELUSR: IF ERROR RETURNED                                                                    XDREN
1322    4          ENDIF                                                                                                 XDREN
1323    3       ENDIF                                                                                                    XDREN
1324    2    ENDIF                                                                                                       XDREN
1325    1 EXIT XDREN                                                                                                     XDREN

1326    2    :SYNTAX: CALL XRMSG TO DISPLAY SYNTAX ERROR AND EXIT                                                        XDREN

1327    2    :TOOLNG: CALL XRMSG ("NEW NAME IS TOO LONG") AND EXIT                                                       XDREN

1328    2    :CALGET:                                                                                                    XDREN
1329    2    CALL EXEC TO GET BUFFER CONTAINING NEW TABLE NAME                                                           XDREN

1330    2    :NAMERR: CALL XRMSG TO OUTPUT APPROPRIATE MESSAGE AND EXIT                                                  XDREN

1331    2    :CLASER: CALL XRMSG TO DISPLAY CLASS DESIGATION ERROR AND EXIT                                              XDREN

1332    2    :INVALD: CALL XRMSG ("AN MDB CANNOT BE RENAMED") AND EXIT                                                   XDREN
```

```
1333    2     :NAMAGN:
1334    2     CALL FILE MANAGER TO CHANGE NAME BACK                                                    XDREN
                                                                                                       XDREN
1335    2     :UNDO:
1336    2     BUILD AND ISSUE AN AWA MANAGER REQUEST TO CHANGE NAME BACK                               XDREN
1337    2     CALL XRMSG ("FILE MANAGER ERROR #    ; RENAME UNSUCCESSFUL") AND EXIT                    XDREN
                                                                                                       XDREN
1338    2     :TELUSR:
1339    2     CALL XRMSG ("PDB NOT LOGGED IN  XPDB; SYSTEM ERROR # ...") AND EXIT                      XDREN
1340    1 END XDREN                                                                                    XDREN
                                                                                                       XDREN
```

```
1342    1 C*********                                                                    XDREQ
1343    1 CD0                                                                           XDREQ
1344    1 CD0       FORTRAN CALLING PROCEEDURE                                          XDREQ
1345    1 CD0                                                                           XDREQ
1346    1 CD0   CALL XDREQ                                                              XDREQ
1347    1 CD0                                                                           XDREQ
1348    1 C*********                                                                    XDREQ
1349    1 CD1                                                                           XDREQ
1350    1 CD1   XDREQ IS CALLED BY XDRDE AND XDRDF TO PROCESS ACCUMULATED AWA           XDREQ
1351    1 CD1   REQUESTS.                                                               XDREQ
1352    1 CD1                                                                           XDREQ
1353    1 C*********                                                                    XDREQ
1354    1 CD2                                                                           XDREQ
1355    1 CD2       INPUT                                                               XDREQ
1356    1 CD2                                                                           XDREQ
1357    1 CD2   COMMON XE - LU, REQPTR, REQBUF                                          XDREQ
1358    1 CD2                                                                           XDREQ
1359    1 CD2   COMMON XB - DEBUG, ERFLG                                                XDREQ
1360    1 CD2                                                                           XDREQ
1361    1 CD2   COMMON XS - DRDDCB                                                      XDREQ
1362    1 CD2                                                                           XDREQ
1363    1 C*********                                                                    XDREQ
1364    1 CD3                                                                           XDREQ
1365    1 CD3       OUTPUT                                                              XDREQ
1366    1 CD3                                                                           XDREQ
1367    1 CD3   COMMON XB - ABFLG                                                       XDREQ
1368    1 CD3                                                                           XDREQ
1369    1 C*********                                                                    XDREQ
1370    1 CD5                                                                           XDREQ
1371    1 CD5       NOTES                                                               XDREQ
1372    1 CD5                                                                           XDREQ
1373    1 CD5   EXTERNAL REFERENCES                                                     XDREQ
1374    1 CD5                                                                           XDREQ
1375    1 CD5       EXEC                                                                XDREQ
1376    1 CD5       IAND                                                                XDREQ
1377    1 CD5       PURGE                                                               XDREQ
1378    1 CD5       XREQ                                                                XDREQ
1379    1 CD5       XREXT                                                               XDREQ
1380    1 CD5       XRI6                                                                XDREQ
1381    1 CD5       XRMOV                                                               XDREQ
1382    1 CD5       XRMSG                                                               XDREQ
1383    1 CD5                                                                           XDREQ
1384    1 C*********                                                                    XDREQ
```

```
1386    1 BEGIN XDREQ                                                                                                    XDREQ
1387    2    CALL XREQ TO PROCESS AWA REQUEST(S)                                                                         XDREQ
1388    2    IF AN ERROR RETURNED BY AWA MANAGER, THEN                                                                   XDREQ
1389    3       IF ERROR FLAG (ERFLG) IS ZERO, THEN                                                                      XDREQ
1390    4          TURN ON ERFLG INDICATING THAT MSG 234 HAS BEEN ISSUED                                                 XDREQ
1391    4          CALL XRMSG TO OUTPUT MSG 234 - 'FOLLOWING ELEMENTS NOT RESTORED'                                      XDREQ
1392    3       ENDIF                                                                                                    XDREQ
1393    3       CALL EXEC TO WRITE ELEMENT NAME, CLASS AND REASON                                                        XDREQ
1394    3       IF CLASS OF ELEMENT IS DRDE, THEN                                                                        XDREQ
1395    4          CALL PURGE TO DELETE THE FILE                                                                         XDREQ
1396    4          ELSE, ELEMENT RESIDES IN AWA                                                                          XDREQ
1397    4          CALL EXEC TO FREE CLASS NO. AND SAM BUFFER                                                            XDREQ
1398    3       ENDIF                                                                                                    XDREQ
1399    3       IF AWA REQUESTS EXIST IN REQBUF BEYOND FAILING REQUEST, THEN                                             XDREQ
1400    4          MOVE THESE REQUESTS TO TOP OF REQUEST BUFFER                                                          XDREQ
1401    3       ENDIF                                                                                                    XDREQ
1402    2    ELSE                                                                                                        XDREQ
1403    3       SET REQPTR TO 1 INDICATING NO REQUESTS PRESENT                                                           XDREQ
1404    2    ENDIF                                                                                                       XDREQ
1405    1 EXIT XDREQ                                                                                                     XDREQ
1406    1 END XDREQ                                                                                                      XDREQ
```

```
1408    1 C*********                                                                    XDRES
1409    1 CD0                                                                           XDRES
1410    1 CD0       FORTRAN CALLING PROCEDURE                                           XDRES
1411    1 CD0                                                                           XDRES
1412    1 CD0    CALL XDRES (DATBUF)                                                    XDRES
1413    1 CD0                                                                           XDRES
1414    1 C*********                                                                    XDRES
1415    1 CD1                                                                           XDRES
1416    1 CD1    XDRES PROCESSES THE RESTORE DIRECTIVE. THE SPECIFIED UTDB OR MDB       XDRES
1417    1 CD1    FILE IS OPENED AND ITS TOC IS READ. XDLIS IS CALLED TO MARK            XDRES
1418    1 CD1    TOC ENTRIES FOR RESTORE. XDRDE IS CALLED TO RESTORE DATA               XDRES
1419    1 CD1    ELEMENTS AND TABLES. XDRDF IS CALL TO RESTORE DRDE'S.                  XDRES
1420    1 CD1                                                                           XDRES
1421    1 C*********                                                                    XDRES
1422    1 CD2                                                                           XDRES
1423    1 CD2       INPUT                                                               XDRES
1424    1 CD2    COMMON XE - LU,     FLAGS,  REQPTR, REQBUF, TOKENS,                    XDRES
1425    1 CD2                CARTRG, COMPTR, COMBUF                                     XDRES
1426    1 CD2                                                                           XDRES
1427    1 CD2    CALLING SEQUENCE                                                       XDRES
1428    1 CD2                                                                           XDRES
1429    1 CD2          DATBUF - 1408 WORD BUFFER USED TO READ UTDB FILE                 XDRES
1430    1 CD2                                                                           XDRES
1431    1 C*********                                                                    XDRES
1432    1 CD3                                                                           XDRES
1433    1 CD3       OUTPUT                                                              XDRES
1434    1 CD3                                                                           XDRES
1435    1 CD3    COMMON XE - REQBUF                                                     XDRES
1436    1 CD3                                                                           XDRES
1437    1 CD3    COMMON XB - ABFLG  = ABORT FLAG, NON-ZERO VALUE FROM XDRDE,            XDRES
1438    1 CD3                         XDRDF, XDLIS OR XDREQ INDICATES ABORT            XDRES
1439    1 CD3                         OF RESTORE                                       XDRES
1440    1 CD3                ALLFLG = SET NON-ZERO IF LIST OF ELEMENTS SPECIFIED      XDRES
1441    1 CD3                DEBUG  = DEBUG BIT OF 'FLAGS' IN XE COMMON               XDRES
1442    1 CD3                ENDBLK = BLOCK # OF LAST BLOCK READ                      XDRES
1443    1 CD3                ERFLG  = SET NON-ZERO IF MESSAGE #34 ISSUED SO           XDRES
1444    1 CD3                         THAT IT IS ISSUED ONLY ONCE                     XDRES
1445    1 CD3                FILNAM = UTDB/MDB FILE NAME                              XDRES
1446    1 CD3                TOCBUF = UTDB TOC ENTRIES, 8 WORDS EACH, MAX 1200        XDRES
1447    1 CD3                         WORDS                                           XDRES
1448    1 CD3                TOCENT = AWA TOC ENTRY FOR THE UTDB                      XDRES
1449    1 CD3                TOCNDX = INDEX TO NEXT UTDB TOC ENTRY                    XDRES
1450    1 CD3                TOTSIZ = NO. OF BLOCKS REMAINING IN UTDB FILE            XDRES
1451    1 CD3                                                                         XDRES
1452    1 C*********                                                                  XDRES
1453    1 CD5                                                                         XDRES
1454    1 CD5       NOTES                                                             XDRES
1455    1 CD5                                                                         XDRES
1456    1 CD5    ROUTINES CALLED                                                      XDRES
1457    1 CD5                                                                         XDRES
1458    1 CD5       CLOSE                                                             XDRES
1459    1 CD5       EXEC                                                              XDRES
1460    1 CD5       OPEN                                                              XDRES
1461    1 CD5       READF                                                             XDRES
1462    1 CD5       XDLIS                                                             XDRES
1463    1 CD5       XDRDE                                                             XDRES
1464    1 CD5       XDRDF                                                             XDRES
1465    1 CD5       XDREQ                                                             XDRES
1466    1 CD5       XREQ                                                              XDRES
```

```
XDRES
XDRES
XDRES
XDRES
XDRES
XDRES
XDRES
```



```
1467   1 CDS      XREXT
1468   1 CDS      XRI6
1469   1 CDS      XRMOV
1470   1 CDS      XDMSG
1471   1 CDS      XDQFN
1472   1 CDS      XUDBG
1473   1 CDS
1474   1 C*********
```

```
1476    1 BEGIN XDRES                                                                                                   XDRES
1477    2    SET ABFLG TO ZERO                                                                                          XDRES
1478    2    ERREXIT IF NEXT TOKEN IS NOT A COMMA :ERR09:                                                               XDRES
1479    2    ERREXIT IF FOLLOWING TOKEN IS NOT A NAME :ERR09:                                                           XDRES
1480    2    RETAIN THIS NAME AS DATA BASE TO BE RESTORED                                                               XDRES
1481    2    INCREMENT TO NEXT TOKEN                                                                                    XDRES
1482    2    IF TOKEN IS NOT A COMMA, THEN                                                                              XDRES
1483    3       ERREXIT IF TOKEN IS NOT EOS :ERR04:                                                                     XDRES
1484    2    ENDIF                                                                                                      XDRES
1485    2    BUILD AWA REQUEST FOR TOC ENTRY RETRIEVE                                                                   XDRES
1486    2    CALL XREQ TO PROCESS AWA REQUEST                                                                           XDRES
1487    2    ERREXIT IF AWA REQUEST FAILED :ERR10:                                                                      XDRES
1488    2    ERREXIT IF DATA BASE FOUND IS A PDB :ERR33:                                                                XDRES
1489    2    IF DATA BASE IS A UTDB, THEN                                                                               XDRES
1490    3       CALL XRQFN TO CONSTRUCT QUALIFIED FILE NAME                                                             XDRES
1491    2    ENDIF                                                                                                      XDRES
1492    2    CALL OPEN TO OPEN SPECIFIED FILE                                                                           XDRES
1493    2    ERREXIT IF OPEN FAILED :ERR08:                                                                             XDRES
1494    2    CALL READF TO READ FIRST RECORD OF DATA BASE FILE INTO TOCBUF                                              XDRES
1495    2    ERREXIT IF READ FAILED :ERR08:                                                                             XDRES
1496    2    INITIALIZE ENDBLK TO NUMBER OF TOC BLOCKS                                                                  XDRES
1497    2    IF TOC IS MORE THAN 1 BLOCK LONG, THEN                                                                     XDRES
1498    3       CALL READF TO READ REMAINING TOC ENTRIES INTO TOCBUF                                                    XDRES
1499    3       ERREXIT IF READ FAILED :ERR08:                                                                          XDRES
1500    2    ENDIF                                                                                                      XDRES
1501    2    UPDATE TOTSIZ TO NUMBER OF BLOCKS REMAINING IN FILE (DECREMENT BY ENDBLK)                                  XDRES
1502    2    CLEAR ERROR MESSAGE FLAG (ERFLG)                                                                           XDRES
1503    2    IF TOKEN IS EOS (I.E. NO LIST OF ELEMENTS), THEN                                                           XDRES
1504    3       SET ALLFLG TO ZERO INDICATING TO RESTORE ALL TOC ENTRIES                                                XDRES
1505    2    ELSE                                                                                                       XDRES
1506    3       SET ALLFLG NON-ZERO INDICATING TO RESTORE ONLY FLAGGED TOC ENTRIES                                      XDRES
1507    3       CALL XDLIS TO PROCESS ELEMENTS SPECIFIED AND TO FLAG TOC ENTRIES                                        XDRES
1508    2    EXIT XDRES IF ABFLG SET BY XDLIS                                                                           XDRES
1509    2    ENDIF                                                                                                      XDRES
1510    2    CALL XDRDE TO RESTORE AWA RESIDENT ELEMENTS                                                                XDRES
1511    1 EXIT XDRES IF ABFLG SET BY XDRDE                                                                              XDRES
1512    2    CALL XDRDF TO RESTORE DRDE'S                                                                               XDRES
1513    1 EXIT XDRES IF ABFLG SET BY XDRDF                                                                              XDRES
1514    2    CALL CLOSE TO CLOSE DATA BASE FILE                                                                         XDRES
1515    2    ERREXIT IF CLOSE FAILED :ERR08:                                                                            XDRES
1516    2    DO WHILE AWA REQUESTS REMAIN IN REQBUF                                                                     XDRES
1517    3       CALL XDREQ TO PROCESS AWA REQUESTS                                                                      XDRES
1518    2    EXIT XDRES IF ABFLG SET BY XDREQ                                                                           XDRES
1519    2    ENDDO                                                                                                      XDRES
1520    1 EXIT XDRES                                                                                                    XDRES

1521    2    :ERR04: CALL XRMSG - 'SYNTAX ERROR. EXTRANEOS DATA'                                                        XDRES

1522    2    :ERR08: CALL XRMSG - 'FILE MANAGER ERROR ... ......'                                                       XDRES

1523    2    :ERR09: CALL XRMSG - 'SYNTAX ERROR. MISSING OR ILLEGAL FIELD'                                              XDRES

1524    2    :ERR10: CALL XRMSG - '..... NOT FOUND'                                                                     XDRES

1525    2    :ERR33: CALL XRMSG - 'CAN NOT RESTORE A PDB'                                                               XDRES
1526    2    DO UNTIL ALL AWA REQUESTS IN REQBUF HAVE BEEN PROCESSED                                                    XDRES
1527    3       IF REQUEST IS TO STORE DATA, THEN                                                                       XDRES
1528    4          CALL EXEC TO FREE THE SPECIFIED CLASS NO. AND SAM BUFFER                                             XDRES
1529    3       ENDIF                                                                                                   XDRES
```

```
XDRES
XDRES
XDRES
```

```
1530      END/O
1531    \ EXIT XDRES
1532    ; END XDRES
```

```
1534    1 C*********                                                                    XDSAV
1535    1 CD0                                                                           XDSAV
1536    1 CD0       FORTRAN CALLING PROCEDURE                                           XDSAV
1537    1 CD0                                                                           XDSAV
1538    1 CD0    CALL XDSAV                                                             XDSAV
1539    1 CD0                                                                           XDSAV
1540    1 C*********                                                                    XDSAV
1541    1 CD1                                                                           XDSAV
1542    1 CD1    XDSAV PROCESSES THE SAVE DIRECTIVE. A PDB IS CREATED AND THE           XDSAV
1543    1 CD1    CONTENTS OF THE SPECIFIED UTDB ARE COPIED TO IT.                       XDSAV
1544    1 CD1                                                                           XDSAV
1545    1 C*********                                                                    XDSAV
1546    1 CD2                                                                           XDSAV
1547    1 CD2       INPUT                                                               XDSAV
1548    1 CD2                                                                           XDSAV
1549    1 CD2    COMMON XE - CARTRG, COMBUF, COMPTR, FLAGS, LU, TOKENS                  XDSAV
1550    1 CD2                                                                           XDSAV
1551    1 CD2    FILES     - *XXXXQ (UTDB FILE SPECIFIED)                               XDSAV
1552    1 CD2                                                                           XDSAV
1553    1 C*********                                                                    XDSAV
1554    1 CD3                                                                           XDSAV
1555    1 CD3       OUTPUT                                                              XDSAV
1556    1 CD3                                                                           XDSAV
1557    1 CD3    COMMON XE - REQBUF, REQPTR                                             XDSAV
1558    1 CD3                                                                           XDSAV
1559    1 CD3    FILES     - )XXXXQ (PDB FILE SPECIFIED)                                XDSAV
1560    1 CD3                                                                           XDSAV
1561    1 C*********                                                                    XDSAV
1562    1 CD4                                                                           XDSAV
1563    1 CD4       INTERNAL VARIABLES                                                  XDSAV
1564    1 CD4                                                                           XDSAV
1565    1 CD4    DCBPDB - DCB FOR THE PDB FILE; ALLOCATED IN XS COMMON                  XDSAV
1566    1 CD4    DCBUTD - DCB FOR THE UTDB FILE; ALLOCATED IN XB COMMON;                XDSAV
1567    1 CD4             CONTAINS 1152 WORD BUFFER USED TO READ THE                    XDSAV
1568    1 CD4             UTDB AND TO WRITE THE PDB.                                    XDSAV
1569    1 CD4                                                                           XDSAV
1570    1 C*********                                                                    XDSAV
```

```
1572    1 BEGIN XDSAV                                                                                    XDSAV
1573    2    SET STATUS FLAGS TO INDICATE NO FILES OPEN, NO PDB ALLOCATED                                XDSAV
1574    2    ERREXIT IF ',' IS NOT NEXT TOKEN :ERR09:                                                    XDSAV
1575    2    INCREMENT TO NEXT TOKEN                                                                     XDSAV
1576    2    ERREXIT IF TOKEN IS NOT 'NAME' :ERR09:                                                      XDSAV
1577    2    ERREXIT IF THIS NAME IS MORE THAN 4 CHARACTERS :ERR06:                                      XDSAV
1578    2    ERREXIT IF NAME BEGINS WITH DOUBLE EXCLAMATION :ERR06:                                      XDSAV
1579    2    RETAIN THIS NAME AS UTDB                                                                    XDSAV
1580    2    INCREMENT TO NEXT TOKEN                                                                     XDSAV
1581    2    ERREXIT IF ',' IS NOT NEXT TOKEN :ERR09:                                                    XDSAV
1582    2    INCREMENT TO NEXT TOKEN                                                                     XDSAV
1583    2    ERREXIT IF TOKEN IS NOT 'NAME' :ERR09:                                                      XDSAV
1584    2    ERREXIT IF THIS NAME IS MORE THAN 4 CHARACTERS :ERR16:                                      XDSAV
1585    2    ERREXIT IF NAME BEGINS WITH DOUBLE EXCLAMATION :ERR16:                                      XDSAV
1586    2    RETAIN THIS NAME AS PDB                                                                     XDSAV
1587    2    INCREMENT TO NEXT TOKEN                                                                     XDSAV
1588    2    ERREXIT IF TOKEN IS NOT EOS :ERR04:                                                         XDSAV
1589    2    BUILD AWA REQUEST TO RETRIEVE UTDB'S TOC ENTRY                                              XDSAV
1590    2    CALL XREQ TO PROCESS AWA REQUESTS                                                           XDSAV
1591    2    ERREXIT IF TOC RETRIEVE FAILED :ERR06:                                                      XDSAV
1592    2    ERREXIT IF TOC ENTRY DOES NOT INDICATE UTDB :ERR06:                                         XDSAV
1593    2    CALL XDDBA TO ADD PDB TO  XPDB                                                              XDSAV
1594    2    EPREXIT IF FMGR ERROR :ERR08:                                                               XDSAV
1595    2    ERREXIT IF PDB LIMIT EXCEEDED :ERR20:                                                       XDSAV
1596    2    SET STATUS FLAG INDICATING PDB LOGGED                                                       XDSAV
1597    2    CALL XRQFN TO BUILD UTDB FILE NAME                                                          XDSAV
1598    2    CALL OPEN TO OPEN UTDB FILE (SPECIFYING TYPE 1)                                             XDSAV
1599    2    ERREXIT IF OPEN FAILED :ERR18:                                                              XDSAV
1600    2    SET STATUS FLAG INDICATING UTDB FILE OPEN                                                   XDSAV
1601    2    BUILD AWA REQUEST TO ALLOCATE PDB -- USE SIZE OF UTDB                                       XDSAV
1602    2    CALL XREQ TO PROCESS AWA REQUEST                                                            XDSAV
1603    2    EPREXIT IF PDB IS DUPLICATE   :ERR17:                                                       XDSAV
1604    2    ERREXIT IF AWA OVERFLOW   :ERR21:                                                           XDSAV
1605    2    SET STATUS FLAG INDICATING PDB LOGGED IN AWA                                                XDSAV
1606    2    CALL XRQFN TO BUILD PDB FILE NAME                                                           XDSAV
1607    2    CALL CREAT  TO CREATE PDB FILE (SPE IFYING TYPE 1) USING SIZE                               XDSAV
1608    2    OF UTDB FILE FROM TOC ENTRY                                                                 XDSAV
1609    2    ERREXIT IF CREAT  FAILED :ERR18:                                                            XDSAV
1610    2    SET STATUS FLAG INDICATING PDB FILE OPEN                                                    XDSAV
1611    2    DO FOR ALL DATA IN UTDB FILE                                                                XDSAV
1612    2       CALL READF TO READ 1 BUFFER OF UTDB DATA                                                 XDSAV
1613    3       ERREXIT IF READ FAILED :ERR08:                                                           XDSAV
1614    3       CALL WRITF  TO WRITE 1 BUFFER TO PDB FILE                                                XDSAV
1615    3       ERREXIT IF WRITE FAILED :ERR08:                                                          XDSAV
1616    2    ENDDO                                                                                       XDSAV
1617    2    CLOSE UTDB FILE                                                                             XDSAV
1618    2    CLOSE PDB FILE                                                                              XDSAV
1619    1 EXIT XDSAV

1620    2    :ERR09: ISSUE MESSAGE "SYNTAX ERROR -- ILLEGAL OR MISSING FIELD"                            XDSAV

1621    2    :ERR06: ISSUE MESSAGE "INVALID UTDB FILE NAME ..."                                          XDSAV

1622    2    :ERR16: ISSUE MESSAGE "INVALID PDB FILE NAME ..."                                           XDSAV

1623    2    :ERR04: ISSUE MESSAGE "SYNTAX ERROR. EXTRANEOUS DATA"                                       XDSAV

1624    2    :ERR17: ISSUE MESSAGE "PDB .... ALREADY EXISTS"                                             XDSAV
```

```
1625    2    :ERR18: ISSUE MESSAGE 'PDB FILE ACCESS ERROR ... "                                  XDSAV
1626    2    :ERR08: ISSUE MESSAGE "FILE MANAGER ERROR ... ......"                               XDSAV
1627    2    :ERR20: ISSUE MESSAGE "AUTHORIZED LIMIT OF ... PDB'S                                XDSAV
1628    2    ALREADY REACHED"                                                                    XDSAV
1629    2    :ERR21: ISSUE MESSAGE "AWA OVERFLOW.  .... NOT LOGGED"                              XDSAV
1630    2    :RETURN:                                                                            XDSAV
1631    2    IF STATUS FLAG INDICATES PDB FILE IS OPEN, THEN                                     XDSAV
1632    3       PURGE PDB FILE                                                                   XDSAV
1633    2    ENDIF                                                                               XDSAV
1634    2    IF FLAG INDICATES PDB IS IN AWA, THEN                                               XDSAV
1635    3       CALL XREQ TO DELETE PDB FROM AWA                                                 XDSAV
1636    2    ENDIF                                                                               XDSAV
1637    2    IF FLAG INDICATES UTDB IS OPEN, THEN                                                XDSAV
1638    3       CLOSE UTDB                                                                       XDSAV
1639    2    ENDIF                                                                               XDSAV
1640    2    IF FLAG INDICATES PDB IS IN  XPDB, THEN                                             XDSAV
1641    3       CALL XDDBD TO DELETE PDB FROM  XPDB                                              XDSAV
1642    2    ENDIF                                                                               XDSAV
1643    1 END XDSAV                                                                              XDSAV
```

```
1641    1 CD0      FORTRAN CALLING PROCEDURE                                                                    XDSTA
1642    1 CD0                                                                                                   XDSTA
1643    1 CD0   CALL XDSTA (LU)                                                                                 XDSTA
1644    1 CD0                                                                                                   XDSTA
1645    1 C*********                                                                                            XDSTA
1646    1 CD1                                                                                                   XDSTA
1647    1 CD1      OUTPUT AWA AND BWA USAGE DATA AND STATISTICS ON AWA COMPACTION                                XDSTA
1648    1 CD1      PHASES                                                                                       XDSTA
1649    1 CD1                                                                                                   XDSTA
1650    1 C*********                                                                                            XDSTA
1651    1 CD2      INPUT                                                                                        XDSTA
1652    1 CD2   LU     - LOGICAL UNIT OF OUTPUT DEVICE                                                          XDSTA
1653    1 CD2                                                                                                   XDSTA
1654    1 CD2   XB COMMON                                                                                       XDSTA
1655    1 CD2   ASZ    - TOTAL SIZE OF AWA                                                                      XDSTA
1656    1 CD2   FRE    - AMOUNT OF FREE SPACE IN AWA                                                            XDSTA
1657    1 CD2   BSZ    - TOTAL SIZE OF BWA (IN 64 WORD SECTORS)                                                 XDSTA
1658    1 CD2   BFR    - AMOUNT OF FREE SECTORS IN BWA                                                          XDSTA
1659    1 CD2   NP     - ARRAY OF THREE COMPACTION PHASE COUNTS                                                 XDSTA
1660    1 CD2                                                                                                   XDSTA
1661    1 C*********                                                                                            XDSTA
1662    1 CD3      OUTPUT                                                                                       XDSTA
1663    1 CD3   USAGE AND STATISTICS TO LU                                                                      XDSTA
1664    1 C*********                                                                                            XDSTA
1665    1 CD5      NOTES                                                                                        XDSTA
1666    1 CD5   USES EXEC, XRI6                                                                                 XDSTA
1667    1 CD5                                                                                                   XDSTA
1668    1 C*********                                                                                            XDSTA
1669    1 *                                                                                                     XDSTA
1670    1 *                                                                                                     XDSTA
1671    1 *                                                                                                     XDSTA
1672    1 *                                                                                                     XDSTA
1673    1 BEGIN XDSTA                                                                                           XDSTA
1674    2    FORMAT AND PRINT AWA USAGE DATA                                                                    XDSTA
1675    2    FORMAT AND PRINT BWA USAGE DATA                                                                    XDSTA
1676    2    FORMAT AND PRINT COMPACTION STATISTICS                                                             XDSTA
1677    1 END XDSTA                                                                                             XDSTA
```

```
1683   1 CD**************                                                                                    XDSTO
1684   1 CD0                                                                                               XDSTO
1685   1 CD0       CALL XDSTO (DATBUF)                                                                     XDSTO
1686   1 CD0                                                                                               XDSTO
1687   1 CD**************                                                                                    XDSTO
1688   1 CD1                                                                                               XDSTO
1689   1 CD1       XDSTO IS THE STORE DIRECTIVE HANDLER. IT VERIFIES INPUTS ON                             XDSTO
1690   1 CD1       DIRECTIVE, BUILDS UTDB TOC, CREATES UTDB AND STORES UTDB TOC                            XDSTO
1691   1 CD1       ENTRY IN AWA.                                                                           XDSTO
1692   1 CD1                                                                                               XDSTO
1693   1 CD**************                                                                                    XDSTO
1694   1 CD2                                                                                               XDSTO
1695   1 CD2       INPUTS FROM CALLING SEQUENCE:                                                           XDSTO
1696   1 CD2                                                                                               XDSTO
1697   1 CD2            DATBUF (INTEGER 1480 WORDS) - BUFFER USED TO READ IN AWA TOC.                      XDSTO
1698   1 CD2                                                                                               XDSTO
1699   1 CD**************                                                                                    XDSTO
1700   1 CD4                                                                                               XDSTO
1701   1 CD4       INTERNAL XB COMMON USED:                                                                XDSTO
1702   1 CD4                                                                                               XDSTO
1703   1 CD4            XB(151) ABFLG - (INTEGER, 1 WORD) ABORT FLAG                                       XDSTO
1704   1 CD4            XB(152) ERFLG - (INTEGER, 1 WORD) ERROR MESSAGE FLAG                               XDSTO
1705   1 CD4            XB(153) MSGNO - (INTEGER, 1 WORD) MESSAGE NUMBER TO BE DISPLAYED                   XDSTO
1706   1 CD4            XB(157) TOTSIZ- (INTEGER, 1 WORD) TOTAL SIZE OF UTDB FILE                          XDSTO
1707   1 CD4            XB(158) TOTWRD- (INTEGER, 1 WORD) TOTAL WORDS IN A DRDE FILE                       XDSTO
1708   1 CD4            XB(159) FILE  - (INTEGER, 3 WORDS) UTDB FILE NAME (*XXXXC)                         XDSTO
1709   1 CD4            XB(162) DATREC- (INTEGER, 1 WORD) RECORD # WHERE DATA GOES NEXT                    XDSTO
1710   1 CD4            XB(164) UDBERR- (INTEGER, 1 WORD) UTDB FILE ERROR FLAG                             XDSTO
1711   1 CD4            XB(166) UDBNAM- (INTEGER, 1 WORD) UTDB NAME (XXXX)                                 XDSTO
1712   1 CD4            XB(201) NOTOC - (INTEGER, 1 WORD) NUMBER OF TOC ENTRIES                            XDSTO
1713   1 CD4            XB(201) TOCBUF- (INTEGER, 1200 WORDS) UTDB TOC BUFFER                              XDSTO
1714   1 CD4                                                                                               XDSTO
1715   1 CD**************                                                                                    XDSTO
1716   1 CD5                                                                                               XDSTO
1717   1 CD5       COMMON USED:                                                                            XDSTO
1718   1 CD5                                                                                               XDSTO
1719   1 CD5            EQUIVALENCE          (XE(1),   LU    ),                                            XDSTO
1720   1 CD5           + (XE(4),   FLAGS ),  (XE(19),  REQPTR),                                            XDSTO
1721   1 CD5           + (XE(20),  REQBUF),  (XE(85),  EOS   ),                                            XDSTO
1722   1 CD5           + (XE(89),  NAME  ),  (XE(113), COMMA ),                                            XDSTO
1723   1 CD5           + (XE(142), ICR   ),  (XE(144), COMPTR),                                            XDSTO
1724   1 CD5           + (XE(145), COMBUF)                                                                 XDSTO
1725   1 CD5                                                                                               XDSTO
1726   1 CD5       RTE ROUTINES USED:                                                                      XDSTO
1727   1 CD5                                                                                               XDSTO
1728   1 CD5           CLOSE, CREAT, EXEC, KCVT, PURGE, WRITF                                              XDSTO
1729   1 CD5                                                                                               XDSTO
1730   1 CD5       FDS ROUTINES USED:                                                                      XDSTO
1731   1 CD5                                                                                               XDSTO
1732   1 CD5            XDLIS, XDWRT, XRCPR, XREQ, XREXT, XRMOV,                                           XDSTO
1733   1 CD5            XRMSG, XRSET, XRSFL, XRSFR, XRQFN, XUDBG                                           XDSTO
1734   1 CD5                                                                                               XDSTO
1735   1 CD**************                                                                                    XDSTO
```

```
1737    1 BEGIN XDSTO                                                                    XDSTO
1738    2    SET ABFLG TO ZERO (ABORT FLAG)                                              XDSTO
1739    2    ERREXIT IF UTDB NAME IS NOT VALID TO :ERR2:                                 XDSTO
1740    2    BUILD REQUEST FOR AWA TOC                                                   XDSTO
1741    2    CALL XREQ TO MAKE MANAGER REQUEST                                           XDSTO
1742    2    CALL EXEC TO GET AWA TOC                                                    XDSTO
1743    2    ERREXIT IF SIZE OF TOC > MAXIMUM SIZE TO :ERR2:                             XDSTO
1744    2    DO FOR ALL DATA BASE CLASS ENTRIES                                          XDSTO
1745    3       ERREXIT IF NAME SPECIFIED ALREADY EXISTS TO :ERR2:                       XDSTO
1746    2    ENDDO                                                                       XDSTO
1747    2    SET ERFLG OFF (ERROR MESSAGE FLAG)                                          XDSTO
1748    2    SET TOTSIZ = 0 (UTDB TOTAL SIZE)                                            XDSTO
1749    2    IF WHOLE AWA IS TO BE STORED THEN                                           XDSTO
1750    3       DO FOR IT, ST, DE, DRDE AWA TOC ENTRIES                                  XDSTO
1751    4          DO FOR EACH ENTRY IN THIS CHAIN                                       XDSTO
1752    5             IF PREFIX IS NOT DOUBLE EXCLAMATION AND                            XDSTO
1753    6                PREFIX IS NOT AN AMPERSAND THEN                                 XDSTO
1754    6                SET STORE/RESTORE BIT ON IN TOC ENTRY                           XDSTO
1755    6                INCREMENT TOTSIZ BY SIZE OF THIS ELEMENT                        XDSTO
1756    5             ENDIF                                                              XDSTO
1757    4          ENDDO                                                                 XDSTO
1758    3       ENDDO                                                                    XDSTO
1759    2    ELSE                                                                        XDSTO
1760    3       CALL XDLIS TO PROCESS LIST TO BE STORED                                  XDSTO
1761    3       ERREXIT IF ABFLG IS NOT ZERO TO :ERR5:                                   XDSTO
1762    2    ENDIF                                                                       XDSTO
1763    2    SET NOTOC = 0 (NUMBER OF UTDB TOC ENTRIES)                                  XDSTO
1764    2    DO FOR IT, ST, DE, DRDE CHAINS                                              XDSTO
1765    3       DO FOR EACH ENTRY IN THIS CHAIN                                          XDSTO
1766    4          IF STORE/RESTORE BIT IS ON THEN                                       XDSTO
1767    5             TURN STORE/RESTORE BIT OFF                                         XDSTO
1768    5             BUILD UTDB TOC ENTRY                                               XDSTO
1769    5             INCREMENT NOTOC BY 1                                               XDSTO
1770    4          ENDIF                                                                 XDSTO
1771    3       ENDDO                                                                    XDSTO
1772    2    ENDDO                                                                       XDSTO
1773    2    ERREXIT IF THERE ARE NO UTDB TOC ENTRIES (NOTOC=0) TO :ERR2:                XDSTO
1774    2    COMPUTE DATREC AS FIRST RECORD AVAILABLE FOR DATA                           XDSTO
1775    2    CALL XP  M TO CREATE FILE NAME                                              XDSTO
1776    2    CALL       TO CREATE UTDB FOR TOTSIZ                                        XDSTO
1777    2    ERREX.  . ERROR IN CREAT TO :ERR3:                                          XDSTO
1778    2    CALL XDWRT TO WRITE UTDB FILE                                               XDSTO
1779    2    ERREXIT IF ABFLG IS 4 (DRDE LARGER THAN SPECIFIED) TO :ERR1:                XDSTO
1780    2    ERREXIT IF ABFLG IS 3 (DRDE FILE ERROR) TO:ERR4:                            XDSTO
1781    2    ERREXIT IF ABFLG IS 2 (UTDB FILE ERROR) TO:ERR3:                            XDSTO
1782    2    CALL WRITF TO WRITE TOC RECORDS AT RECORD 1                                 XDSTO
1783    2    ERREXIT IF ERROP IN WRITF TO :ERR3:                                         XDSTO
1784    2    CALL CLOSE TO CLOSE UTDB FILE                                               XDSTO
1785    2    ERREXIT IF ERROR IN CLOSE TO :ERR3:                                         XDSTO
1786    2    BUILD REQUEST TO ALLOCATE UTDB IN AWA                                       XDSTO
1787    2    CALL XREQ TO MAKE REQUEST                                                   XDSTO
1788    2    ERREXIT IF AWA OVERFLOW TO :ERR1:                                           XDSTO
1789    1 EXIT XDSTO                                                                     XDSTO

1790    2    :ERR1:                                                                      XDSTO
1791    2    CALL XRMSG TO DISPLAY MSGNO                                                 XDSTO
1792    2    GO TO :ERR4:                                                                XDSTO

1793    2    :ERR2:                                                                      XDSTO
```

```
1794    2     CALL XRMSG TO DISPLAY MSGNO                                                    XDSTO
1795    2     GO TO :ERR5:                                                                   XDSTO

1796    2     :ERR3:                                                                         XDSTO
1797    2     CALL XRMSG TO DISPLAY MSGNO WITH UTDB FILE ERROR CODE                          XDSTO

1798    2     :ERR4:                                                                         XDSTO
1799    2     CALL CLOSE TO CLOSE UTDB                                                       XDSTO
1800    2     CALL PURGE TO PURGE UTDB                                                       XDSTO

1801    2     :ERR5:                                                                         XDSTO
1802    2     CALL XRMSG TO DISPLAY STORE ABORTED/ UTDB NOT CREATED MESSAGE                  XDSTO
1803    1   END XDSTO                                                                        XDSTO
```

```
1805    1 CD0       FORTRAN CALLING PROCEDURE                                                    XDTOC
1806    1 CD0                                                                                    XDTOC
1807    1 CD0    CALL XDTOC                                                                      XDTOC
1808    1 CD0                                                                                    XDTOC
1809    1 C*********                                                                             XDTOC
1810    1 CD1                                                                                    XDTOC
1811    1 CD1       FDS AWA/DWA TABLE OF CONTENTS DIRECTIVE HANDLER.  XDTOC INTER-               XDTOC
1812    1 CD1       PRETS THE TOC DIRECTIVE, RETRIEVES THE INDICATED TOC (AWA OR                 XDTOC
1813    1 CD1       DATA BASE FILE) AND FORMATS AND OUPUTES THE REQUESTED ENTRIES.               XDTOC
1814    1 CD1                                                                                    XDTOC
1815    1 C*********                                                                             XDTOC
1816    1 CD2       INPUT                                                                        XDTOC
1817    1 CD2    XE COMMON - COMBUF, COMPTR, FLAGS, LU                                           XDTOC
1818    1 CD2    MANAGER   - AWA HEADER AND TOC (SEE MODULE XMAWA)                               XDTOC
1819    1 CD2    FILES     - DATA BASE FILES AS APPROPRIATE                                      XDTOC
1820    1 CD2                                                                                    XDTOC
1821    1 C*********                                                                             XDTOC
1822    1 CD3       OUTPUT                                                                       XDTOC
1823    1 CD3    XE COMMON - COMPTR, REQBUF, REQPTR                                              XDTOC
1824    1 CD3    PRINTED OUTPUT - AWA OR UTDB TOC DISPLAY                                        XDTOC
1825    1 CD3                                                                                    XDTOC
1826    1 C*********                                                                             XDTOC
1827    1 CD4       LOCAL VARIABLES                                                              XDTOC
1828    1 CD4    ASZ     - TOTAL ALLOCATABLE SIZE OF AWA (SEE MODULE XMAWA)                      XDTOC
1829    1 CD4    AWA     - FIRST PORTION OF AWA (HEADER AND TOC)                                 XDTOC
1830    1 CD4    CLASS   - ARRAY OF ALLOWABLE CLASS DESIGNATORS FOR TOC DIRECTIVE                XDTOC
1831    1 CD4    CLS     - CLASS CHAIN NUMBER BEING PROCESSED PLUS ONE                           XDTOC
1832    1 CD4    CODE    - CLASS NUMBERS CORRESPONDING TO ELEMENTS OF 'CLASS'                    XDTOC
1833    1 CD4    EOC     - END OF TOC CHAIN DESIGNATOR (-32768)                                  XDTOC
1834    1 CD4    FRE     - CURRENT AMOUNT OF FREE SPACE IN AWA (SEE MODULE XMAWA)                XDTOC
1835    1 CD4    HD      - ARRAY OF TOC CHAIN HEADS (SEE MODULE XMAWA)                           XDTOC
1836    1 CD4    IDIM    - TOC ENTRY DIM FIELD (WORD 8)                                          XDTOC
1837    1 CD4    LINE    - BUFFER FOR CONSTRUCTING CURRENT OUTPUT IMAGE                          XDTOC
1838    1 CD4    MULT    - ARRAY OF MULTIPLIERS USED TO DETERMINE NUMBER OF WORDS                XDTOC
1839    1 CD4              IN COLUMNS OF TWO DIMENSIONAL DATA ELEMENTS (FUNCTION OF              XDTOC
1840    1 CD4              DATA TYPE)                                                            XDTOC
1841    1 CD4    OPTION  - ARRAY OF SELECTED CHAINS TO LIST IN TOC DISPLAY                       XDTOC
1842    1 CD4    ORG     - ORIGIN ADDRESS OF AWA (SEE MODULE XMAWA)                              XDTOC
1843    1 CD4    POS     - VALUE OF 2 OR 22 INDICATING FIRST OR SECOND POSITION IN             XDTOC
1844    1 CD4              TOC DISPLAY LINE                                                      XDTOC
1845    1 CD4    SIZE    - TOC ENTRY SIZE FIELD (WORD 7)                                         XDTOC
1846    1 CD4    TOCMAX  - MAXIMUM SIZE OF TOC WHICH CAN BE TOTALLY ACCOMODATED BY              XDTOC
1847    1 CD4              INTERNAL BUFFER                                                       XDTOC
1848    1 CD4    TYPE    - TOC ENTRY SIZE FIELD (BYTE 2 OF WORD 1)                               XDTOC
1849    1 CD4    TYPEID  - ARRAY OF TYPE ID CODES FOR DE AND DRDE DISPLAYS                       XDTOC
1850    1 CD4    UNIT    - LOGICAL UNIT SELECTED FOR OUTPUT OF DISPLAY                           XDTOC
1851    1 CD4                                                                                    XDTOC
1852    1 C*********                                                                             XDTOC
1853    1 CD5       NOTES                                                                        XDTOC
1854    1 CD5    USES CLOSE, EXEC, IAND, IXOR, KCVT, OPEN, READF, XDSTA, XREQ,                   XDTOC
1855    1 CD5         XRI6, XRMOV, XRMSG, XRO6, XRQFN, XUDBG                                     XDTOC
1856    1 CD5                                                                                    XDTOC
1857    1 CD5    IN THE EVENT THE COMPACTED AWA TOC AND HEADER DATA WILL NOT FIT                 XDTOC
1858    1 CD5    IN THE ALLOCATED BUFFER SPACE, AS MUCH OF EACH CHAIN AS POSSIBLE                XDTOC
1859    1 CD5    WILL BE DISPLAYED UNTIL THE LINK FIELDS LEAD BEYOND THE BUFFER.                 XDTOC
1860    1 CD5                                                                                    XDTOC
1861    1 CD5    IT IS ASSUMED THAT XDTOC AND XDSTO USE THE SAME SIZE BUFFER FOR                 XDTOC
1862    1 CD5    TOC MANIPULATION; THUS, A DATA BASE FILE MAY NOT HAVE A TOC TOO                 XDTOC
1863    1 CD5    LARGE FOR THE XDTOC INTERNAL BUFFER.                                            XDTOC
```

XDIOC

1864 1 C*********

```
1866  1 BEGIN XDTOC                                                                                                    XDTOC
1867  2    SET FOR OUTPUT OF ALL USER CLASSES FROM AWA TO TERMINAL                                                     XDTOC
1868  2    IF NEXT TOKEN IS NOT EOS                                                                                    XDTOC
1869  2    THEN                                                                                                        XDTOC
1870  3       IF TOKEN IS A HYPHEN                                                                                     XDTOC
1871  3       THEN                                                                                                     XDTOC
1872  4          INCREMENT TO NEXT TOKEN                                                                               XDTOC
1873  4          IF TOKEN IS NOT A CHARACTER T                                                                         XDTOC
1874  4          THEN                                                                                                  XDTOC
1875  4          EXIT TO :ERR03: IF TOKEN IS NOT A CHARACTER P                                                         XDTOC
1876  5             SET OUTPUT UNIT FOR LINE PRINTER                                                                   XDTOC
1877  4          ENDIF                                                                                                 XDTOC
1878  4          INCREMENT TO NEXT TOKEN                                                                               XDTOC
1879  3       ENDIF                                                                                                    XDTOC
1880  3       IF TOKEN IS NOT EOS                                                                                      XDTOC
1881  3       THEN                                                                                                     XDTOC
1882  3       EXIT TO :ERR04: IF TOKEN IS NOT A COMMA                                                                  XDTOC
1883  4          INCREMENT TO NEXT TOKEN                                                                               XDTOC
1884  4          IF TOKEN IS NOT A COMMA                                                                               XDTOC
1885  4          THEN                                                                                                  XDTOC
1886  4          EXIT TO :ERR05: IF TOKEN DOES NOT INDICATE A VALID CLASS (D, F, I, S OR B)                            XDTOC
1887  5             SET OPTION FOR INDICATED CLASS                                                                     XDTOC
1888  5             INCREMENT TO NEXT TOKEN                                                                            XDTOC
1889  4          ENDIF                                                                                                 XDTOC
1890  4          IF TOKEN IS NOT EOS                                                                                   XDTOC
1891  4          THEN                                                                                                  XDTOC
1892  4          EXIT TO :ERR04: IF TOKEN IS NOT A COMMA                                                               XDTOC
1893  5             INCREMENT TO NEXT TOKEN                                                                            XDTOC
1894  4          EXIT TO :ERR06: IF TOKEN IS NOT A FOUR CHARACTER NAME                                                 XDTOC
1895  5             INCREMENT TO NEXT TOKEN                                                                            XDTOC
1896  4          EXIT TO :ERR04: IF TOKEN IS NOT EOS                                                                   XDTOC
1897  5             CALL XREQ TO RETRIEVE DATA BASE FILE TOC ENTRY                                                     XDTOC
1898  4          EXIT TO :ERR06: IF NOT LOGGED IN TOC                                                                  XDTOC
1899  5             GET TYPE AND SET NAME IN HEADER                                                                    XDTOC
1900  5             CONSTRUCT FILE NAME                                                                                XDTOC
1901  5             READ FIRST DATA BASE FILE TOC RECORD                                                               XDTOC
1902  5             INITIALIZE TOC HEAD TO APPEAR SIMILAR TO AWA TOC                                                   XDTOC
1903  5             READ SUBSEQUENT TOC RECORDS                                                                        XDTOC
1904  4          EXIT TO :ERR07: IF FILE ACCESS FAILS                                                                  XDTOC
1905  5             DO FOR EACH NON-EMPTY TOC CHAIN                                                                    XDTOC
1906  6                INDEX TO TOC ENTRY POINTED TO BY CHAIN HEAD                                                     XDTOC
1907  6                IF NOT FIRST ENTRY IN TABLE, I.E., A PREVIOUS NON-NULL CHAIN EXISTED                            XDTOC
1908  6                THEN                                                                                            XDTOC
1909  7                   MARK PREVIOUS ENTRY AS AN END OF CHAIN                                                       XDTOC
1910  6                ENDIF                                                                                           XDTOC
1911  5             ENDDO                                                                                              XDTOC
1912  5             DO FOR EACH ENTRY IN TOC                                                                           XDTOC
1913  6                IF NOT MARKED AS AN END OF CHAIN                                                                XDTOC
1914  6                THEN                                                                                            XDTOC
1915  7                   STORE POINTER TO NEXT SEQUENTIAL TOC ENTRY IN CHAIN POINTER FIELD                            XDTOC
1916  6                ENDIF                                                                                           XDTOC
1917  5             ENDDO                                                                                              XDTOC
1918  5             EXCLUDE CHAIN 8 (DATA BASE FILES) FROM DISPLAY                                                     XDTOC
1919  4          ENDIF                                                                                                 XDTOC
1920  3       ENDIF                                                                                                    XDTOC
1921  2    ENDIF                                                                                                       XDTOC
1922  2    IF REFERENCING AWA TOC                                                                                      XDTOC
1923  2    THEN                                                                                                        XDTOC
1924  3       CALL XREQ TO RETRIEVE AWA TOC                                                                            XDTOC
```

```
1925  3       IF DEBUG AND/OR TRACE FLAGS ARE SET                                                    XDTOC
1926  3       THEN                                                                                   XDTOC
1927  4          EXTEND OPTION TO INCLUDE SYSTEM CLASSES (0, 1, 5 AND 7)                             XDTOC
1928  3       ENDIF                                                                                  XDTOC
1929  2    ENDIF                                                                                     XDTOC
1930  2    OUTPUT TOC HEADER                                                                         XDTOC
1931  2    DO FOR EACH CLASS INDICATED BY OPTION                                                     XDTOC
1932  3       OUTPUT CLASS HEADER                                                                    XDTOC
1933  3       LOCATE CLASS CHAIN HEAD                                                                XDTOC
1934  3       DO UNTIL END OF CHAIN FOUND (-32768)                                                   XDTOC
1935  4          INDEX TO NEXT TOC ENTRY                                                             XDTOC
1936  4          IF CHAIN POINTS WITHIN LIMIT OF BUFFER                                              XDTOC
1937  4          THEN                                                                                XDTOC
1938  5             FORMAT NAME & SIZE FIELDS                                                        XDTOC
1939  5             IF DEBUG AND/OR TRACE FLAGS SET                                                  XDTOC
1940  5             THEN                                                                             XDTOC
1941  6                FORMAT I-DIM FIELD IN OCTAL                                                   XDTOC
1942  5             ENDIF                                                                            XDTOC
1943  5             FORMAT TYPE FIELD IN INTEGER                                                     XDTOC
1944  5             IF CLASS 2 OR 3                                                                  XDTOC
1945  5             THEN                                                                             XDTOC
1946  6                FORMAT TYPE FIELD USING DATA TYPE TABLE                                       XDTOC
1947  6                IF CLASS 2                                                                    XDTOC
1948  6                THEN                                                                          XDTOC
1949  7                   FORMAT I-DIM & J-DIM FIELDS                                                XDTOC
1950  6                ENDIF                                                                         XDTOC
1951  5             ELSE                                                                             XDTOC
1952  6                IF CALSS 8                                                                    XDTOC
1953  6                THEN                                                                          XDTOC
1954  7                   FORMAT TYPE FIELD USING FILE TYPE TABLE                                    XDTOC
1955  6                ENDIF                                                                         XDTOC
1956  5             ENDIF                                                                            XDTOC
1957  4          ELSE                                                                                XDTOC
1958  5             PRINT 'DATA LOST' MESSAGE                                                        XDTOC
1959  4          EXIT PROCESSING FOR THIS CHAIN                                                      XDTOC
1960  4          ENDIF                                                                               XDTOC
1961  4          PRINT ENTRY                                                                         XDTOC
1962  3       ENDDO                                                                                  XDTOC
1963  2    ENDDO                                                                                     XDTOC
1964  2    IF PROCESSING AWA TOC                                                                     XDTOC
1965  2    THEN                                                                                      XDTOC
1966  3       CALL XDSTA TO DISPLAY AWA USAGE STATISTICS                                             XDTOC
1967  2    ENDIF                                                                                     XDTOC
1968  1 EXIT XDTOC                                                                                   XDTOC

1969  2    :ERR03: EXIT WITH INVALID OUTPUT DEVICE ID                                                XDTOC

1970  2    :ERR04: EXIT WITH SYNTAX ERROR                                                            XDTOC

1971  2    :ERR05: EXIT WITH INVALID CLASS DESISNATOR                                                XDTOC

1972  2    :ERR06: EXIT WITH INVALID UTDB FILE NAME                                                  XDTOC

1973  2    :ERR07: EXIT WITH UTDB FILE ACCESS ERROR                                                  XDTOC
1974  1 END XDTOC
```

```
1976    1 CD**************                                                                  XDWRT
1977    1 CD0                                                                               XDWRT
1978    1 CD0       CALL XDWRT (DATBUF)                                                     XDWRT
1979    1 CD0                                                                               XDWRT
1980    1 CD**************                                                                  XDWRT
1981    1 CD1                                                                               XDWRT
1982    1 CD1       XDWRT WRITES INTERFACE TABLES, SEQUENCE TABLES, DATA ELEMENTS           XDWRT
1983    1 CD1       AND DRDE FILES OUT TO A      .                                          XDWRT
1984    1 CD1                                                                               XDWRT
1985    1 CD**************                                                                  XDWRT
1986    1 CD2                                                                               XDWRT
1987    1 CD2       INPUTS FROM CALLING SEQUENCE:                                           XDWRT
1988    1 CD2                                                                               XDWRT
1989    1 CD2             DATBUF - (INTEGER, 1480 WORDS) BUFFER USED TO READ                XDWRT
1990    1 CD2                      IN AWA RESIDENT DATA AND DRDE DATA                       XDWRT
1991    1 CD2                                                                               XDWRT
1992    1 CD**************                                                                  XDWRT
1993    1 CD4                                                                               XDWRT
1994    1 CD4       INTERNAL XB COMMON USED:                                                XDWRT
1995    1 CD4                                                                               XDWRT
1996    1 CD4             XB(151) ABFLG - (INTEGER, 1 WORD) ABORT FLAG                      XDWRT
1997    1 CD4             XB(153) MSGNO - (INTEGER, 1 WORD) MESSAGE NUMBER TO BE DISPLAYED  XDWRT
1998    1 CD4             XB(155) NOREQ - (INTEGER, 1 WORD) NUMBER OF REQUESTS IN REQBUF    XDWRT
1999    1 CD4             XB(158) TOTWRD- (INTEGER, 1 WORD) TOTAL # WORDS IN A FILE         XDWRT
2000    1 CD4             XB(162) DATREC- (INTEGER, 1 WORD) RECORD # WHERE DATA GOES NEXT   XDWRT
2001    1 CD4             XB(163) WRDNO - (INTEGER, 1 WORD) INDEX INTO DATBUF WHERE         XDWRT
2002    1 CD4                             NEXT DATA ELEMENT IS TO START                     XDWRT
2003    1 CD4             XB(164) UDBERR- (INTEGER, 1 WORD) UTDB FILE ERROR FLAG            XDWRT
2004    1 CD4             XB(165) DRDERR- (INTEGER, 1 WORD) DRDE FILE ERROR FLAG            XDWRT
2005    1 CD4             XB(169) DRDE3 - (INTEGER, 1 WORD) DRDE TYPE 3 FILE FLAG           XDWRT
2006    1 CD4             XB(170) TOCIND- (INTEGER, 1 WORD) TOC INDEX FOR CURRENT REQUEST   XDWRT
2007    1 CD4             XB(171) TOCPTR- (INTEGER, 1 WORD) CURRENT TOC INDEX               XDWRT
2008    1 CD4             XB(172) DRDFIL- (INTEGER, 3 WORDS) DRDE FILE NAME                 XDWRT
2009    1 CD4             XB(175) LEN   - (INTEGER, 1 WORD) LENGTH OF 1 COPY (DRDE TO UTDB) XDWRT
2010    1 CD4             XB(176) NOREC - (INTEGER, 1 WORD) NUMBER RECORDS IN 1 COPY        XDWRT
2011    1 CD4             XB(201) NOTOC - (INTESER, 1 WORD) NUMBER OF UTDB TOC ENTRIES      XDWRT
2012    1 CD4             XB(201) TOCBUF- (INTEGER, 1200 WORDS) UTDB TOC BUFFER             XDWRT
2013    1 CD4                                                                               XDWRT
2014    1 CD**************                                                                  XDWRT
2015    1 CD5                                                                               XDWRT
2016    1 CD5       COMMON USED:                                                            XDWRT
2017    1 CD5                                                                               XDWRT
2018    1 CD5             EQUIVALENCE          (XE(1),    LU    ),                        XDWRT
2019    1 CD5           + (XE(4),    FLAGS ),  (XE(19),   REQPTR),                        XDWRT
2020    1 CD5           + (XE(20),   REQBUF)                                              XDWRT
2021    1 CD5                                                                               XDWRT
2022    1 CD5       RTE ROUTINES USED:                                                      XDWRT
2023    1 CD5                                                                               XDWRT
2024    1 CD5             CLOSE, EXEC, KCVT, OPEN, READF, WRITF                             XDWRT
2025    1 CD5                                                                               XDWRT
2026    1 CD5       FDS ROUTINES USED:                                                      XDWRT
2027    1 CD5                                                                               XDWRT
2028    1 CD5             XREQ, XREXT, XRMOV, XRMSG,                                        XDWRT
2029    1 CD5             XRQFN, XRSFL, XRSFR                                               XDWRT
2030    1 CD5                                                                               XDWRT
2031    1 CD**************                                                                  XDWRT
```

```
2033  1 BEGIN XDWRT                                                                                                    XDWRT
2034  2    SET WRDNO = 1 ( WORD INDEX INTO DATREC WHERE ELEMENT BEGINS)                                                XDWRT
2035  2    SET NOREQ = 0 ( NUMBER OF AWA REQUESTS IN REQBUF)                                                           XDWRT
2036  2    DO FOR ALL UTDB TOC ENTRIES UNTIL CLASS IS DRDE                                                             XDWRT
2037  3       BUILD REQUEST FOR DATA FROM AWA                                                                          XDWRT
2038  3       INCREMENT NOREQ BY 1                                                                                     XDWRT
2039  3       IF REQUEST BUFFER IS FULL (NOREQ=8) THEN                                                                 XDWRT
2040  4          PERFORM REQDAT TO REQUEST DATA FROM MANAGER AND HANDLE OUTPUT TO FILE                                  XDWRT
2041  3       ENDIF                                                                                                    XDWRT
2042  2    ENDDO                                                                                                       XDWRT
2043  2    IF THERE ARE REMAINING REQUESTS (NOREQ>0) THEN                                                              XDWRT
2044  3       SET NEXT REQUEST TO SAY END OF REQUEST LIST                                                              XDWRT
2045  3       PERFORM REQDAT TO REQUEST DATA FROM MANAGER AND HANDLE OUTPUT TO FILE                                     XDWRT
2046  3       IF THERE IS A PARTIAL DATA RECORD LEFT (WRDNO>1) THEN                                                    XDWRT
2047  4          PERFORM WRITE TO OUTPUT DATA TO UTDB FILE                                                             XDWRT
2048  3       ENDIF                                                                                                    XDWRT
2049  2    ENDIF                                                                                                       XDWRT
2050  2    DO FOR EACH DRDE UTDB TOC ENTRY                                                                             XDWRT
2051  3       STORE DATREC IN UTDB TOC ENTRY                                                                           XDWRT
2052  3       CALL XRQFN TO CREATE FILE NAME                                                                           XDWRT
2053  3       IF DRDE FILE IS TYPE 3 THEN                                                                              XDWRT
2054  4          CALL OPEN TO OPEN FILE AS CORRECT TYPE                                                                XDWRT
2055  4          ERREXIT IF OPEN ERROR TO :ERR1:                                                                       XDWRT
2056  4          DO UNTIL EOF IS READ                                                                                  XDWRT
2057  5             CALL READF TO READ 1 RECORD                                                                        XDWRT
2058  5             ERREXIT IF READF ERROR TO :ERR1:                                                                   XDWRT
2059  5             STORE RECORD LENGTH AT FRONT AND REAR OF DATA                                                      XDWRT
2060  5             INCREMENT WRDNO BY LENGTH + 2                                                                      XDWRT
2061  5             IF THERE IS ENOUGH DATA TO WRITE (WRDNO>128) THEN                                                  XDWRT
2062  6                PERFORM WRITE TO OUTPUT DATA TO UTDB FILE                                                       XDWRT
2063  5             ENDIF                                                                                              XDWRT
2064  4          ENDDO                                                                                                 XDWRT
2065  4          IF THERE IS REMAINING DATA (WRDNO>1) THEN                                                             XDWRT
2066  5             PERFORM WRITE TO OUTPUT DATA TO UTDB FILE                                                          XDWRT
2067  4          ENDIF                                                                                                 XDWRT
2068  4          SET DATREC TO NEXT AVAILABLE RECORD FOR DATA                                                          XDWRT
2069  3       ELSE                                                                                                     XDWRT
2070  4          CALL OPEN TO OPEN FILE AS TYPE 1                                                                      XDWRT
2071  4          ERREXIT IF OPEN ERROR TO :ERR1:                                                                       XDWRT
2072  4          COMPUTE TOTAL SIZE OF FILE IN WORDS                                                                   XDWRT
2073  4          DO UNTIL ALL DATA IS COPIED TO UTDB (SIZE=0)                                                          XDWRT
2074  5             IF SIZE IS LESS THAN LENGTH TO BE WRITTEN THEN                                                     XDWRT
2075  6                SET LENGTH = SIZE                                                                               XDWRT
2076  5             ENDIF                                                                                              XDWRT
2077  5             CALL READF TO READ LENGTH DATA                                                                     XDWRT
2078  5             ERREXIT IF READF ERROR TO :ERR1:                                                                   XDWRT
2079  5             CALL WRITF TO WRITE LENGTH DATA                                                                    XDWRT
2080  5             ERREXIT IF WRITF ERROR TO :ERR3:                                                                   XDWRT
2081  5             INCREMENT DATREC BY NUMBER OF RECORDS WRITTEN                                                      XDWRT
2082  5             DECREMENT SIZE BY LENGTH IN WORDS WRITTEN                                                          XDWRT
2083  4          ENDDO                                                                                                 XDWRT
2084  3       ENDIF                                                                                                    XDWRT
2085  3       CALL CLOSE TO CLOSE DRDE FILE                                                                            XDWRT
2086  3       ERREXIT IF CLOSE ERROR TO :ERR1:                                                                         XDWRT
2087  2    ENDDO                                                                                                       XDWRT
2088  1 EXIT XDWRT                                                                                                     XDWRT
```

```
2090    2     BEGIN REQDAT                                                                                    XDWRT
2091    3        CALL XREQ TO MAKE REQUEST                                                                    XDWRT
2092    3        ERREXIT IF THERE IS AN AWA OVERFLOW TO :ERR4:                                                XDWRT
2093    3        DO FOR NUMBER OF REQUESTS (NOREQ) IN BUFFER                                                  XDWRT
2094    4           CALL EXEC TO GET DATA FROM SAM                                                            XDWRT
2095    4           SET DATREC AND WRDNO INTO THIS UTDB TOC ENTRY                                             XDWRT
2096    4           INCREMENT WRDNO BY SIZE OF THIS ELEMENT                                                   XDWRT
2097    4           IF THERE IS ENOUGH DATA TO WRITE (WRDNO>128) THEN                                         XDWRT
2098    5              PERFORM WRITE TO OUTPUT DATA TO UTDB FILE                                              XDWRT
2099    4           ENDIF                                                                                     XDWRT
2100    3        ENDDO                                                                                        XDWRT
2101    3        REINITIALIZE REQUEST BUFFER (NOREQ=0)                                                        XDWRT
2102    2     END REQDAT                                                                                      XDWRT
2103    2  *                                                                                                  XDWRT
2104    2  *                                                                                                  XDWRT
2105    2  *                                                                                                  XDWRT
2106    2     BEGIN WRITE                                                                                     XDWRT
2107    3        COMPUTE LENGTH TO WRITE IN WORDS AND RECORDS                                                 XDWRT
2108    3        IF THERE IS NOT AT LEAST 1 RECORD TO WRITE THEN                                              XDWRT
2109    4           SET LENGTH TO MINIMUM OF 1 RECORD                                                         XDWRT
2110    3        ENDIF                                                                                        XDWRT
2111    3        IF THIS IS A DRDE TYPE 3 FILE THEN                                                           XDWRT
2112    4           ERREXIT IF THERE ARE MORE BLOCKS THAN SPECIFIED TO :ERR2:                                 XDWRT
2113    3        ENDIF                                                                                        XDWRT
2114    3        CALL WRITF TO WRITE LENGTH DATA TO UTDB FILE                                                 XDWRT
2115    3        ERREXIT IF WRITF ERROR TO :ERR3:                                                             XDWRT
2116    3        DECREMENT WRDNO BY LENTH                                                                     XDWRT
2117    3        INCREMENT DATREC BY LENTH/128                                                                XDWRT
2118    3        IF THERE IS REMAINING DATA (WRDNO>1) THEN                                                    XDWRT
2119    4           CALL XRMOV TO MOVE REMAINING DATA UP IN BUFFER                                            XDWRT
2120    3        ENDIF                                                                                        XDWRT
2121    2     END WRITE                                                                                       XDWRT

2122    2     :ERR1:                                                                                          XDWRT
2123    2     CALL XRMSG WITH DRDE NAME AND RC                                                                XDWRT
2124    2     SET ABFLG TO SAY A DRDE ERROR                                                                   XDWRT
2125    2     GO TO :ERR3:                                                                                    XDWRT

2126    2     :ERR2:                                                                                          XDWRT
2127    2     CALL XRMSG TO SAY DRDE FILE LARGER THAN SPECIFIED IN TOC                                        XDWRT
2128    2     SET ABFLG TO SAY DRDE FILE ERROR                                                                XDWRT

2129    2     :ERR3:                                                                                          XDWRT
2130    2     CALL CLOSE TO CLOSE DRDE FILE                                                                   XDWRT
2131    2     SET ABFLG TO SAY UTDB FILE ERROR                                                                XDWRT
2132    1  EXIT XDWRT                                                                                         XDWRT

2133    2     :ERR4:                                                                                          XDWRT
2134    2     SET ABFLG TO SAY AWA OVERFLOW                                                                   XDWRT
2135    2     DO FOR REMAINING REQUESTS                                                                       XDWRT
2136    3        CALL EXEC TO FREE CLASS NUMBER                                                               XDWRT
2137    2     ENDDO                                                                                           XDWRT
2138    1  END XDWRT                                                                                          XDWRT
```

SYMBOL DEFINITION TABLE

```
:BATC   :   133
:CALGET:  1328
:CLASER:   377
:CLASER:  1331
:CLEA   :    86
:COPY   :    85
:DELE   :    83
:END    :   384
:ERR03  :  1969
:ERR04  :  1970
:ERR04  :  1521
:ERR04  :  1198
:ERR04  :  1623
:ERR05  :  1971
:ERR06  :  1199
:ERR06  :  1972
:ERR06  :  1621
:ERR07  :  1973
:ERR07  :  1200
:ERR08  :  1626
:ERR08  :  1522
:ERR09  :  1620
:ERR09  :  1201
:ERR09  :   779
:ERR09  :   573
:ERR09  :  1523
:ERR1   :  2122
:ERR1   :   678
:ERR1   :  1099
:ERR1   :  1790
:ERR1   :   977
:ERR10  :  1524
:ERR10  :   775
:ERR11  :   776
:ERR16  :  1202
:ERR16  :  1622
:ERR17  :  1624
:ERR18  :  1525
:ERR18  :  1203
:ERR19  :  1204
:ERR2   :   682
:ERR2   :   981
:ERR2   :  2126
:ERR2   :  1101
:ERR2   :  1793
:ERR20  :  1627
:ERR21  :  1205
:ERR21  :  1629
:ERR22  :  1206
:ERR23  :   217
:ERR24  :   233
:ERR3   :   689
:ERR3   :  2129
:ERR3   :   983
:ERR3   :  1796
:ERR33  :  1525
:ERR4   :  1798
:ERR4   :  2133
```

```
:ERR44  :  1207
:ERR48  :   232
:ERR5   :  1801
:FILERR:    381
:FILERR:    481
:INVALD:    380
:INVALD:   1332
:LIST   :     35
:LOAD   :    132
:MAXERR:    480
:MAXERR:    379
:NAMAGN:   1333
:NAMERR:   1330
:NAMERR:    378
:NAMERR:    479
:OFF    :     87
:RECA   :     82
:RENA   :     84
 REQDAT    2090
:REST   :    130
:RETURN:   1208
:RETURN:    780
:RETURN:   1630
:RETURN:    574
:RETURN:    482
:RTN1   :    650
:RTN2   :    671
:SAVE   :     81
:STOR   :    129
:SYNTAX:    376
:SYNTAX:   1326
:TELUSR:   1338
:TOC    :     80
:TOCERR:    382
:TOOLNG:   1327
:TYPERR:    383
:UNDO   :   1335
:UNLO   :    131
 WRITE     2106
 XDCLD       32
 XDCLE      188
 XDCLF       77
 XDCLU      126
 XDCOP      299
 XDDBA      445
 XDELE      534
 XDLIS      647
 XDLST      736
 XDOFF      829
 XDRDE      948
 XDRDF     1061
 XDREC     1144
 XDREN     1280
 XDREQ     1386
 XDRES     1476
 XDSAV     1572
 XDSTA     1677
 XDSTO     1737
 XDTOC     1866
```

XDUMT 2033

SXQT F.POLIST

C-2

```
 2    1 CDO      FORTRAN CALLING PROCEDURE                                                                    XECAL
 3    1 CDO                                                                                                   XECAL
 4    1 CDO   CALL XELDS (THREE WORD ARRAY CONTAINING 'XECAL ')                                               XECAL
 5    1 CDO                                                                                                   XECAL
 6    1 C*********                                                                                            XECAL
 7    1 CD1                                                                                                   XECAL
 8    1 CD1   FDS EXECUTIVE INITIALIZATION SEGMENT MAIN ROUTINE                                               XECAL
 9    1 CD1                                                                                                   XECAL
10    1 C*********                                                                                            XECAL
11    1 CD2      INPUT                                                                                        XECAL
12    1 CD2   COMMON XE - SUBSTA                                                                              XECAL
13    1 CD2                                                                                                   XECAL
14    1 C*********                                                                                            XECAL
15    1 CD3      OUTPUT                                                                                       XECAL
16    1 CD3   COMMON XB - INITIALIZED ACCORDING TO THE VALUE OF SUBSTA                                        XECAL
17    1 CD3               (SEE APPROPRIATE INITIALIZATION SUBROUTINE)                                         XECAL
18    1 CD3                                                                                                   XECAL
19    1 C*********                                                                                            XECAL
20    1 CD5      NOTES                                                                                        XECAL
21    1 CD5   USES XEIND, XEINT, XEINI, XEINS, XEINX, XERTN                                                   XECAL
22    1 CD5                                                                                                   XECAL
23    1 C*********                                                                                            XECAL
24    1 *                                                                                                     XECAL
25    1 *                                                                                                     XECAL
26    1 *                                                                                                     XECAL
27    1 *                                                                                                     XECAL
28    1 *                                                                                                     XECAL
29    1 BEGIN XECAL                                                                                           XECAL
30    2    CASE (:GLOBAL:, :DIRECT:, :EXECUT:, :SEQEDT:, :INTEDT:) SUBSTA (-1 TO 3)                           XECAL

31    3       :GLOBAL:  CALL XEINE TO INITIALIZE GLOBAL COMMON                                              XECAL

32    3       :DIRECT:  CALL XEIND TO INITIALIZE DYNAMIC COMMON FOR DIRECTIVES                              XECAL

33    3       :EXECUT:  CALL XEINX TO INITIALIZE DYNAMIC COMMON FOR EXECUTION CONTROL                       XECAL

34    3       :SEQEDT:  CALL XEINS TO INITIALIZE DYNAMIC COMMON FOR SEQUENCE EDITING                        XECAL

35    3       :INTEDT:  CALL XEINI TO INITIALIZE DYNAMIC COMMON FOR INTERFACE EDITING                       XECAL
36    2    ENDCASE                                                                                          XECAL
37    1 END XECAL                                                                                           XECAL
```

```
39    1 CD0      FORTRAN CALLING PROCEDURE                                                    XEIND
40    1 CD0                                                                                   XEIND
41    1 CD0   CALL XEIND                                                                      XEIND
42    1 CD0                                                                                   XEIND
43    1 C*********                                                                            XEIND
44    1 CD1                                                                                   XEIND
45    1 CD1   INITIALIZE XB COMMON FOR DIRECTIVE LEVEL                                        XEIND
46    1 CD1                                                                                   XEIND
47    1 C*********                                                                            XEIND
48    1 CD2      INPUT                                                                        XEIND
49    1 CD2   COMMON XE - CARTRG, FLAGS                                                       XEIND
50    1 CD2    XOPRM - FDS DIRECTIVE PROMPT FILE                                              XEIND
51    1 CD2                                                                                   XEIND
52    1 C*********                                                                            XEIND
53    1 CD3      OUTPUT                                                                       XEIND
54    1 CD3   COMMON XB - DIRECT, NUMDIR                                                      XEIND
55    1 CD3                                                                                   XEIND
56    1 C*********                                                                            XEIND
57    1 CD5      NOTES                                                                        XEIND
58    1 CD5   USES CLOSE, EXEC, OPEN, PRTN, READF, XREXT, XRIG, XRMSG, XVABN                  XEIND
59    1 CD5                                                                                   XEIND
60    1 C*********                                                                            XEIND
61    1 *                                                                                     XEIND
62    1 *                                                                                     XEIND
63    1 *                                                                                     XEIND
64    1 *                                                                                     XEIND
65    1 BEGIN XEIND                                                                           XEIND
66    2    READ FDS DIRECTIVE PROMPT FILE                                                     XEIND
67    2    STORE DIRECTIVES IN DYNAMIC COMMON                                                 XEIND
68    1 END XEIND                                                                             XEIND
```

```
 70    1 CD0      FORTRAN CALLING PROCEDURE                                                                XEINE
 71    1 CD0                                                                                               XEINE
 72    1 CD0   CALL XEINE                                                                                  XEINE
 73    1 CD0                                                                                               XEINE
 74    1 C*********                                                                                        XEINE
 75    1 CD1                                                                                               XEINE
 76    1 CD1   INITIALIZE XE COMMON FOR FDS START UP                                                       XEINE
 77    1 CD1                                                                                               XEINE
 78    1 C*********                                                                                        XEINE
 79    1 CD2      INPUT                                                                                    XEINE
 80    1 CD2   COMMON XE - CARTRG, FLAGS, LU, QUAL                                                         XEINE
 81    1 CD2    XLIBD - LIBRARY DIRECTORY FILE                                                             XEINE
 82    1 CD2     XPDB  - PDB AND MDB DIRECTORY                                                             XEINE
 83    1 CD2                                                                                               XEINE
 84    1 C*********                                                                                        XEINE
 85    1 CD3      OUTPUT                                                                                   XEINE
 86    1 CD3   COMMON XE - MASSTA, NOPROC, REQBUF, REQPTR, SUBSTA, TKNLNG, TOKENS                          XEINE
 87    1 CD3   COMMON XB - INITIALIZED FOR DIRECTIVE LEVEL (SEE XEIND)                                     XEINE
 88    1 CD3   AWA       - LIBRARY DIRECTORY NAME TABLE                                                    XEINE
 89    1 CD3               - MDB'S AND PDB'S LOGGED INTO AWA                                               XEINE
 90    1 CD3                                                                                               XEINE
 91    1 C*********                                                                                        XEINE
 92    1 CD4      LOCAL                                                                                    XEINE
 93    1 CD4   IDDSP  -  DISPLACEMENT (NEGATIVE) BETWEEN ORIGIN OF XE COMMON AND                           XEINE
 94    1 CD4             ID SEGMENT POINTER LIST                                                           XEINE
 95    1 CD4   KEYWRD -  ABSOLUTE ADDRESS OF ORIGIN OF ID SEGMENT LIST                                     XEINE
 96    1 CD4   LEN    -  LENGTH OF LIBRARY DIRECTORY RECORD BEING PROCESSED                                XEINE
 97    1 CD4   NAMDSP -  DISPLACEMENT (NEGATIVE) BETWEEN ORIGIN OF XE COMMON AND                          XEINE
 98    1 CD4             ID SEGMENT NAME FIELD OF ID BEING PROCESSED                                       XEINE
 99    1 CD4   ORG    -  MAPPED ADDRESS OF ORIGIN OF XE COMMON                                             XEINE
100    1 CD4                                                                                               XEINE
101    1 C*********                                                                                        XEINE
102    1 CD5      NOTES                                                                                    XEINE
103    1 CD5   USES CLOSE, EXEC, IAND, OPEN, PRTN, READF, XEIND, XRCPR, XREQ,                              XEINE
104    1 CD5        XREXT, XRI6, XRLOC, XRMOV, XRMSG, XUDBG, XVABN                                         XEINE
105    1 CD5                                                                                               XEINE
106    1 C*********                                                                                        XEINE
```

```
108   1 BEGIN XEINE                                                                                                    XEINE
109   2    INITIALIZE COMMON TO ZEROS                                                                                  XEINE
110   2    SET FILE CARTRIDGE NUMBER                                                                                   XEINE
111   2    SET STATES TO DIRECTIVE LEVEL                                                                               XEINE
112   2    SET TKNLNG                                                                                                  XEINE
113   2    INITIALIZE TOKENS                                                                                           XEINE
114   2    READ LIBRARY DIRECTORY FIRST RECORD                                                                         XEINE
115   1 EXIT TO :ERR01: IF READ FAILED                                                                                 XEINE
116   2    STORE NUMBER OF PROCESSORS IN NOPROC                                                                        XEINE
117   2    READ PROCESSOR NAME RECORD                                                                                  XEINE
118   1 EXIT TO :ERR01: IF READ FAILED                                                                                 XEINE
119   2    CALL XREQ TO ALLOCATE AND STORE PROCESSOR DIRECTORY IN AWA                                                   XEINE
120   1 EXIT TO :ERR01: IF REQUEST FAILED                                                                              XEINE
121   2    DO FOR EACH ID SEGMENT IN SYSTEM                                                                            XEINE
122   3       DO FOR EACH PROCESSOR IN LIBRARY DIRECTORY                                                               XEINE
123   4          IF NAMES ARE THE SAME                                                                                 XEINE
124   4          THEN                                                                                                  XEINE
125   5             NOTE EXISTANCE OF ID                                                                               XEINE
126   4          ENDIF                                                                                                 XEINE
127   3       ENDDO                                                                                                    XEINE
128   2    ENDDO                                                                                                       XEINE
129   2    IF ANY PROCESSORS NOT MARKED                                                                                XEINE
130   2    THEN                                                                                                        XEINE
131   3       LIST PROCESSORS MISSING ID SEGMENTS                                                                      XEINE
132   2    EXIT XEIND WITH INITIALIZATION FAILURE                                                                      XEINE
133   2    ENDIF                                                                                                       XEINE
134   2    CALL OPEN TO OPEN PDB/MDB DIRECTORY ( XPDB)                                                                 XEINE
135   1 EXIT TO :ERR01: IF OPEN FAILED                                                                                 XEINE
136   2    CALL READF TO READ 1ST RECORD OF  XPDB (LIST OF MDB'S)                                                      XEINE
137   1 EXIT TO :ERR01: IF READ FAILED                                                                                 XEINE
138   2    PERFORM DBLOG TO LOG MDB'S IN AWA                                                                           XEINE
139   2    CALCULATE RECORD NO. OF PDB'S FOR THIS QUALIFIER                                                            XEINE
140   2    CALL READF TO READ THAT RECORD OF  XPDB                                                                     XEINE
141   2    CALCULATE CORRECT INDEX INTO BUFFER (EACH RECORD IS FOR 2 QUALIFIERS)                                        XEINE
142   2    CALL CLOSE TO CLOSE XPDB                                                                                    XEINE
143   2    PERFORM DBLOG TO LOG PDB'S IN AWA                                                                           XEINE
144   2    IF THERE ARE REQUESTS IN THE AWA REQUEST BUFFER, THEN                                                       XEINE
145   3       CALL XREQ TO PROCESS THE AWA REQUESTS                                                                    XEINE
146   2    ENDIF                                                                                                       XEINE
147   2    CALL XEIND TO INITIALIZE DYNAMIC COMMON FOR DIRECTIVE LEVEL                                                 XEINE
148   1 EXIT XEINE                                                                                                     XEINE
149   2    BEGIN DBLOG                                                                                                 XEINE
150   3       DO WHILE THERE ARE DATA BASE FILES TO BE LOGGED                                                          XEINE
151   4          BUILD AN ENTRY IN AWA REQUEST BUFFER TO ALLOCATE THIS DATA BASE                                       XEINE
152   4          IF 8 AWA REQUESTS HAVE BEEN BUILT, THEN                                                               XEINE
153   5             CALL XREQ TO PROCESS AWA REQUESTS                                                                  XEINE
154   4          ENDIF                                                                                                 XEINE
155   3       EXIT TO :ERR01: IF A REQUEST FAILED                                                                      XEINE
156   3       ENDDO                                                                                                    XEINE
157   2    END DBLOG                                                                                                   XEINE

158   2    :ERR01: LIBRARY INITIALIZATION ERROR TERMINATION                                                            XEINE
159   1 END XEINE                                                                                                      XEINE
```

```
161   1 CD0      FORTRAN CALLING PROCEDURE                                                    XEINI
162   1 CD0                                                                                   XEINI
163   1 CD0   CALL XEINI                                                                      XEINI
164   1 CD0                                                                                   XEINI
165   1 C*********                                                                            XEINI
166   1 CD1                                                                                   XEINI
167   1 CD1   INITIALIZE XE AND XB COMMON FOR INTERFACE TABLE EDITING                         XEINI
168   1 CD1                                                                                   XEINI
169   1 C*********                                                                            XEINI
170   1 CD2      INPUT                                                                        XEINI
171   1 CD2   COMMON XE -                                                                     XEINI
172   1 CD2                MASSTA = MASTER STATE                                              XEINI
173   1 CD2                TOKENS = LEXICAL TOKEN VALUES                                      XEINI
174   1 CD2                NOPROC = NUMBER OF ENTRIES IN LIBRARY DIRECTORY                    XEINI
175   1 CD2                COMPTR = INDEX OF NEXT TOKEN IN COMBUF                             XEINI
176   1 CD2                COMBUF = COMMUNICATIONS BUFFER                                     XEINI
177   1 CD2                PRCNAM = PROCESSOR NAME FROM EXECUTION CONTROLLER                  XEINI
178   1 CD2   AWA       -                                                                     XEINI
179   1 CD2   LIBRARY   -                                                                     XEINI
180   1 CD2                INTNAM = OLD INTERFACE TABLE TO BE EDITED                          XEINI
181   1 CD2                <XXXXX = DEFAULT INTERFACE TABLE                                   XEINI
182   1 CD2                >XXXXX = PROMPT TABLE                                              XEINI
183   1 CD2                                                                                   XEINI
184   1 C*********                                                                            XEINI
185   1 CD3      OUTPUT                                                                       XEINI
186   1 CD3   COMMON XE -                                                                     XEINI
187   1 CD3                SUBSTA = SUBSTATE -- SET TO X IF INIT. ERROR                       XEINI
188   1 CD3                REQBUF = REQUEST BUFFER FOR AWA INPUTS                             XEINI
189   1 CD3                PRCNAM = PROCESSOR NAME ON INTE DIRECTIVE                          XEINI
190   1 CD3   COMMON XB -                                                                     XEINI
191   1 CD3                WKBLNG = LENGTH OF WKBUF (CONSTANT)                                XEINI
192   1 CD3                WKBUF  = WORKING BUFFER CONTAINING INTE. TABLE WITH                XEINI
193   1 CD3                         SHORT PROMPTS                                             XEINI
194   1 CD3                DIRECT = SUPPORTED INTE DIRECTIVES                                 XEINI
195   1 CD3                NUMDIR = NO. OF ENTRIES IN DIRECT                                  XEINI
196   1 CD3                NEWTAB = NEW INTE. TABLE NAME                                      XEINI
197   1 CD3                NUMARG = NO. OF ARGUMENTS                                          XEINI
198   1 CD3                NARG   = NO. OF WORDS IN SPC. AND HEADER OF WKBUF                  XEINI
199   1 CD3                LITLEN - NO. OF WORDS IN LITERAL AREA                              XEINI
200   1 CD3                LITPTR - INDEX TO START OF LITERAL AREA                            XEINI
201   1 CD3                ISIZES - ARRAY MAPPING ARG. TYPE TO EFFECTIVE LENGTH               XEINI
202   1 CD3                                                                                   XEINI
203   1 C*********                                                                            XEINI
204   1 CD5      NOTES                                                                        XEINI
205   1 CD5   USES FILES -                                                                    XEINI
206   1 CD5                <XXXXX - DEFAULT INTERFACE TABLE FOR THIS PROCESSER                XEINI
207   1 CD5                >XXXXX - PROMPT TABLE FOR THIS PROCESSOR                           XEINI
208   1 CD5   USES ROUTINES                                                                   XEINI
209   1 CD5                XRMOV          EXEC                                               XEINI
210   1 CD5                XRMSG          CLOSE                                              XEINI
211   1 CD5                XREXT          OPEN                                               XEINI
212   1 CD5                XREQ           READF                                              XEINI
213   1 CD5                XRUPK          XRNXB                                              XEINI
214   1 CD5                XRPCK                                                             XEINI
215   1 CD5                XRCPR                                                             XEINI
216   1 CD5                                                                                  XEINI
217   1 C*********                                                                           XEINI
218   1 BEGIN XEINI                                                                          XEINI
219   2    IF CALLED AS A RESULT INTE DIRECTIVE, THEN                                        XEINI
```

```
220  3      POSITION TO 1ST TOKEN AFTER 'INTE'                                                    XEINI
221  3      ERREXIT IF TOKEN IS NOT ',' :ERR1:                                                    XEINI
222  3      INCREMENT TO NEXT TOKEN                                                               XEINI
223  3      ERREXIT IF TOKEN IS NOT A NAME :ERR2:                                                 XEINI
224  3      SET PRCNAM TO THIS NAME                                                               XEINI
225  3      INCREMENT TO NEXT TOKEN                                                               XEINI
226  3      SET INTNAM TO 0                                                                       XEINI
227  3      SET NEWTAB TO '&INTAB'                                                                XEINI
228  3      IF TOKEN IS NOT EOS (END-OF-STATEMENT), THEN                                          XEINI
229  4         ERREXIT IF TOKEN IS NOT A COMMA, :ERR2:                                            XEINI
230  4         INCREMENT TO NEXT TOKEN                                                            XEINI
231  4         IF TOKEN IS NOT EOS, THEN                                                          XEINI
232  5            IF TOKEN IS A NAME, THEN                                                        XEINI
233  6               SET INTNAM TO THE NAME                                                       XEINI
234  6               INCREMENT TO NEXT TOKEN                                                      XEINI
235  5            ENDIF                                                                           XEINI
236  5            IF TOKEN IS NOT EOS, THEN                                                       XEINI
237  6               ERREXIT IF TOKEN IS NOT A COMMA :ERR2:                                       XEINI
238  6               INCREMENT TO NEXT TOKEN                                                      XEINI
239  6               IF TOKEN IS A NAME, THEN                                                     XEINI
240  7                  SET NEWTAB TO THE NAME                                                    XEINI
241  7                  INCREMENT TO NEXT TOKEN                                                   XEINI
242  6               ENDIF                                                                        XEINI
243  5            ENDIF                                                                           XEINI
244  5            ERREXIT IF TOKEN IS NOT EOS :ERR2:                                              XEINI
245  4         ENDIF                                                                              XEINI
246  3      ENDIF                                                                                 XEINI
247  2   ELSE                                                                                     XEINI
248  3      INTNAM AND PRCNAM ARE INITIALIZED BY THE EXECUTION CONTROLLER                         XEINI
249  3      NEWTAB IS SET TO '&INTAB'                                                             XEINI
250  3      ISSUE MESSAGE THAT INTERFACE TABLE EDITOR HAS BEEN INVOKED                            XEINI
251  2   ENDIF                                                                                    XEINI
252  2   MAKE MANAGER REQUEST FOR LIBRARY DIRECTORY FILE                                          XEINI
253  2   ERREXIT IF REQUEST IS UNSUCCESSFUL :ERR11:                                               XEINI
254  2   START SEARCH UNTIL ALL OF DIRECTORY IS SEARCHED, DO                                      XEINI
255  2   EXIT IF PROCESSOR PRCNAM IS FOUND                                                        XEINI
256  2   ORELSE                                                                                   XEINI
257  3      INCREMENT TO NEXT ENTRY                                                               XEINI
258  2   ENDLOOP                                                                                  XEINI
259  3      ERREXIT :ERR8:                                                                        XEINI
260  2   ENDSEARCH                                                                                XEINI
261  2   SET IVERS TO VERSION NO. OF DIRECTORY ENTRY FOUND                                        XEINI
262  2   IF INTNAM = 0, THEN                                                                      XEINI
263  3      CONSTRUCT THE NAME OF THE DEFAULT INTERFACE TABLE AS "<PRCNAM"                         XEINI
264  3      OPEN THE DEFAULT INTERFACE TABLE FILE                                                 XEINI
265  3      ERREXIT IF OPEN FAILED :ERR12:                                                        XEINI
266  3      READ THE DEFAULT INTERFACE TABLE                                                      XEINI
267  3      ERREXIT IF THE READ FAILED :ERR16:                                                    XEINI
268  3      READ LITERAL RECORD FROM DEFAULT INTERFACE TABLE FILE INTO BOTTOM                      XEINI
269  3      OF WKBUF                                                                              XEINI
270  3      ERREXIT IF READ FAILED :ERR14:                                                        XEINI
271  2   ELSE                                                                                     XEINI
272  3      MAKE MANAGER REQUEST FOR INTNAM INTERFACE TABLE                                       XEINI
273  3      ERREXIT IF ERROR OR COULD NOT FIND :ERR4:                                             XEINI
274  2   ENDIF                                                                                    XEINI
275  2   ERREXIT IF VERS .NE. VERSION NO. OF TABLE TO BE EDITED :ERR10:                           XEINI
276  2   CONSTRUCT NAME OF PROMPT TABLE AS ">PRCNAM"                                              XEINI
277  2   OPEN THE PROMPT TABLE                                                                    XEINI
278  2   ERREXIT IF OPEN FAILED :ERR13:                                                           XEINI
```

```
279   2    READ 2ND RECORD OF THIS PROMPT TABLE                                                    XEINI
280   2    ERREXIT IF READ FAILED :ERR17:                                                          XEINI
281   2    MOVE LITERAL DATA ENTRIES UP FROM BOTTOM OF WKBUF TO AREA FOLLOWING                     XEINI
282   2    THE PROMPT TABLE                                                                        XEINI
283   2    THIS IS DONE BY LOOPING THROUGH ALL ARGUMENTS TO FIND THE                               XEINI
284   2    LITERAL DISPL. THAT MATCH EACH LITERAL DATA ENTRY                                       XEINI
285   2    THE TYPE OF EACH ARGUMENT FOUND TO HAVE LITERAL DATA IS USED IN THE                     XEINI
286   2    MOVING PROCESS TO DETERMINE THE NUMBER OF WORDS IN EACH DATA ELEMENT                    XEINI
287   2    AS WELL AS THE NUMBER OF WORDS TO LEAVE FOR INCOMPLETE ELEMENTS.                        XEINI
288   1 EXIT XEINI                                                                                 XEINI

289   2    :ERR2: "SYNTAX ERROR ... "                                                              XEINI

290   2    :ERR4: "INTNAM" NOT FOUND                                                               XEINI

291   2    :ERR6: INVALID NAME FIELD                                                               XEINI

292   2    :ERR8: "PRCNAM" NOT IN DIRECTORY                                                        XEINI

293   2    :ERR10:  VERSION OF INTNAM DOES MATCH CURRENT VERSION                                   XEINI

294   2    :ERR11:  XEINI OP 1 INITIALIZATION ERROR                                                XEINI

295   2    :ERR12:  XEINI OP 2 INITIALIZATION ERROR                                                XEINI

296   2    :ERR13:  XEINI OP 3 INITIALIZATION ERROR                                                XEINI

297   2    :ERR14:  XEINI OP 4 INITIALIZATION ERROR                                                XEINI

298   2    :ERR15:  XEINI OP 5 INITIALIZATION ERROR                                                XEINI

299   2    :ERR16:  XEINI OP 6 INITIALIZATION ERROR                                                XEINI

300   2    :ERR17:  XEINI OP 7 INITIALIZATION ERROR                                                XEINI
301   2    SET SUBSTATE TO DIRECTIVE LEVEL TO INDICATE THE ERROR                                   XEINI
302   1 END XEINI                                                                                  XEINI
```

```
304   1 CD0
305   1 CD0       FORTRAN CALLING PROCEEDURE                                                    XEINS
306   1 CD0                                                                                     XEINS
307   1 CD0    CALL XEINS                                                                       XEINS
308   1 CD0                                                                                     XEINS
309   1 C*********                                                                              XEINS
310   1 CD1                                                                                     XEINS
311   1 CD1    INITIALIZE XE AND XB COMMON FOR SEQUENCE TABLE EDITING                           XEINS
312   1 CD1                                                                                     XEINS
313   1 C*********                                                                              XEINS
314   1 CD2                                                                                     XEINS
315   1 CD2       INPUT                                                                         XEINS
316   1 CD2                                                                                     XEINS
317   1 CD2    COMMON XE - CARTRG, COMBUF, COMPTR, LU, NOPROC, TOKENS                           XEINS
318   1 CD2                                                                                     XEINS
319   1 C*********                                                                              XEINS
320   1 CD3                                                                                     XEINS
321   1 CD3       OUTPUT                                                                        XEINS
322   1 CD3                                                                                     XEINS
323   1 CD3    COMMON XE - REQBUF, REQPTR, SUBSTA,                                              XEINS
324   1 CD3                                                                                     XEINS
325   1 CD3    COMMON XB - DEBUG,  DIRECT, NEWTAB, NUMDIR, NUMENT,                              XEINS
326   1 CD3                OLDTAB, PRMTMD, WKBLNG, WKBUF,  XLIBD                               XEINS
327   1 CD3                                                                                     XEINS
328   1 C*********                                                                              XEINS
329   1 CD5                                                                                     XEINS
330   1 CD5       NOTES                                                                         XEINS
331   1 CD5                                                                                     XEINS
332   1 CD5    USES FILES - >XSPRM - SEQUENCE TABLE EDITOR PROMPT FILE                          XEINS
333   1 CD5                                                                                     XEINS
334   1 CD5    USES ROUTINES                                                                    XEINS
335   1 CD5                                                                                     XEINS
336   1 CD5       EXEC                                                                          XEINS
337   1 CD5       IAND                                                                          XEINS
338   1 CD5       OPEN                                                                          XEINS
339   1 CD5       READF                                                                         XEINS
340   1 CD5       XRCPR                                                                         XEINS
341   1 CD5       XREQ                                                                          XEINS
342   1 CD5       XREXT                                                                         XEINS
343   1 CD5       XRMOV                                                                         XEINS
344   1 CD5       XRMSG                                                                         XEINS
345   1 CD5       XUDBG                                                                         XEINS
346   1 CD5                                                                                     XEINS
347   1 C*********                                                                              XEINS
```

```
350   1 BEGIN XEINS                                                          XEINS
351   2    BUILD AWA REQUEST TO RETRIEVE "XLIBD                              XEINS
352   2    ERREXIT IF TOKEN IS NOT COMMA  :ERR02:                            XEINS
353   2    INCREMENT TO NEXT TOKEN                                           XEINS
354   2    IF TOKEN IS NAME, THEN                                            XEINS
355   3       BUILD AWA REQUEST TO RETRIEVE OLDTAB                           XEINS
356   3       RETAIN THIS NAME AS OLDTAB                                     XEINS
357   3       INCREMENT TO NEXT TOKEN                                        XEINS
358   2    ELSE                                                              XEINS
359   3       SET OLDTAB TO ZERO                                             XEINS
360   2    ENDIF                                                             XEINS
361   2    ERREXIT IF TOKEN IS NOT COMMA  :ERR02:                            XEINS
362   2    INCREMENT TO NEXT TOKEN                                           XEINS
363   2    ERREXIT IF TOKEN IS NOT NAME :ERR02:                              XEINS
364   2    RETAIN THIS NAME AS NEWTAB                                        XEINS
365   2    IF NEWTAB NOT EQUAL TO OLDTAB, THEN                               XEINS
366   3       BUILD AWA REQUEST TO VERIFY EXISTENCE OF NEWTAB                 XEINS
367   2    ENDIF                                                             XEINS
368   2    CALL XREQ TO PROCESS AWA REQUEST(S)                               XEINS
369   2    IF NEWTAB NOT EQUAL TO OLDTAB, THEN                               XEINS
370   3       ERREXIT IF NO ERROR FROM XREQ  :ERR12:                         XEINS
371   2    ENDIF                                                             XEINS
372   2    IF OLDTAB NOT ZERO, THEN                                          XEINS
373   3       ERREXIT IF 2ND REQUEST (RETRIEVE OLDTAB) FAILED :ERR04:        XEINS
374   3       SET PROMPT MODE AS UPDATE                                      XEINS
375   3       SET NO. ENTRIES AS OLDTAB SIZE / 7                             XEINS
376   3       READ OLDTAB INTO WORKING BUFFER                                XEINS
377   3       ERREXIT IF FIRST REQUEST (RETRIEVE "XLIBD) FAILED :ERR01:      XEINS
378   3       READ "XLIBD INTO COMMON                                        XEINS
379   2    ELSE                                                              XEINS
380   3       SET PROMPT MODE AS CREATE                                      XEINS
381   3       SET NO. ENTRIES TO ZERO                                        XEINS
382   2    ENDIF                                                             XEINS
383   2    OPEN, READ AND CLOSE FILE >XSPRM                                  XEINS
384   2    SET COUNT AND SEQEDIT DIRECTIVES INTO XB COMMON                   XEINS
385   2    SET SUBSTATE FLAG TO SEQ. EDIT. (=2)                              XEINS
386   1 EXIT XEINS                                                           XEINS

387   2    :ERR01: CALL XRMSG - 'INITIALIZATION ERROR ...'                   XEINS

388   2    :ERR02: CALL XRMSG - 'SYNTAX ERROR'                               XEINS

389   2    :ERR04:  DEFAULT MESSAGE TO '...NOT FOUND'                        XEINS
390   2    IF ERROR WAS NO AWA SPACE THEN                                    XEINS
391   3       SET MSG TO '...NO AWA SPACE'                                   XEINS
392   2    ENDIF                                                             XEINS
393   2    CALL XRMSG TO DISPLAY MESSAGE                                     XEINS

394   2    :ERR12: CALL XRMSG - '... ALREADY EXISTS'                         XEINS
395   1 END XEINS                                                            XEINS
```

```
397   1 CD4       FORTRAN CALLING PROCEEDURE                                                    XEINT
398   1 CD5                                                                                     XEINT
399   1 CD6     CALL XEINT                                                                      XEINT
400   1 CD0                                                                                     XEINT
401   1 C*********                                                                              XEINT
402   1 CD1                                                                                     XEINT
403   1 CD1     INTERFACE TABLE LITERAL AREA INITIALIZATION                                     XEINT
404   1 CD1                                                                                     XEINT
405   1 C*********                                                                              XEINT
406   1 CD2                                                                                     XEINT
407   1 CD2       INPUT                                                                         XEINT
408   1 CD2                                                                                     XEINT
409   1 CD2     COMMON XB - LITPTR, NUMARG, WKBLNG, WKBUF                                       XEINT
410   1 CD2                                                                                     XEINT
411   1 CD2                 NOTE: WKBUF IS INPUT WITH THE INTERFACE                           XEINT
412   1 CD2                       TABLE'S CHARACTERISTICS IN THE TOP                          XEINT
413   1 CD2                       AND THE "PACKED" LITERAL AREA IN THE                        XEINT
414   1 CD2                       BOTTOM.                                                     XEINT
415   1 CD2                                                                                     XEINT
416   1 C*********                                                                              XEINT
417   1 CD3                                                                                     XEINT
418   1 CD3       OUTPUT                                                                        XEINT
419   1 CD3                                                                                     XEINT
420   1 CD3     COMMON XB - LITPTR, LITDWN, NARG, WKBUF                                         XEINT
421   1 CD3                                                                                     XEINT
422   1 CD3                 NOTE: WKBUF IS OUTPUT WITH THE LITERAL AREAS                      XEINT
423   1 CD3                       IN THEIR "UNPACKED" FORM.                                   XEINT
424   1 CD3                                                                                     XEINT
425   1 C*********                                                                              XEINT
426   1 CD5                                                                                     XEINT
427   1 CD5       NOTES                                                                         XEINT
428   1 CD5                                                                                     XEINT
429   1 CD5     USES ROUTINES                                                                   XEINT
430   1 CD5                                                                                     XEINT
431   1 CD5       XIEXT                                                                         XEINT
432   1 CD5       XRMOV                                                                         XEINT
433   1 CD5       XRNXB                                                                         XEINT
434   1 CD5       XRMSG                                                                         XEINT
435   1 CD5       XRSET                                                                         XEINT
436   1 CD5                                                                                     XEINT
437   1 C*********                                                                              XEINT
```

```
439   1 BEGIN XEINT                                                                                                    XEINT
440   2    DO UNTIL ALL LITERAL ENTRIES HAVE BEEN PROCESSED                                                            XEINT
441   3       START SEARCH UNTIL ALL ARGUMENTS SEARCHED                                                                XEINT
442   3       EXITIF 'DISP' FIELD FOR ARGUMENT = DISP. OF LITEAL ENTRY, AND                                            XEINT
443   4          S-BIT IS ON                                                                                           XEINT
444   4          BUILD 3 WORD LITERAL ENTRY (3, I-SUB, J-SUB)                                                          XEINT
445   3       EXITIF 'DISP' FIELD FOR ARGUMENT = DISP. OF LITERAL ENTRY, AND                                           XEINT
446   4          D-BIT IS ON                                                                                           XEINT
447   4          IF ARGUMENT IS COMPLETE (C-BIT ON), THEN                                                              XEINT
448   5             CREATE BIT MASK WORD(S) IN NEW LITERAL ENTRY AREA                                                  XEINT
449   4          ELSE                                                                                                  XEINT
450   5             MOVE BIT MASK WORDS UP TO NEW LITERAL ENTRY AREA                                                   XEINT
451   4          ENDIF                                                                                                 XEINT
452   4          DO UNTIL ALL BITS OF BIT MASK PROCESSED                                                               XEINT
453   5             DETERMINE NUMBER OF CONTIGUOUS BITS ON (OR OFF)                                                    XEINT
454   5             AND MOVE CORRESPONDING NUMBER OF DATA WORDS                                                        XEINT
455   5             (OR O'S) INTO NEW LITERAL ENTRY AREA                                                               XEINT
456   4          ENDDO                                                                                                 XEINT
457   3       ORELSE                                                                                                   XEINT
458   4          INCREMENT TO NEXT ARGUMENT                                                                            XEINT
459   3       ENDLOOP                                                                                                  XEINT
460   4          ERREXIT IF NOT CALLED BY LIBRARY MAINTANENCE  :ERR5:                                                  XEINT
461   4          SKIP TO NEXT LITERAL ENTRY                                                                            XEINT
462   3       ENDSEARCH                                                                                                XEINT
463   2    ENDDO                                                                                                       XEINT
464   1 EXIT TO :RETURN:                                                                                               XEINT
                                                                                                                       XEINT
465   2    :ERR5: CALL XRMSG - 'SYSTEM INITIALIZATION ERR 5 '                                                          XEINT
                                                                                                                       XEINT
466   2    :RETURN:                                                                                                    XEINT
467   1 END XEINT                                                                                                      XEINT
                                                                                                                       XEINT
```

```
469    1 CDC      FORTRAN CALLING PROCEDURE                                                                  XEINX
470    1 CDC                                                                                                 XEINX
471    1 CDC    CALL XEINX                                                                                   XEINX
472    1 CDC                                                                                                 XEINX
473    1 C*********                                                                                          XEINX
474    1 CD1                                                                                                 XEINX
475    1 CD1    INITIALIZE XE AND XB COMMON FOR EXECUTION CONTROLER                                          XEINX
476    1 CD1                                                                                                 XEINX
477    1 C********                                                                                           XEINX
478    1 CD2      INPUT                                                                                      XEINX
479    1 CD2    COMMON XE - COMBUF, COMPTR, FLAGS, LU, MASSTA, NOPROC, TOKENS                                XEINX
480    1 CD2    AVA       - SEQUENCE TABLE, LIBRARY DIRECTORY NAME TABLE                                     XEINX
481    1 CD2                                                                                                 XEINX
482    1 C*********                                                                                          XEINX
483    1 CD3      OUTPUT                                                                                     XEINX
484    1 CD3    COMMON XE - COMPTR, MASSTA, REQBUF, REQPTR, SEQEND, SEQNAM, SEQPTR                           XEINX
485    1 CD3                SEQSTR, SUBSTA                                                                 XEINX
486    1 CD3    COMMON XB - LIBD, NOPRC2, SEQLNG, WKBLNG, WKBUF                                              XEINX
487    1 CD3                                                                                                 XEINX
488    1 C*********                                                                                          XEINX
489    1 CD5      NOTES                                                                                      XEINX
490    1 CD5    USES EXEC, PRTN, XREQ, XREXT, XRI6, XRMOV, XRMSG, XUDEG, XVABN                               XEINX
491    1 CD5                                                                                                 XEINX
492    1 C*********                                                                                          XEINX
```

```
494   1 BEGIN XEINX                                                                                                    XEINX
495   2    IF INITIALIZATION FROM DIRECTIVE                                                                            XEINX
496   2    THEN                                                                                                        XEINX
497   3       IF DIRECTIVE IS MANU                                                                                     XEINX
498   3       THEN                                                                                                     XEINX
499   3       EXIT TO :ERR02: IF NEXT TOKEN IS NOT EOS                                                                 XEINX
500   3       ELSE                                                                                                     XEINX
501   4          IF DIRECTIVE IS AUTO                                                                                  XEINX
502   4          THEN                                                                                                  XEINX
503   5             IF TOKEN IS A HYPHEN                                                                               XEINX
504   5             THEN                                                                                               XEINX
505   6                INCREMENT TO NEXT TOKEN                                                                         XEINX
506   5             EXIT TO :ERR03: IF TOKEN IS NOT THE NAME 'T'                                                       XEINX
507   6                CHANGE EXECUTION CONTROL MODE TO AUTO-T                                                         XEINX
508   6                INCREMENT TO NEXT TOKEN                                                                         XEINX
509   5             ENDIF                                                                                              XEINX
510   4          ENDIF                                                                                                 XEINX
511   3       EXIT TO :ERR02: IF NEXT TOKEN IS NOT A COMMA                                                             XEINX
512   4          INCREMENT TO NEXT TOKEN                                                                               XEINX
513   3       EXIT TO :ERR02: IF NEXT TOKEN IS NOT A NAME                                                              XEINX
514   4          STORE NAME IN SEQNAM                                                                                  XEINX
515   4          CALL XREQ TO RETRIEVE SEQUENCE TABLE                                                                  XEINX
516   3       EXIT TO :ERR04: IF NON-ZERO RETURN CODE                                                                  XEINX
517   4          INCREMENT TO NEXT TOKEN                                                                               XEINX
518   4          SET SEQSTR TO FIRST SEQUENCE NUMBER                                                                   XEINX
519   4          SET SEQEND TO LAST SEQUENCE NUMBER                                                                    XEINX
520   4          IF TOKEN NOT EOS                                                                                      XEINX
521   4          THEN                                                                                                  XEINX
522   4          EXIT TO :ERR02: IF TOKEN NOT A COMMA                                                                  XEINX
523   5             INCREMENT TO NEXT TOKEN                                                                            XEINX
524   5             IF TOKEN IS AN INTEGER                                                                             XEINX
525   5             THEN                                                                                               XEINX
526   6                STORE STARTING RANGE NUMBER                                                                     XEINX
527   6                SEARCH SEQUENCE NUMBERS FOR STARTING VALUE                                                      XEINX
528   5             EXIT TO :ERR13: IF NOT FOUND                                                                       XEINX
529   6                INCREMENT TO NEXT TOKEN                                                                         XEINX
530   5             ENDIF                                                                                              XEINX
531   5             IF TOKEN NOT EOS                                                                                   XEINX
532   5             THEN                                                                                               XEINX
533   5             EXIT TO :ERR02: IF TOKEN NOT A COMMA                                                               XEINX
534   6                INCREMENT TO NEXT TOKEN                                                                         XEINX
535   5             EXIT TO :ERR02: IF TOKEN NOT AN INTEGER                                                            XEINX
536   6                ERREXIT TO :ERR05: IF ENDING SEQ # < BEGINNING SEQ #                                            XEINX
537   6                SEARCH SEQUENCE NUMBERS FOR ENDING VALUE                                                        XEINX
538   5             EXIT TO :ERR13: IF NOT FOUND                                                                       XEINX
539   6                INCREMENT TO NEXT TOKEN                                                                         XEINX
540   5             EXIT TO :ERR02: IF TOKEN NOT EOS                                                                   XEINX
541   5             ENDIF                                                                                              XEINX
542   4          ENDIF                                                                                                 XEINX
543   4          SET SEQPTR TO SEQSTR                                                                                  XEINX
544   3       ENDIF                                                                                                    XEINX
545   2    ENDIF                                                                                                       XEINX
546   2    INITIALIZE DYNAMIC COMMON WITH NUMBER OF PROCESSORS AND DIRECTORY NAME TABLE                                XEINX
547   1 EXIT TO :ERR01: IF INITIALIZATION FAILS                                                                        XEINX
548   1 EXIT XEINX                                                                                                     XEINX

549   2    :ERR01: INITIALIZATION FAILURE TERMINATION                                                                  XEINX

550   2    :ERR02: SET SUBSTA TO DIRECTIVE LEVEL & EXIT WITH SYNTAX ERROR                                              XEINX
```

```
551   2     :ERR03: SET SUBSTA TO DIRECTIVE LEVEL & EXIT WITH INVALID TRACE OPTION                    XEINX

552   2     :ERR04:  SET SUBSTA TO DIRECTIVE LEVEL                                                    XEINX
553   2     IF ERROR WAS NO AWA SPACE THEN                                                            XEINX
554   2     EXIT WITH NO AWA SPACE ERROR                                                              XEINX
555   2     ELSE                                                                                      XEINX
556   2     EXIT WITH SEQUENCE TABLE NOT FOUND ERROR                                                  XEINX
557   2     ENDIF                                                                                     XEINX

558   2     :ERR05: SET SUBSTA TO DIRECTIVE LEVEL & EXIT WITH RANGE ERROR                             XEINX

559   2     :ERR13: SET SUBSTA TO DIRECTIVE LEVEL & EXIT WITH NUMBER NOT FOUND                        XEINX
560   1 END XEINX                                                                                     XEINX
```

```
562   1 CD************                                                    XELDS
563   1 CD0                                                              XELDS
564   1 CD0       FORTRAN CALLING PROCEDURE:                             XELDS
565   1 CD0                                                              XELDS
566   1 CD0             CALL XELDS (SEGNM)   TO CALL SEGMENT FROM MAIN   XELDS
567   1 CD0             CALL XERTN           TO RETURN TO MAIN PROGRAM   XELDS
568   1 CD0                                                              XELDS
569   1 CD************                                                    XELDS
570   1 CD1                                                              XELDS
571   1 CD1       XELDS ALLOWS A MAIN PROGRAM TO "CALL" A SEGMENT AND    XELDS
572   1 CD1       THE ENTRY POINT XERTN PASSES CONTROL BACK TO THE MAIN PROGRAM  XELDS
573   1 CD1                                                              XELDS
574   1 CD************                                                    XELDS
575   1 CD2                                                              XELDS
576   1 CD2       INPUTS IN CALLING SEQUENCE:                            XELDS
577   1 CD2                                                              XELDS
578   1 CD2             SEGNM - (INTEGER, 3 WORDS) ARRAY CONTAINING NAME OF THE  XELDS
579   1 CD2                     SEGMENT TO BE LOADED                     XELDS
580   1 CD2                                                              XELDS
581   1 CD************                                                    XELDS
582   1 CD5                                                              XELDS
583   1 CD5       SUBROUTINES AND FUNCTIONS CALLED:                      XELDS
584   1 CD5             EXEC                                             XELDS
585   1 CD5                                                              XELDS
586   1 CD5       NOTES                                                  XELDS
587   1 CD5           1) XELDS CONTAINS 2 ENTRY POINTS: XELDS AND XERTN  XELDS
588   1 CD5           2) SEGMENT CALLED BY MAIN MUST BEGIN WITH A PROGRAM  XELDS
589   1 CD5              STATEMENT - TYPE 5                              XELDS
590   1 CD5           3) IN ORDER TO RETURN TO MAIN, A SEGMENT MUST "CALL XERTN"  XELDS
591   1 CD5              FOLLOWED BY A CALL TO MAIN WHICH IS NOT EXECUTED  XELDS
592   1 CD5           4) WHEN LOADER IS RUN, THE FOURTH PARAMETER MUST BE 1  XELDS
593   1 CD5              INDICATING TO LOAD MAIN PLUS SEGMENTS           XELDS
594   1 CD5                                                              XELDS
595   1 CD***                                                            XELDS
596   1 *                                                                XELDS
597   1 *                                                                XELDS
598   1 *                                                                XELDS
599   1 BEGIN XELDS                                                      XELDS

600   2     :XELDS:                                                      XELDS
601   2     CALL .ENTR TO RETRIEVE PARAMETERS AND RETURN ADDRESS         XELDS
602   2     MOVE THE ADDRESS OF THE SEGMENT NAME INTO EXEC PARAMETER LIST  XELDS
603   2     CALL EXEC TO LOAD AND PASS CONTROL TO SEGMENT                XELDS

604   2     :XERTN:                                                      XELDS
605   2     RETURN TO MAIN PROGRAM                                       XELDS
606   1 END XELDS                                                        XELDS
```

```
608   1 CD0      FORTRAN CALLING PROCEDURE                                                    XESCN
609   1 CD0                                                                                   XESCN
610   1 CD0   CALL XESCN                                                                      XESCN
611   1 CD0                                                                                   XESCN
612   1 C*********                                                                            XESCN
613   1 CD1                                                                                   XESCN
614   1 CD1   XESCN IS CALLED BY XEXEC AFTER COMPLETION OF ALL AUTOMATIC AND                  XESCN
615   1 CD1   SEMI-AUTOMATIC EXECUTIONS TO ASSURE THAT ALL DO AND SCAN PROCESSOR              XESCN
616   1 CD1   CONTROL DATA IS PURGED FROM THE AWA AND ALL INCOMPLETE DATA BOX                 XESCN
617   1 CD1   FILES ARE PURGED FROM THE SYSTEM.                                               XESCN
618   1 CD1                                                                                   XESCN
619   1 C*********                                                                            XESCN
620   1 CD2      INPUT                                                                        XESCN
621   1 CD2   XE COMMON - CARTRG                                                              XESCN
622   1 CD2   AWA       - &DOSTK, &SCNTB (SEE DO AND SCAN PROCESSORS                          XESCN
623   1 CD2                                                                                   XESCN
624   1 C*********                                                                            XESCN
625   1 CD3      OUTPUT                                                                       XESCN
626   1 CD3   XE COMMON - REQBUF, REQPTR                                                      XESCN
627   1 CD3   XB COMMON - SCRATCH                                                             XESCN
628   1 CD3                                                                                   XESCN
629   1 C*********                                                                            XESCN
630   1 CD5                                                                                   XESCN
631   1 CD5      EXTERNAL ROUTINES                                                            XESCN
632   1 CD5   CLOSE, EXEC, PURGE, XREQ, XRI6, XRMOV, XRUNQ                                    XESCN
633   1 C*********                                                                            XESCN
634   1 *                                                                                     XESCN
635   1 *                                                                                     XESCN
636   1 *                                                                                     XESCN
637   1 *                                                                                     XESCN
638   1 BEGIN XESCN                                                                           XESCN
639   2    BUILD REQUESTS TO DELETE &DOSTK AND RETRIEVE &SCNTB THEN DELETE IT                 XESCN
640   2    CALL XREQ TO ATTEMPT REQUESTS                                                      XESCN
641   2    ISSUE MESSAGE XE18 FOR SUCCESSFUL DELETES                                          XESCN
642   2    IF XREQ COMPLETED REQUESTS, I.E., &SCNTB EXISTED, THEN                             XESCN
643   3       DO FOR EACH SCAN CONTROL ENTRY IN &SCNTB                                        XESCN
644   4          CALL CLOSE TO CLOSE THE DATA BOX FILE ASSOCIATED WITH THIS SCAN              XESCN
645   4          CALL PURGE TO PURGE THE FILE                                                 XESCN
646   4          IF PURGE RETURNED AN ERROR, THEN                                             XESCN
647   5             CALL XRMSG TO DISPLAY WARNING MESSAGE                                     XESCN
648   4          ENDIF                                                                        XESCN
649   4          BUILD REQUEST TO DELETE DATA BOX FROM TOC                                    XESCN
650   3       ENDDO                                                                           XESCN
651   3       CALL XREQ TO DELETE DATA BOXES FROM AWA TOC                                     XESCN
652   2    ENDIF                                                                              XESCN
653   1 END XESCN                                                                             XESCN
```

```
1 CD1     FDS EXECUTIVE TASK MAIN PROGRAM.  SCHEDULED BY FDS MANAGER.
1 CD1     ONE PROGRAM PER SIGNED-ON USER
1 CD1
1 C*********
1 CD2        INPUT
1 CD2     SCHEDULING PARAMETERS - LU, CLASNO, QUAL, FLAGS (SEE XE COMMON)
1 CD2
1 C*********
1 CD3        OUTPUT
1 CD5     COMMON XE - COMBUF, COMPTR, MASSTA, SUBSTA, PLUS XEINE
1 CD5                 INITIALIZATION
1 CD5     COMMON XB - INITIALIZATIONS FROM XEIND, XEINI, XEINS, XEINX
1 CD3
1 C*********
1 CD5        NOTES
1 CD5     USES RMPAR, XDCLD, XDCLF, XELDS, XINTE, XRCPR, XRMSG, XSEQE,
1 CD5          XTCOM, XXCNT
1 CD5
1 CD5     THE CALLS TO XELDS PROVIDE LINKAGE TO THE INITIALIZATION SEGMENT
1 CD5     XECAL AND DIRECTIVE SEGMENTS XDCLD AND XDCLF.
1 CD5
1 CD5     THE LOOP STRUCTURE ASSOCIATED WITH EXECUTION CONTROL OCCURS
1 CD5     BECAUSE OF PARTITION SIZE LIMITATIONS WHICH PROHIBIT XXCNT FROM
1 CD5     CALLING XSEQE AND XINTE DIRECTLY.  LOGIC FLOW BETWEEN THESE
1 CD5     MODULES IS GOVERNED BY THE VALUE OF SUBSTA.  CYCLING TERMINATES
1 CD5     WHEN MASSTA IS SET TO THE DIRECTIVE LEVEL.
1 CD5
1 C*********
```

```
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
XEXEC
```

```
685   1 BEGIN XEXEC                                                                                                    XEXEC
686   2    RETRIEVE SCHEDULING PARAMETERS AND SET LU, CLASNO, QUAL & FLAGS                                             XEXEC
687   2    CALL XEINE TO INITIALIZE GLOBAL COMMON                                                                      XEXEC
688   2    DO FOREVER  -- TERMINATES INSIDE HANDLER FOR XOFF                                                           XEXEC
689   3       CALL XTCOM FOR INPUT OF DIRECTIVE                                                                        XEXEC
690   3       IF ERROR OR NOT A VALID DIRECTIVE NAME                                                                   XEXEC
691   3       THEN                                                                                                     XEXEC
692   4          ISSUE MESSAGE E06                                                                                     XEXEC
693   3       ELSE                                                                                                     XEXEC
694   4          IF NAME IS INTE                                                                                       XEXEC
695   4          THEN                                                                                                  XEXEC
696   5             SET STATES TO INTE LEVEL                                                                           XEXEC
697   5             CALL XEINI TO INITIALIZE DYNAMIC COMMON                                                            XEXEC
698   4          EXIT TO :RESET: IF ERROR                                                                              XEXEC
699   5             CALL XINTE TO EDIT TABLE                                                                           XEXEC
700   4          ELSE                                                                                                  XEXEC
701   5             IF NAME IS SEQE                                                                                    XEXEC
702   5             THEN                                                                                               XEXEC
703   6                SET STATES TO SEQE LEVEL                                                                        XEXEC
704   6                CALL XEINS TO INITIALIZE DYNAMIC COMMON                                                         XEXEC
705   5             EXIT TO :RESET: IF ERROR                                                                           XEXEC
706   6                CALL XSEQE TO EDIT TABLE                                                                        XEXEC
707   5             ELSE                                                                                               XEXEC
708   6                IF NAME IS FOR SOME EXECUTION CONTROL OPTION                                                    XEXEC
709   6                THEN                                                                                            XEXEC
710   7                   SET STATES TO APPROPRIATE EXECUTION CONTROL MODE                                             XEXEC
711   7                   DO UNTIL MASSTA IS AT DIRECTIVE LEVEL                                                        XEXEC
712   8                      CALL XEINX TO INITIALIZE DYNAMIC COMMON                                                   XEXEC
713   7                   EXIT TO :RESET: IF ERROR                                                                     XEXEC
714   8                      CALL XXCNT TO PERFORM EXECUTIONS                                                          XEXEC
715   8                      IF SUBSTA IS SET TO SEQE LEVEL                                                            XEXEC
716   8                      THEN                                                                                      XEXEC
717   9                         CALL XEINS TO REINITIALIZE DYNAMIC COMMON                                              XEXEC
718   8                      EXIT TO :RESET: IF ERROR                                                                  XEXEC
719   9                         CALL XSEQE TO SUPPORT EXECUTION CONTROL                                                XEXEC
720   8                      ENDIF                                                                                     XEXEC
721   8                      IF SUBSTA IS SET TO INTE LEVEL                                                            XEXEC
722   8                      THEN                                                                                      XEXEC
723   9                         CALL XEINI TO REINITIALIZE DYNAMIC COMMON                                              XEXEC
724   8                      EXIT TO :RESET: IF ERROR                                                                  XEXEC
725   9                         CALL XINTE TO SUPPORT EXECUTION CONTROL                                                XEXEC
726   8                      ENDIF                                                                                     XEXEC
727   7                   ENDDO                                                                                        XEXEC
728   7                   IF EXECUTION MODE WAS SEMI OR AUTO                                                           XEXEC
729   7                   THEN                                                                                         XEXEC
730   8                      CALL XESCN TO PURGE ANY RESIDUAL SCAN CONTOOL DATA AND FILES                              XEXEC
731   7                   ENDIF                                                                                        XEXEC
732   6                ELSE                                                                                            XEXEC
733   7                   CALL APPROPRIATE DIRECTIVE HANDLER VIA XDCL?                                                 XEXEC
734   6                ENDIF                                                                                           XEXEC
735   5             ENDIF                                                                                              XEXEC
736   4          ENDIF                                                                                                 XEXEC

737   4          :RESET:                                                                                               XEXEC
738   4          IF SUBSTA IS NOT DIRECTIVE LEVEL                                                                      XEXEC
739   4          THEN                                                                                                  XEXEC
740   5             CALL XEIND TO REINITIALIZE DYNAMIC COMMON                                                          XEXEC
741   4          ENDIF                                                                                                 XEXEC
742   3       ENDIF                                                                                                    XEXEC
```

```
XEXE
XEXE

743     2 ENDDO
744     1 END XEXEC
```

```
 SYMBOL DEFINITION TABLE
 DBLOG     149
:DIRECT:    32
:ERR01 :   549
:ERR01 :   387
:ERR01 :   158
:ERR02 :   388
:ERR02 :   550
:ERR03 :   551
:ERR04 :   389
:ERR04 :   552
:ERR05 :   558
:ERR10 :   293
:ERR11 :   294
:ERR12 :   295
:ERR12 :   394
:ERR13 :   559
:ERR13 :   296
:ERR14 :   297
:ERR15 :   298
:ERR16 :   299
:ERR17 :   300
:ERR2  :   289
:ERR4  :   290
:ERR5  :   465
:ERR6  :   291
:ERR8  :   292
:EXECUT:    33
:GLOBAL:    31
:INTEDT:    35
:RESET :   737
:RETURN:   466
:SEQEDT:    34
 XECAL      29
 XEIND      65
 XEINE     108
 XEINI     218
 XEINS     350
 XEINT     439
 XEINX     494
:XELDS :   600
 XELDS     599
:XERTN :   604
 XESCN     638
 XEXEC     685

@XQT F.PDLIST
```

```
    1 CD0   FORTRAN CALLING PROCEEDURE                                                  XINTE
    1 CD0                                                                               XINTE
    1 CD0  CALL XINTE                                                                   XINTE
    1 CD0                                                                               XINTE
    1 C*********                                                                        XINTE
    1 CD1                                                                               XINTE
    1 CD1  OVERLAY INTERFACE ROUTINE FOR INTERFACE TABLE EDITOR                         XINTE
    1 CD1                                                                               XINTE
    1 C*********                                                                        XINTE
    1 CD2                                                                               XINTE
    1 CD2       INPUT                                                                   XINTE
    1 CD2                                                                               XINTE
    1 CD2   COMMON XE -                                                                 XINTE
    1 CD2                LU     = USER'S LOGICAL UNIT NO.                               XINTE
    1 CD2                                                                               XINTE
    1 CD2   COMMON XB -                                                                 XINTE
    1 CD2                DEBUG  = DEBUG AND TRACE FLAG FOR INTERFACE                    XINTE
    1 CD2                         TABLE EDITOR ROUTINES                                 XINTE
    1 CD2                                                                               XINTE
    1 C*********                                                                        XINTE
    1 CD5       NOTES                                                                   XINTE
    1 CD5                                                                               XINTE
    1 CD5   USES ROUTINES                                                               XINTE
    1 CD5                 XINIX                                                         XINTE
    1 CD5                 XUDBG                                                         XINTE
    1 CD5                 XERTN                                                         XINTE
    1 CD5                                                                               XINTE
    1 C*********                                                                        XINTE
    1 *                                                                                 XINTE
    1 *                                                                                 XINTE
    1 * XINTE IS THE INTERFACE ROUTINE FOR THE INTERFACE TABLE EDITOR                   XINTE
    1 * WHEN CALLED IN THE FDS EXECUTIVE'S OVERLAY STRUCTURE.                           XINTE
    1 *                                                                                 XINTE
    1 BEGIN XINTE                                                                       XINTE
    2    CALL XINIX TO EXECUTE INTERFACE TABLE EDITOR                                   XINTE
    2    CALL XERTN TO RETURN TO XEXEC IN MAIN SEGMENT                                  XINTE
    1 END XINTE                                                                         XINTE
                                                                                        XINTE
```

```
40    1 CD0   FORTRAN CALLING PROCEEDURE                                                    XINIX
41    1 CD0                                                                                XINIX
42    1 CD0   CALL XINIX                                                                   XINIX
43    1 CD0                                                                                XINIX
44    1 C*********                                                                         XINIX
45    1 CD1                                                                                XINIX
46    1 CD1  MAIN PROGRAM FOR INTERFACE TABLE EDITOR                                       XINIX
47    1 CD1                                                                                XINIX
48    1 C*********                                                                         XINIX
49    1 CD2                                                                                XINIX
50    1 CD2       INPUT                                                                    XINIX
51    1 CD2                                                                                XINIX
52    1 CD2   COMMON XE -                                                                  XINIX
53    1 CD2                COMBUF = TERMINAL COMMUNICATIONS OUTPUT BUFFER                  XINIX
54    1 CD2                COMPTR = INDEX TO NEXT TOKEN IN COMBUF                          XINIX
55    1 CD2                BACKSL = TOKEN FOR "/"                                          XINIX
56    1 CD2                TOKENS = IDENTIFYING VALUES OF TOKENS IN COMBUF                 XINIX
57    1 CD2                                                                                XINIX
58    1 CD2   COMMON XB -                                                                  XINIX
59    1 CD2                LITLEN = LENGTH OF LITERAL AREA OF INTERFACE                    XINIX
60    1 CD2                         TABLE IN WKBUF                                         XINIX
61    1 CD2                NARG   = INDEX TO START OF SHORT PROMPT AREA OF                 XINIX
62    1 CD2                         WKBUF                                                  XINIX
63    1 CD2                NEWTAB = NAME OF INTERFACE TABLE TO BE CREATED BY               XINIX
64    1 CD2                         THIS EDIT. INPUT = 0 IF XINTE IS CALLED                XINIX
65    1 CD2                         BY FDS LIBRARY MAINTENANCE PROGRAM.                    XINIX
66    1 CD2                NUMARG = NUMBER OF ARGUMENTS IN THIS INTERFACE                  XINIX
67    1 CD2                         TABLE                                                  XINIX
68    1 CD2                WKBUF  = WORKING BUFFER CONTAINING INTERFACE TABLE              XINIX
69    1 CD2                         AND SHORT PROMPTS WITH LITERAL IN EXPANDED             XINIX
70    1 CD2                         FORM.                                                  XINIX
71    1 CD2                                                                                XINIX
72    1 C*********                                                                         XINIX
73    1 CD3                                                                                XINIX
74    1 CD3       OUTPUT                                                                   XINIX
75    1 CD3                                                                                XINIX
76    1 CD3   COMMON XE -                                                                  XINIX
77    1 CD3                MASSTA = MASTER STATUS SET TO 'X' LEVEL IF ERROR OR             XINIX
78    1 CD3                         IF 'X' INPUT                                           XINIX
79    1 CD3                                                                                XINIX
80    1 CD3   COMMON XB -                                                                  XINIX
81    1 CD3                PRMTMD = CURRENT PROMPT MODE.  SET TO 5 (MODIFY MODE)           XINIX
82    1 CD3                         IF '/' INPUT                                           XINIX
83    1 CD3                WKBUF  = WORKING BUFFER CONTAINING INTERFACE TABLE              XINIX
84    1 CD3                         WITH LITERALS IN COMPACTED FORM                        XINIX
85    1 CD3                                                                                XINIX
86    1 CD3   AWA       - XXXXXX = NEW INTERFACE TABLE AND LITERAL AREA                    XINIX
87    1 CD3                         (NOT DONE IF CALLED BY LIBRARY MAINTENANCE)            XINIX
88    1 CD3                                                                                XINIX
89    1 C*********                                                                         XINIX
90    1 CD5                                                                                XINIX
91    1 CD5       NOTES                                                                    XINIX
92    1 CD5                                                                                XINIX
93    1 CD5   USES ROUTINES                                                                XINIX
94    1 CD5                EXEC                                                            XINIX
95    1 CD5                XREQ                                                            XINIX
96    1 CD5                XILIT                                                           XINIX
97    1 CD5                XIPRM                                                           XINIX
98    1 CD5                XINPT                                                           XINIX
```

```
 99    1 CDS               XRMOV                                                                                    XINIX
100    1 CDS               XRMSG                                                                                    XINIX
101    1 CDS               XTCOM                                                                                    XINIX
102    1 CDS                                                                                                        XINIX
103    1 C*********                                                                                                 XINIX
104    1 *                                                                                                          XINIX
105    1 * THE INTERFACE TABLE EDITOR IS ENTERED AS A RESULT OF THE 'INTF' DIRECTIVE                                XINIX
106    1 * OR FROM THE EXECUTION CONTROLLER TO COMPLETE AN INTERFACE TABLE. THE DIRECTIVE                           XINIX
107    1 * PROVIDES THE NAME OF THE TABLE TO BE EDITED AND THE NAME FOR THE NEW                                     XINIX
108    1 * TABLE.  THE EDITOR INTERACTS WITH THE USER IN ORDER TO ACQUIRE DATA VALUES                               XINIX
109    1 * OR VARIABLE NAMES FOR EACH OF THE PARAMETERS IN THE INTERFACE TABLE.                                     XINIX
110    1 * NOTE : ALL INITIALIZATION, INCLUDING WKBUF (OLD INTERFACE TABLE),                                        XINIX
111    1 *        HAS BEEN PERFORMED BY XEINI.                                                                      XINIX
112    1 *                                                                                                          XINIX
113    1 BEGIN XINIX                                                                                                XINIX
114    2    IF #P (NO. OF PARAMETERS) > 0, THEN                                                                     XINIX
115    3       SET ARGNO (NO. OF CURRENT ARGUMENT BEING PROCESSED) TO 0                                             XINIX
116    3 * PRMTMD = 1 => CREATE N MODE                                                                              XINIX
117    3 * PRMTMD = 3 => CREATE A MODE                                                                              XINIX
118    3 * PRMTMD = 4 => CREATE CONTINE MODE                                                                        XINIX
119    3 * PRMTMD = 5 => MODIFY MODE                                                                                XINIX
120    3       DO UNTIL 'EXIT' OR '%' IS ENTERED                                                                    XINIX
121    4          CALL XIPRM TO CONSTRUCT A PROMPT BASED ON PRMTMD, SIZE, TYPE, AND STATUS                          XINIX
122    4          OF NEXT ARGUMENT                                                                                  XINIX
123    4          CALL XTCOM TO PROMPT USER AND RETURN PARSED INPUT                                                 XINIX
124    4          IF '%' WAS NOT ENTERED, THEN                                                                      XINIX
125    5             IF '\' WAS ENTERED, THEN                                                                       XINIX
126    6                SET PRMTMD TO 5                                                                             XINIX
127    5             ELSE                                                                                           XINIX
128    6                IF NOTHING WAS ENTERED (I.E. TOKEN IS EOS), THEN                                            XINIX
129    7                   INCREMENT TO NEXT ARGUMENT                                                               XINIX
130    6                ELSE                                                                                        XINIX
131    7                   CALL XINPT TO PROCESS THE USER'S INPUT                                                   XINIX
132    6                ENDIF                                                                                       XINIX
133    5             ENDIF                                                                                          XINIX
134    4          ENDIF                                                                                             XINIX
135    3       ENDDO                                                                                                XINIX
136    3       IF A '%' WAS ENTERED, THEN                                                                           XINIX
137    4          SET RETURN CODE INDICATING %                                                                      XINIX
138    4          (I.E.  MASSTA = 0)                                                                                XINIX
139    3       ELSE                                                                                                 XINIX
140    4          COMPRESS THE LITERAL LIST AREA                                                                    XINIX
141    3       ENDIF                                                                                                XINIX
142    2    ENDIF                                                                                                   XINIX
143    2    STORE INTERFACE TABLE AS NEWNAME                                                                        XINIX
144    2    IF STORE INTO AWA FAILED, THEN                                                                          XINIX
145    3       SET MASSTA TO INDICATE DIRECTIVE LEVEL (=0)                                                          XINIX
146    2    ELSE                                                                                                    XINIX
147    3       SET GOOD RETURN CODE                                                                                 XINIX
148    2    ENDIF                                                                                                   XINIX
149    1 END XINIX                                                                                                  XINIX
```

```
152   1 CD0       FORTRAN CALLING PROCEEDURE                                                    XIPRM
153   1 CD0                                                                                     XIPRM
154   1 CD0   CALL XIPRM                                                                        XIPRM
155   1 CD0                                                                                     XIPRM
156   1 C**********                                                                             XIPRM
157   1 CD1                                                                                     XIPRM
158   1 CD1       PROMPT CONSTUCTOR FOR INTERFACE TABLE EDIYOR                                  XIPRM
159   1 CD1                                                                                     XIPRM
160   1 C**********                                                                             XIPRM
161   1 CD2                                                                                     XIPRM
162   1 CD2       INPUT                                                                         XIPRM
163   1 CD2                                                                                     XIPRM
164   1 CD2   COMMON XB -                                                                       XIPRM
165   1 CD2               ARGNO,  CFLAG,  DFLAG,  IARG,   IOFLAG,                               XIPRM
166   1 CD2               ISIZE,  ITYPE,  LENEFF, LITSIZ, NARG,                                 XIPRM
167   1 CD2               NDXBTM, NUMARG, PRMTMD, WKBUF                                         XIPRM
168   1 CD2                                                                                     XIPRM
169   1 C**********                                                                             XIPRM
170   1 CD3                                                                                     XIPRM
171   1 CD3       OUTPUT                                                                        XIPRM
172   1 CD3                                                                                     XIPRM
173   1 CD3   COMMON XB -                                                                       XIPRM
174   1 CD3               ARGNO,  ISUB,   MODSAV, PRMLEN,                                       XIPRM
175   1 CD3               PRMTMD, PROMPT                                                        XIPRM
176   1 CD3                                                                                     XIPRM
177   1 C**********                                                                             XIPRM
178   1 CD5                                                                                     XIPRM
179   1 CD5       NOTES                                                                         XIPRM
180   1 CD5                                                                                     XIPRM
181   1 CD5   USES ROUTINES                                                                     XIPRM
182   1 CD5                CNUMD                                                              XIPRM
183   1 CD5                XIEXT                                                              XIPRM
184   1 CD5                XILSB                                                              XIPRM
185   1 CD5                XRMOV                                                              XIPRM
186   1 CD5                XRNXB                                                              XIPRM
187   1 CD5                XRPCK                                                              XIPRM
188   1 CD5                XRUPK                                                              XIPRM
189   1 CD5                                                                                   XIPRM
190   1 C**********                                                                           XIPRM
```

```
192   1 * CONSTRUCT PROMPT TO BE ISSUED                                                                                    XIPRM
193   1 BEGIN XIPRM                                                                                                        XIPRM
194   2    DO UNTIL A PROMPT IS CONSTUCTED                                                                                 XIPRM
195   3       IF PRMTMD = 5, THEN                                                                                          XIPRM
196   4          CONSTRUCT A '\:' PROMPT                                                                                   XIPRM
197   3       ELSE                                                                                                         XIPRM
198   4          IF PRMTMD = 4 (CONTINUE MODE), OR                                                                         XIPRM
199   5             PRMTMD = 6 (CONTINUE HERE MODE), THEN                                                                  XIPRM
200   5             IF ARGNO IS A SCALAR, THEN                                                                             XIPRM
201   6                CONSTRUCT PROMPT AS '\ARG=:' OR '\ARG@:'                                                            XIPRM
202   5             ELSE                                                                                                   XIPRM
203   6                IF PRMTMD NOT = 6, THEN                                                                             XIPRM
204   7                   COMPUTE ISUB AS NEXT EMPTY ELEMENT BEYOND LAST ENTERED (LASTE)                                   XIPRM
205   6                ELSE                                                                                                XIPRM
206   7                   SET PRMTMD TO 4                                                                                  XIPRM
207   7                   COMPUTE CURRENT ELEMENT NO. (SUBSCRIPT) FROM CURRENT INDEX                                       XIPRM
208   6                ENDIF                                                                                               XIPRM
209   6                IF THERE ARE NO EMPTY SLOTS BEYOND LASTE, THEN                                                      XIPRM
210   7                   SET PRMTMD TO MODSAV (EXIT THE CONTINUE MODE)                                                    XIPRM
211   6                ELSE                                                                                                XIPRM
212   7                   IF ARGNO IS DOUBLY SUBSCRIPTED PARAMETER, THEN                                                   XIPRM
213   8                      COMPUTE I & J FROM ISUB AND I-DIMENSION (IDIM)                                                XIPRM
214   8                      CONSTRUCT PROMPT AS '\ARG=(I,J):'                                                             XIPRM
215   7                   ELSE                                                                                             XIPRM
216   8                      CONSTUCT PROMPT AS '\ARG=(ISUB):'                                                             XIPRM
217   7                   ENDIF                                                                                            XIPRM
218   6                ENDIF                                                                                               XIPRM
219   5             ENDIF                                                                                                  XIPRM
220   4          ELSE                                                                                                      XIPRM
221   5             IF ARGNO = #P, THEN                                                                                    XIPRM
222   6                SET PRMTMD TO 5                                                                                     XIPRM
223   5             ELSE                                                                                                   XIPRM
224   6                INCREMENT ARGNO TO NEXT PARAMETER                                                                   XIPRM
225   6                SET IOFLAG TO 1, 2, OR 3 INDICATING I, O, OR IO                                                     XIPRM
226   6                SET LAST ENTERED INDICATOR (LASTE) TO 0                                                             XIPRM
227   6                IF PRMTMD = 3, THEN                                                                                 XIPRM
228   7                   IF SOME DATA VALUE(S) OR PARM NAME EXISTS FOR ARGNO, THEN                                        XIPRM
229   8                      CALL XILSD TO LIST DATA FOR THIS ARGUMENT                                                     XIPRM
230   7                   ENDIF                                                                                            XIPRM
231   7                   CONSTRUCT PROMPT AS '\ARG=:' OR @ OR =@                                                          XIPRM
232   6                ELSE                                                                                                XIPRM
233   7                   IF ARGNO IS MARKED INCOMPLETE, THEN                                                              XIPRM
234   8                      IF A PARTIAL LITERAL LIST EXISTS, OR                                                          XIPRM
235   9                         THIS ARGUMENT IS A SCALAR, THEN                                                            XIPRM
236   9                         COMPUTE ISUB AS FIRST EMPTY ELEMENT                                                        XIPRM
237   9                         IF DOUBLY SUBSCRIPTED PARAMETER, THEN                                                      XIPRM
238  10                            COMPUTE I & J FROM ISUB AND I-DIMENSION (IDIM)                                          XIPRM
239  10                            CONSTRUCT PROMPT '\ARG=(I,J):'                                                          XIPRM
240   9                         ELSE                                                                                       XIPRM
241  10                            CONSTRUCT PROMPT '\ARG=(ISUB):'                                                         XIPRM
242   9                         ENDIF                                                                                      XIPRM
243   8                      ELSE                                                                                          XIPRM
244   9                         CONSTRUCT PROMPT AS '/ARG=:' OR @ OR =@                                                    XIPRM
245   8                      ENDIF                                                                                         XIPRM
246   7                   ENDIF                                                                                            XIPRM
247   6                ENDIF                                                                                               XIPRM
248   5             ENDIF                                                                                                  XIPRM
249   4          ENDIF                                                                                                     XIPRM
250   3       ENDIF                                                                                                        XIPRM
```

```
251   2   ENDDO
252   1 END XIPMM
```

```
254   1 CDC       FORTRAN CALLING PROCEDURE                                    XIPMT
255   1 CDC                                                                    XIPMT
256   1 CDC   CALL XIPMT                                                       XIPMT
257   1 CDD                                                                    XIPMT
258   1 C***********                                                           XIPMT
259   1 CD1                                                                    XIPMT
260   1 CD1   PROMPT DIRECTIVE PROCESSOR                                       XIPMT
261   1 CD1                                                                    XIPMT
262   1 C***********                                                           XIPMT
263   1 CD2                                                                    XIPMT
264   1 CD2       INPUT                                                        XIPMT
265   1 CD2                                                                    XIPMT
266   1 CD2   COMMON XE - COMBUF, COMPTR, TOKENS                               XIPMT
267   1 CD2                                                                    XIPMT
268   1 C***********                                                           XIPMT
269   1 CD3                                                                    XIPMT
270   1 CD3       OUTPUT                                                       XIPMT
271   1 CD3                                                                    XIPMT
272   1 CD3   COMMON XB - ARGNO, PRMTMD                                        XIPMT
273   1 CD3                                                                    XIPMT
274   1 C***********                                                           XIPMT
275   1 * XIPMT PROCEDURE IS THE PROMPT DIRECTIVE                              XIPMT
276   1 BEGIN XIPMT                                                            XIPMT
277   2    ERREXIT IF TOKEN IS NOT COMMA  :ERR02:                              XIPMT
278   2    POSITION TO NEXT TOKEN                                              XIPMT
279   2    ERREXIT IF TOKEN IS NOT NAME  :ERR02:                               XIPMT
280   2    ERREXIT IF TOKEN(S) FOLLOW THE NAME  :ERR02:                        XIPMT
281   2    IF NAME IS 'N', THEN                                                XIPMT
282   3       SET PRMTMD TO 1                                                  XIPMT
283   2    ELSE                                                                XIPMT
284   3       IF NAME IS 'A', THEN                                             XIPMT
285   4          SET PRMTMD TO 3                                               XIPMT
286   3       ELSE                                                             XIPMT
287   4          ERREXIT  :ERR02:                                              XIPMT
288   3       ENDIF                                                            XIPMT
289   2    ENDIF                                                               XIPMT
290   2    SET ARGNO TO 0                                                      XIPMT
291   1 EXIT TO :RETURN:                                                       XIPMT

292   2    :ERR02: CALL XRMSG FOR 'INVALID SYNTAX'                             XIPMT

293   2    :RETURN:                                                            XIPMT
294   1 END XIPMT                                                              XIPMT
```

```
296    1 CD0       FORTRAN CALLING PROCEEDURE                                                    XILST
297    1 CD0                                                                                     XILST
298    1 CD0    CALL XILS'                                                                       XILST
299    1 CD0                                                                                     XILST
300    1 C*********                                                                              XILST
301    1 CB1                                                                                     XILST
302    1 CB1    LIST DIRECTIVE PROCESSOR (WITHIN INTERFACE TABLE EDITOR)                          XILST
303    1 CB1                                                                                     XILST
304    1 C*********                                                                              XILST
305    1 CD2                                                                                     XILST
306    1 CD2       INPUT                                                                         XILST
307    1 CD2                                                                                     XILST
308    1 CD2    COMMON XE - COMBUF, COMPTR, TOKENS                                               XILST
309    1 CD2                                                                                     XILST
310    1 CD2    COMMON XB - CFLAG,  LITPTR, NARG                                                 XILST
311    1 CD2                NUMARG, VERSON, WKBUF                                              XILST
312    1 CD2                                                                                     XILST
313    1 C*********                                                                              XILST
314    1 CD3                                                                                     XILST
315    1 CD3       OUTPUT                                                                        XILST
316    1 CD3                                                                                     XILST
317    1 CD3    COMMON XB - ARGNO                                                                XILST
318    1 CD3                                                                                     XILST
319    1 C*********                                                                              XILST
320    1 CD5                                                                                     XILST
321    1 CD5       NOTES                                                                         XILST
322    1 CD5                                                                                     XILST
323    1 CD5    USES ROUTINES                                                                    XILST
324    1 CD5                                                                                     XILST
325    1 CD5      EXEC                                                                           XILST
326    1 CD5      XICHR                                                                          XILST
327    1 CD5      XIEXT                                                                          XILST
328    1 CD5      XILSB                                                                          XILST
329    1 CD5      XRCPR                                                                          XILST
330    1 CD5      XRI6                                                                           XILST
331    1 CD5      XRMOV                                                                          XILST
332    1 CD5      XRMSG                                                                          XILST
333    1 CD5      XRSET                                                                          XILST
334    1 CD5                                                                                     XILST
335    1 C*********                                                                              XILST
```

```
337   1 * XILST PROCESSES THE LIST DIRECTIVE                                                                XILST
338   1 BEGIN XILST                                                                                         XILST
339   2    IF TOKEN IS '-', THEN                                                                            XILST
340   3       POSITION TO NEXT TOKEN                                                                        XILST
341   3       ERREXIT IF TOKEN IS NOT NAME  : ERR02:                                                        XILST
342   3       ERREXIT IF NAME IS NOT 'C', 'V', OR 'A'   :ERR02:                                             XILST
343   3       SET MODEFG TO INDICATE SPECIFIED MODE (C=1, V=2, A=3)                                         XILST
344   3       POSITION TO NEXT TOKEN                                                                        XILST
345   2    ELSE                                                                                             XILST
346   3       SET MODEFG TO 2                                                                               XILST
347   2    ENDIF                                                                                            XILST
348   2    IF TOKEN IS EOS, THEN                                                                            XILST
349   3       WRITE A HEADER LINE INDICATING TABLE NAME, PROCESSOR VERSION                                  XILST
350   3       AND STATUS                                                                                    XILST
351   3       DO UNTIL ALL ARGUMENTS HAVE BEEN PROCESSED                                                    XILST
352   4          IF MODEFG = 1 OR MODEFG = 3, THEN                                                          XILST
353   5             CALL XICHR TO WRITE CHARACTERISTICS OF THIS ARGUMENT                                    XILST
354   4          ENDIF                                                                                      XILST
355   4          IF MODEFG = 2 OR MODEFG = 3, THEN                                                          XILST
356   5             CALL XILSD TO WRITE DATA VALUES OF THIS ARGUMENT                                        XILST
357   4          ENDIF                                                                                      XILST
358   3       ENDDO                                                                                         XILST
359   2    ELSE                                                                                             XILST
360   3       DO UNTIL EOS IS REACHED                                                                       XILST
361   4          ERREXIT IF TOKEN IS NOT COMMA  :ERR02:                                                     XILST
362   4          ERREXIT IF NEXT TOKEN IS NOT NAME  :ERR02:                                                 XILST
363   4          SET ARGNO TO 1                                                                             XILST
364   4          START SEARCH DO UNTIL ALL ARGUMENTS HAVE BEEN PROCESSED                                    XILST
365   4          EXIT IF NAME = ARGNO'S NAME IN PROMPT TABLE                                                XILST
366   5             IF MODEFG = 1 OR MODEFG = 3, THEN                                                       XILST
367   6                CALL XICHR TO WRITE CHARACTERISTICS OF THIS ARGUMENT                                 XILST
368   5             ENDIF                                                                                   XILST
369   5             IF MODEFG = 2 OR MODEFG = 3, THEN                                                       XILST
370   6                CALL XILSD TO WRITE DATA VALUES OF THIS ARGUMENT                                     XILST
371   5             ENDIF                                                                                   XILST
372   4          ENDLOOP                                                                                    XILST
373   5             PRINT MESSAGE THAT NAME IS NOT A VALID PARAMETER                                        XILST
374   4          ENDSEARCH                                                                                  XILST
375   4          INCREMENT TO NEXT TOKEN                                                                    XILST
376   3       ENDDO                                                                                         XILST
377   2    ENDIF                                                                                            XILST
378   1 EXIT TO :RETURN:                                                                                    XILST

379   2    :ERR02: CALL XRMSG TO WRITE 'INVALID SYNTAX'                                                     XILST

380   2    :RETURN:                                                                                         XILST
381   1 END XILST                                                                                           XILST
```

```
383   1 CD0                                                                                                    XISUB
384   1 CD0       FORTRAN CALLING PROCEEDURE                                                                   XISUB
385   1 CD0                                                                                                    XISUB
386   1 CD0    CALL XISUB                                                                                      XISUB
387   1 CD0                                                                                                    XISUB
388   1 C*********                                                                                             XISUB
389   1 CD1                                                                                                    XISUB
390   1 CD1    EFFECTIVE SUBSCRIPT CALCULATION ROUTINE                                                         XISUB
391   1 CD1                                                                                                    XISUB
392   1 C*********                                                                                             XISUB
393   1 CD2                                                                                                    XISUB
394   1 CD2       INPUT                                                                                        XISUB
395   1 CD2                                                                                                    XISUB
396   1 CD2    COMMON XE - COMBUF, COMPTR, TOKENS                                                              XISUB
397   1 CD2                                                                                                    XISUB
398   1 CD2    COMMON XB - IDIM, ISIZE, LENEFF                                                                 XISUB
399   1 CD2                                                                                                    XISUB
400   1 C*********                                                                                             XISUB
401   1 CD3                                                                                                    XISUB
402   1 CD3       OUTPUT                                                                                       XISUB
403   1 CD3                                                                                                    XISUB
404   1 CD3    COMMON XB - IRETC, ISUB                                                                         XISUB
405   1 CD3                                                                                                    XISUB
406   1 C*********                                                                                             XISUB
407   1 CD5                                                                                                    XISUB
408   1 CD5    USES ROUTINES                                                                                   XISUB
409   1 CD5                                                                                                    XISUB
410   1 CD5      XRMSG                                                                                         XISUB
411   1 CD5                                                                                                    XISUB
412   1 C*********                                                                                             XISUB
413   1 * XISUB IS CALLED TO CALCULATE AN EFFECTIVE SUBSCRIPT (ISUB) FROM                                      XISUB
414   1 * THE INPUT SUBSCRIPT                                                                                  XISUB
415   1 BEGIN XISUB                                                                                            XISUB
416   2    INCREMENT TO NEXT TOKEN                                                                             XISUB
417   2    ERREXIT IF TOKEN IS NOT INTEGER VALUE  :ERR14:                                                      XISUB
418   2    IF IDIM FOR THIS ARGUMENT > 0, THEN                                                                 XISUB
419   3       ERREXIT IF SPECIFIED INTEGER VALUE > IDIM  :ERR16:                                               XISUB
420   3       INCREMENT TO NEXT TOKEN                                                                          XISUB
421   3       ERREXIT IF TOKEN IS NOT COMMA  :ERR15:                                                           XISUB
422   3       INCREMENT TO NEXT TOKEN                                                                          XISUB
423   3       ERREXIT IF TOKEN IS NOT INTEGER VALUE  :ERR14:                                                   XISUB
424   3       CALCULATE ISUB AS (J-1)*IDIM+I                                                                   XISUB
425   2    ELSE                                                                                                XISUB
426   3       SET ISUB TO INTEGER VALUE                                                                        XISUB
427   2    ENDIF                                                                                               XISUB
428   2    ERREXIT IF ISUB > ISIZE  :ERR16:                                                                    XISUB
429   2    INCREMENT TO NEXT TOKEN                                                                             XISUB
430   2    ERREXIT IF TOKEN IS NOT RIGHT PARENTHESIS  :ERR14:                                                  XISUB
431   1 EXIT TO :RETURN:                                                                                       XISUB

432   2    :ERR14: CALL XRMSG - 'INVALID SUBSCRIPT SYNTAX'                                                     XISUB

433   2    :ERR15: CALL XRMSG - 'DOUBLY SUBSCRIPTED - MUST SPECIFY BOTH'                                       XISUB

434   2    :ERR16: CALL XRMSG - 'INVALID SUBSCRIPT VALUE'                                                      XISUB

435   2    :RETURN:                                                                                            XISUB
436   1 END XISUB                                                                                              XISUB
```

```
      1 CD0       FORTRAN CALLING PROCEEDURE                                    XIDAT
      1 CD0                                                                     XIDAT
      1 CD0   CALL XIDAT                                                        XIDAT
      1 CD0                                                                     XIDAT
      1 C*********                                                              XIDAT
      1 CD1                                                                     XIDAT
      1 CD1   LITERAL DATA PROCESSOR                                            XIDAT
      1 CD1                                                                     XIDAT
      1 C*********                                                              XIDAT
      1 CD2                                                                     XIDAT
      1 CD2       INPUT                                                         XIDAT
      1 CD2                                                                     XIDAT
      1 CD2   COMMON XE - COMBUF, COMPTR, TOKENS                                XIDAT
      1 CD2                                                                     XIDAT
      1 CD2   COMMON XB - CFLAG,  COMFLG, DFLAG,  IARG,   IARG4,                XIDAT
      1 CD2               ISIZE,  ISIZES, ITYPE,  LENEFF, LITDSP,               XIDAT
      1 CD2               LITDWN, LITSIZ, MODSAV, NDXBTM, NOBITM,               XIDAT
      1 CD2               PRMTHD, SFLAG,  WKBLNG, WKBUF                         XIDAT
      1 CD2                                                                     XIDAT
      1 C*********                                                              XIDAT
      1 CD3                                                                     XIDAT
      1 CD3       OUTPUT                                                        XIDAT
      1 CD3                                                                     XIDAT
      1 CD3   COMMON XB - IRETC,  ISUB,   LITDSP, LITDWN, NDXBTM,               XIDAT
      1 CD3               PRMTHD, WKBUF                                         XIDAT
      1 CD3                                                                     XIDAT
      1 C*********                                                              XIDAT
      1 CD4                                                                     XIDAT
      1 CD4       INTERNAL VARIABLES                                            XIDAT
      1 CD4                                                                     XIDAT
      1 CD4   COMMON XB - BITNO,  BITNUM, LENMOV,                               XIDAT
      1 CD4                                                                     XIDAT
      1 CD4   COMMON XS -                                                       XIDAT
      1 CD4               IDISP  = INDEX INTO WKBUF OF LOCATION FOR THIS        XIDAT
      1 CD4                        LITERAL DATUM                                XIDAT
      1 CD4               NUMCPL = NUMBER OF CONSECUTIVE ELEMENTS TO BE         XIDAT
      1 CD4                        MARKED COMPLETED AS A RESULT OF THIS         XIDAT
      1 CD4                        LITERAL DATUM                                XIDAT
      1 CD4               STACK  = PUSH-DOWN LIST (MAX. OF 4 ENTRIES)           XIDAT
      1 CD4                        DESCRIBING NESTED REPEAT GROUPS.             XIDAT
      1 CD4                        EACH ENTRY IS 3 WORDS :                      XIDAT
      1 CD4                              WORD 1 = INDEX TO 1ST TOKEN (IN COMBUF)XIDAT
      1 CD4                                       AFTER REPEAT SYMBOL           XIDAT
      1 CD4                              WORD 2 = REPEAT COUNT                  XIDAT
      1 CD4                              WORD 3 = FLAG INDICATING WHETHER       XIDAT
      1 CD4                                       REPEAT GROUP IS PARENTHETICALLYXIDAT
      1 CD4                                       GROUPED                       XIDAT
      1 CD4               STKPTR = INDEX TO NEXT ENTRY TO BE BUILT IN 'STACK'   XIDAT
      1 CD4                                                                     XIDAT
      1 C*********                                                              XIDAT
      1 CD5                                                                     XIDAT
      1 CD5       NOTES                                                         XIDAT
      1 CD5                                                                     XIDAT
      1 CD5   USES ROUTINES                                                     XIDAT
      1 CD5                                                                     XIDAT
      1 CD5     XISUB                                                           XIDAT
      1 CD5     XRBIT                                                           XIDAT
      1 CD5     XRMOV                                                           XIDAT
      1 CD5     XRMSG                                                           XIDAT
```

```
497   1 CD5     XRNXB                                                                                    XIDAT
498   1 CD5     XRSET                                                                                    XIDAT
499   1 CD5                                                                                              XIDAT
500   1 C**********                                                                                      XIDAT
501   1 * XIDAT PROCESSES THE INPUT LITERAL LIST                                                         XIDAT
502   1 BEGIN XIDAT                                                                                      XIDAT
503   2    DO UNTIL EOS IS REACHED, OR                                                                   XIDAT
504   3       UNTIL AN ERROR OCCURS                                                                      XIDAT
505   3       IF THIS IS A DATA ELEMENT, THEN                                                            XIDAT
506   4          (I.E. INTEGER, REAL, DOUBLE OR CHAR.)                                                   XIDAT
507   4          IF ARG. TYPE (ITYPE) IS INTEGER, REAL, OR DOUBLE, THEN                                  XIDAT
508   5             ERREXIT IF DATA TYPE IS NOT SAME AS ITYPE :ERR10:                                    XIDAT
509   5             SET LENGTH TO BE MOVED (LENMOV) TO EFFECTIVE LENGTH FOR DATA                         XIDAT
510   5             OF ARG'S TYPE (LENEFF) (WILL BE 1, 2, OR 3 WORDS)                                    XIDAT
511   4          ELSE                                                                                    XIDAT
512   5             IF ARG. TYPE IS FREE, THEN                                                           XIDAT
513   6                SET LENGTH TO BE MOVED (LENMOV) TO EFFECTIVE LENGTH                               XIDAT
514   6                FOR DATA INPUT                                                                    XIDAT
515   5             ELSE  THIS MUST BE CHARACTER DATA BEING INPUT                                        XIDAT
516   6                ERREXIT IF ARGUMENT'S TYPE IS NOT CHARACTER DATA :ER10:                           XIDAT
517   6                INCREMENT TOKEN POINTER TO COUNT OF CHARACTERS                                    XIDAT
518   6                COMPUTE NO. WORDS IN INPUT CHARACTER STRING                                       XIDAT
519   6                ERREXIT IF NO. WORDS (LENMOV) > EFFECTIVE LENGTH OF                               XIDAT
520   6                THIS ARGUMENT'S DATA (LENEFF) :ERR10:                                             XIDAT
521   5             ENDIF                                                                                XIDAT
522   4          ENDIF                                                                                   XIDAT
523   4          INCREMENT TOKEN POINTER TO THE DATA INPUT                                               XIDAT
524   4          VERIFY THAT SUFFICIENT SPACE EXISTS IN LITERAL AREA OF THIS                             XIDAT
525   4          ARGUMENT FOR DATA INPUT (ISUB <= ISIZE - LENMOV + 1)                                    XIDAT
526   4          ERREXIT IF INSUFFICIENT SPACE :ERR11:                                                   XIDAT
527   4          IF DATA DOES NOT EXIST FOR THIS ARGUMENT, THEN                                          XIDAT
528   5             ALLOCATE AND INITIALIZE A LITERAL AREA FOR THIS ARGUMENT                             XIDAT
529   4          ENDIF                                                                                   XIDAT
530   4          MOVE DATA FROM INPUT COMMUNICATIONS BUFFER TO LITERAL AREA                              XIDAT
531   4          SET NUMBER OF ELEMENTS COMPLETED (NUMCMP) TO 1 OR, FOR A FREE                           XIDAT
532   4          ARGUMENT, TO LENMOV                                                                     XIDAT
533   4          IF LENMOV < LENEFF (ONLY POSSIBLE FOR CHARACTER DATA), THEN                             XIDAT
534   5             MOVE LENMOV-LENEFF BLANKS INTO LITERAL AREA AS A FILL                                XIDAT
535   4          ENDIF                                                                                   XIDAT
536   4          INCREMENT TO NEXT TOKEN                                                                 XIDAT
537   4          INCREMENT EFFECTIVE SUBSCRIPT (ISUB) BY LENMOV                                          XIDAT
538   4          SET LAST ENTERED INDICATOR (LASTE) TO ISUB - 1                                          XIDAT
539   3       ELSE , TOKEN INDICATES NON-DATA ELEMENT                                                    XIDAT
540   4          IF THIS IS A "(", THEN                                                                  XIDAT
541   5             CALL XISUB TO CALCULATE EFFECTIVE SUBSCRIPT (ISUB)                                   XIDAT
542   5             BASED ON INPUT SUBSCRIPT, ARGUMENT TYPE (ITYPE) AND                                  XIDAT
543   5             SECONDARY DIMENSION (IDIM)                                                           XIDAT
544   4          EXIT XIDAT IF ERROR (IRETC < 0)                                                         XIDAT
545   4          ELSE                                                                                    XIDAT
546   5             IF THIS IS AN "&", THEN                                                              XIDAT
547   6                IF DATA DOES NOT EXIST FOR THIS ARGUMENT, THEN                                    XIDAT
548   7                   CLEAR THE PARAMETER FIELD IN ARGUMENT'S CHARACTERISTICS                        XIDAT
549   6                ELSE                                                                              XIDAT
550   7                   MARK ONE ELEMENT OF THIS ARGUMENT AT ISUB AS INCOMPLETE                        XIDAT
551   7                   INCREMENT EFFECTIVE SUBSCRIPT (ISUB) BY EFFECTIVE                              XIDAT
552   7                   LENGTH OF ONE ELEMENT (LENEFF)                                                 XIDAT
553   6                ENDIF                                                                             XIDAT
554   6                TURN OFF COMPLETE FLAGS FOR THIS ARGUMENT AND INTERFACE TABLE                     XIDAT
555   6                INCREMENT TO NEXT TOKEN                                                           XIDAT
```

```
556   5            ELSE                                                                                                  XIDAT
557   6               IF TOKEN IS A REPEAT COUNT, THEN                                                                   XIDAT
558   7                  IF NEXT TOKEN IS '(', THEN                                                                      XIDAT
559   8                     SET PARENFLAG TO 1                                                                           XIDAT
560   8                     INCREMENT TO NEXT TOKEN                                                                      XIDAT
561   7                  ELSE                                                                                            XIDAT
562   8                     SET PARENFLAG TO 0                                                                           XIDAT
563   7                  ENDIF                                                                                           XIDAT
564   7                  SAVE REPEAT COUNT, TOKEN INDEX, AND PARENFLAG IN A PUSH-DOWN STACK                              XIDAT
565   6               ELSE                                                                                               XIDAT
566   7                  ERREXIT (INVALID FIELD) :ERR07:                                                                 XIDAT
567   6               ENDIF                                                                                              XIDAT
568   5            ENDIF                                                                                                 XIDAT
569   4         ENDIF                                                                                                    XIDAT
570   4         INCREMENT TO NEXT TOKEN                                                                                  XIDAT
571   3      ENDIF                                                                                                       XIDAT
572   3      DO UNTIL TOKEN IS NOT  ")"                                                                                  XIDAT
573   4         IF PUSH-DOWN STACK IS NOT EMPTY, AND                                                                     XIDAT
574   5         (PREVIOUS TOKEN WAS DATA, AND                                                                            XIDAT
575   5         PARENFLAG OF TOP OF STACK ENTRY IS 0), OR                                                                XIDAT
576   5         (CURRENT TOKEN IS ')', AND                                                                               XIDAT
577   5         PARENFLAG OF TOP OF STACK ENTRY IS 1), THEN                                                              XIDAT
578   5         GET REPEAT COUNT OF TOP OF STACK ENTRY                                                                   XIDAT
579   5         DECREMENT REPEAT COUNT BY 1                                                                              XIDAT
580   5         IF REPEAT COUNT > 0, THEN                                                                                XIDAT
581   6            SET TOKEN INDEX TO INDEX ON PUSH DOWN STACK                                                           XIDAT
582   6            REPLACE NEW REPEAT COUNT ON PUSH-DOWN STACK                                                           XIDAT
583   5         ELSE                                                                                                     XIDAT
584   6            POP (I.E. REMOVE) ENTRY FROM TOP OF STACK                                                             XIDAT
585   6            IF TOKEN IS ")", THEN                                                                                 XIDAT
586   7               INCREMENT TO NEXT TOKEN                                                                            XIDAT
587   6            ELSE                                                                                                  XIDAT
588   6            EXIT LOOP                                                                                             XIDAT
589   6            ENDIF                                                                                                 XIDAT
590   5         ENDIF                                                                                                    XIDAT
591   4      ELSE                                                                                                        XIDAT
592   5         IF PREVIOUS TOKEN WAS NOT A SUBSCRIPT, THEN                                                              XIDAT
593   6            IF THIS TOKEN IS NOT AN EOS, THEN                                                                     XIDAT
594   7               ERREXIT IF TOKEN IS NOT A COMMA  :ERR02:                                                           XIDAT
595   7               INCREMENT TO NEXT TOKEN                                                                            XIDAT
596   6            ENDIF                                                                                                 XIDAT
597   5         ENDIF                                                                                                    XIDAT
598   4      ENDIF                                                                                                       XIDAT
599   3   ENDDO                                                                                                          XIDAT
600   2   ENDDO                                                                                                          XIDAT
601   2   IF PROMPT MODE NOT CONTINUE (=4), AND                                                                          XIDAT
602   3      (THERE ARE EMPTY ELEMENTS BEYOND LASTE, OR                                                                  XIDAT
603   3      PREVIOUS TOKEN WAS A COMMA), THEN                                                                           XIDAT
604   3      RETAIN PROMPT MODE (SET MODSAV TO PRMTMD)                                                                   XIDAT
605   3      SET PROMPT MODE TO CONTINUE (=4)                                                                            XIDAT
606   2   ENDIF                                                                                                          XIDAT
607   2   IF ALL LITERAL SLOTS FILLED, THEN                                                                              XIDAT
608   3      MARK ARGNO COMPLETE                                                                                         XIDAT
609   3      IF ALL ARGUMENTS ARE COMPLETE, THEN                                                                         XIDAT
610   4         SET COMPLETE FLAG FOR INTERFACE TABLE                                                                    XIDAT
611   3      ENDIF                                                                                                       XIDAT
612   2   ELSE                                                                                                           XIDAT
613   3      TURN OFF COMPLETE FLAG FOR INTERFACE TABLE                                                                  XIDAT
614   2   ENDIF                                                                                                          XIDAT
```

```
615    1 EXIT TO :RETURN:                                                                    XIDAT
616    2    :ERR02: CALL XRMSG "INVALID SYNTAX"                                              XIDAT
617    2    :ERR07: CALL XRMSG  "ONLY DATA VALID TO RIGHT OF =                               XIDAT
618    2    :ERR10:  CALL XRMSG  "DATA TYPE INCOMPATIBLE WITH TYPE OF ARGUMENT"              XIDAT
619    2    :RETURN:                                                                         XIDAT
620    1 END XIDAT                                                                           XIDAT
```

```
1 CD0
1 CD0       FORTRAN CALLING PROCEEDURE                                XINPT
1 CD0                                                                 XINPT
1 CD0   CALL XINPT                                                    XINPT
1 CD0                                                                 XINPT
1 C*********                                                          XINPT
1 CD1                                                                 XINPT
1 CD1   INTERFACE TABLE EDITOR'S  INPUT PROCESSOR                     XINPT
1 CD1                                                                 XINPT
1 C*********                                                          XINPT
1 CD2                                                                 XINPT
1 CD2       INPUT                                                     XINPT
1 CD2                                                                 XINPT
1 CD2   COMMON XE - COMBUF, COMPTR, TOKENS                            XINPT
1 CD2                                                                 XINPT
1 CD2   COMMON XB - DIRECT, IOFLAG, ISUB,   NARG,                     XINPT
1 CD2               NDIR,   NUMARG, PRMTMD, WKBUF                     XINPT
1 CD2                                                                 XINPT
1 C*********                                                          XINPT
1 CD3                                                                 XINPT
1 CD3       OUTPUT                                                    XINPT
1 CD3                                                                 XINPT
1 CD3   COMMON XB - ARGNO, IRETC                                      XINPT
1 CD5                                                                 XINPT
1 C*********                                                          XINPT
1 CD5                                                                 XINPT
1 CD5       NOTES                                                     XINPT
1 CD5                                                                 XINPT
1 CD5   USES ROUTINES                                                 XINPT
1 CD5                                                                 XINPT
1 CD5      XIDAT                                                      XINPT
1 CD5      XIEXT                                                      XINPT
1 CD5      XILST                                                      XINPT
1 CD5      XIPAR                                                      XINPT
1 CD5      XIPNT                                                      XINPT
1 CD5      XRCPR                                                      XINPT
1 CD5      XRMOV                                                      XINPT
1 CD5      XRMSG                                                      XINPT
1 CD5                                                                 XINPT
1 C*********                                                          XINPT
```

```
663   1 * XINPT PROCESSES THE USER'S INPUT TEXT                                                    XINPT
664   1 BEGIN XINPT                                                                               XINPT
665   2    IF PRMTMD = 5, THEN                                                                    XINPT
666   3       SET IGFLAG OFF (=0)                                                                 XINPT
667   3       ERREXIT IF TOKEN IS NOT A NAME  :ERR02:                                             XINPT
668   3       SAVE NAME AND POSITION TO NEXT TOKEN                                                XINPT
669   3       IF TOKEN IS '=', THEN                                                               XINPT
670   4          POSITION TO NEXT TOKEN                                                           XINPT
671   4          IF TOKEN IS '@', THEN                                                            XINPT
672   5             SET IOFLAG TO IO                                                              XINPT
673   5             POSITION TO NEXT TOKEN                                                        XINPT
674   4          ELSE                                                                             XINPT
675   5             SET IOFLAG TO I                                                               XINPT
676   4          ENDIF                                                                            XINPT
677   3       ELSE                                                                                XINPT
678   4          IF TOKEN IS '@', THEN                                                            XINPT
679   5             SET IOFLAG TO O                                                               XINPT
680   4          ENDIF                                                                            XINPT
681   3       ENDIF                                                                               XINPT
682   3       IF IOFLAG NOT SET, THEN                                                             XINPT
683   4          CASE NAME (:EXIT:, :PROMPT:, :LIST:)                                             XINPT
684   5             ERREXIT IF ANOTHER TOKEN FOLLOWS  :ERR02:                                     XINPT

685   5             :EXIT: SET IRETC SO THAT PROMPTING LOOP TERMINATES                            XINPT

686   5             :PROMPT: CALL XIPMT TO PROCESS PROMPT DIRECTIVE                               XINPT

687   5             :LIST:   CALL XILST TO PROCESS LIST DIRECTIVE                                 XINPT
688   4          ENDCASE                                                                          XINPT
689   3       ENDIF                                                                               XINPT
690   3       START SEARCH UNTIL @F ENTRIES                                                       XINPT
691   3       EXIT IF NAME FOUND IN PROMPT TABLE                                                  XINPT
692   4          SET ARGNO TO ENTRY NO.                                                           XINPT
693   4          SET ISUB TO 1                                                                    XINPT
694   3       ORELSE                                                                              XINPT
695   4          INCREMENT TO NEXT PROMPT TABLE ENTRY                                             XINPT
696   3       ENDLOOP                                                                             XINPT
697   4          ERREXIT :ERR10:                                                                  XINPT
698   3       ENDSEARCH                                                                           XINPT
699   3       ERREXIT IF IOFLAG IS NOT SAME AS I/O TYPE OF ARGUMENT  :ERR05:                      XINPT
700   2    ENDIF                                                                                  XINPT
701   2    IF NEXT TOKEN IS A NAME, THEN                                                          XINPT
702   3       CALL XIPAR TO PROCESS A PARAMETER FIELD                                             XINPT
703   2    ELSE                                                                                   XINPT
704   3       ERREXIT IF IOFLAG IS NOT I ("=")  :ERR08:                                           XINPT
705   3       CALL XIDAT TO PROCESS DATA LIST                                                     XINPT
706   2    ENDIF                                                                                  XINPT
707   1 EXIT XINPT                                                                                XINPT
708   1 EXIT TO :RETURN:                                                                          XINPT

709   2    :ERR02: CALL XRMSG  "INVALID SYNTAX"                                                   XINPT

710   2    :ERR08: CALL XRMSG  "MUST USE PARAMETER NAME TO RIGHT OF @ OR =@ "                      XINPT

711   2    :RETURN:                                                                               XINPT
712   1 END XINPT                                                                                 XINPT
```

```
714  1 CD0
715  1 CD0        FORTRAN CALLING PROCEDURE
716  1 CD0
717  1 CD0    CALL XIPAR
718  1 CD0
719  1 C*********
720  1 CD1    PROCESS AN INPUT PARAMETER NAME AND ANY ASSOCIATED SUBSCRIPT
721  1 CD1    FIELD(S)
722  1 CD1
723  1 CD1
724  1 C*********
725  1 CD2        INPUT
726  1 CD2
727  1 CD2    COMMON XE - COMBUF, COMPTR, TOKENS
728  1 CD2
729  1 CD2
730  1 CD2    COMMON XB - DFLAG,  IAR6,   IAR64,  ICLASS,
731  1 CD2                LITDSP, SFLAG,  WKBLNG, WKBUF
732  1 CD2
733  1 C*********
734  1 CD3        OUTPUT
735  1 CD3
736  1 CD3
737  1 CD3    COMMON XB - IRETC,  LITDWN, WKBUF
738  1 CD3
739  1 C*********
740  1 CD5        NOTES
741  1 CD5
742  1 CD5
743  1 CD5    USES ROUTINES
744  1 CD5
745  1 CD5        XRMC'
746  1 CD5        XRMSG
747  1 CD5        XRSET
748  1 CD5
749  1 C*********
```

```
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
XIPAR
```

```
751   1 * XIPAR PROCESSES A USER SPECIFIED PARAMETER FIELD                                                    XIPAR
752   1 BEGIN XIPAR                                                                                          XIPAR
753   2    IF A NAME IS SPECIFIED, THEN                                                                      XIPAR
754   3       INCREMENT TO NEXT TOKEN                                                                        XIPAR
755   3       IF TOKEN IS '(', THEN                                                                          XIPAR
756   4          PROCESS I AND J SUBSCRIPTS                                                                  XIPAR
757   4          ERREXIT IF INVALID SUBSCRIPTING  :ERR14:                                                    XIPAR
758   4          IF DOUBLY SUBSCRIPTED, THEN                                                                 XIPAR
759   5             SET S-FLAG IN ARGNO'S SPECS. FIELD                                                       XIPAR
760   5             SET LJTDSP IN ARGNO'S SPECS. TO NEXT LITERAL AREA SPACE (LITDWN)                         XIPAR
761   5             PUT ISUB AND JSUB INTO LITERAL AREA AT THIS SPOT                                         XIPAR
762   4          ELSE                                                                                        XIPAR
763   5             SET LITDSP IN ARGNO'S SPECS TO ISUB                                                      XIPAR
764   4          ENDIF                                                                                       XIPAR
765   3       ELSE                                                                                           XIPAR
766   4          SET LITDSP IN ARGNO'S SPECS TO 0                                                            XIPAR
767   3       ENDIF                                                                                          XIPAR
768   3       ERREXIT IF DRDE HAS MORE THAN 4 CHARACTERS  :ERR18:                                            XIPAR
769   3       ERREXIT IF EXTRANEOUS FIELD INPUT  :ERR02:                                                     XIPAR
770   3       TURN OF D-FLAG (SOME LITERAL DATA) IN ARGNO'S SPECS.                                           XIPAR
771   3       SET PARAMETER NAME INTO ARGUMENT'S CHARACTERISTICS                                             XIPAR
772   3       SET COMPLETE (AND S) FLAG IN ARGUMENT'S CHARACTERISTICS                                        XIPAR
773   3       IF ALL ARGUMENTS ARE COMPLETE, THEN                                                            XIPAR
774   4          SET INTERFACE TABLE COMPLETE FLAG                                                           XIPAR
775   3       ENDI.                                                                                          XIPAR
776   2    ELSE  MUST BE A & INPUT                                                                           XIPAR
777   3       ERREXIT IF NOT AN AMPERSAND (&) INPUT  :ERR02:                                                 XIPAR
778   3       CLEAR PARAMETER NAME IN ARGUMENT'S CHARACTERISTICS                                             XIPAR
779   3       SET ARGUMENT AND INTERFACE TABLE INCOMPLETE                                                    XIPAR
780   2    ENDIF                                                                                             XIPAR
781   1 EXIT TO :RETURN:                                                                                     XIPAR

782   2    :ERR02:  CALL XRMSG - 'INVALID SYNTAX'                                                            XIPAR

783   2    :ERR14:  CALL XRMSG - 'INVALID SUBSCRIPT SYNTAX'                                                  XIPAR

784   2    :ERR18:  CALL XRMSG - 'INVALID DRDE NAME'                                                         XIPAR

785   2    :RETURN:                                                                                          XIPAR
786   1 END XIPAR                                                                                            XIPAR
```

```
1 C00
1 C00          FORTRAN CALLING PROCEDURE
1 C00
1 C00      CALL XILSD
1 C00
1 C**********
1 C01
1 C01  LIST DATA OR PARAMETER VALUES FOR ONE ARGUMENT
1 C01
1 C**********
1 C02
1 C02        INPUT
1 C02
1 C02      COMMON XE - [illegible]
1 C02
1 C02      COMMON XB - [illegible] DFLAG,  IARG,   IDIM,
1 C02                  IWISP,  IOFLAG, ISUB,   ITYPE,
1 C02                  LENEFF, LITDSP, LITSIZ, MARG,
1 C02                  NOBITM, SFLAG,  WKBUF
1 C02
1 C**********
1 C04
1 C04        INTERNAL VARIABLES
1 C04
1 C04      COMMON XS - BUFFER = LINE TO BE OUTPUT
1 C04                  BUFPTR = INDEX INTO BUFFER FOR NEXT ASCII DATA
1 C04                  KRETRN = RETURN INDICATOR FOR INTERNAL ROUTINES
1 C04
1 C**********
1 C05
1 C05        NOTES
1 C05
1 C05      USES ROUTINES
1 C05
1 C05          EXEC
1 C05          XRE14
1 C05          XRI6
1 C05          XRMOV
1 C05          XRNIB
1 C05          XRO6
1 C05          XRPCK
1 C05          XRUPK
1 C05
1 C**********
```

```
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
XILSD
```

```
833   1 * XILSD WILL LIST THE DATA ASSOCIATED WITH ONE ARGUMENT                                                    XILSD
834   1 * IS RETURNED AS A PROMPT.                                                                                XILSD
835   1 BEGIN XILSD                                                                                               XILSD
836   2    SET ARGUMENT NAME INTO BUFFER                                                                          XILSD
837   2    USE IOFLAG TO DETERMINE WHICH OF '@', '=', OR '=@'                                                     XILSD
838   2    WILL GO INTO THE PRINT BUFFER                                                                          XILSD
839   2    IF @-FLAG IS OFF INDICATING NO LITERAL DATA, THEN                                                      XILSD
840   3       IF A PARAMETER NAME IS SPECIFIED, THEN                                                              XILSD
841   4          PUT PARAMETER NAME INTO BUFFER                                                                   XILSD
842   4          IF S-FLAG IS ON INDICATING TWO SUBSCRIPTS, THEN                                                  XILSD
843   5             COMPUTE AND CONVERT TO CHARACTER FORMAT EACH SUBSCRIPT                                        XILSD
844   5             PUT SUBSCRIPT INTO BUFFER                                                                    XILSD
845   4          ELSE                                                                                             XILSD
846   5             IF LITDSP OF ARGUMENT IS > 0, THEN                                                            XILSD
847   6                COMPUTE AND CONVERT THIS SUBSCRIPT                                                         XILSD
848   6                PUT SUBSCRIPT INTO BUFFER                                                                  XILSD
849   5             ENDIF                                                                                         XILSD
850   4          ENDIF                                                                                            XILSD
851   4          WRITE OUT THE PRINT BUFFER BUILT                                                                 XILSD
852   3       ENDIF                                                                                               XILSD
853   2    ELSE                                                                                                   XILSD
854   3       LOCATE LITERAL LIST AND MASK                                                                        XILSD
855   3       IF SYMBOLIC STRING, THEN                                                                            XILSD
856   4          CALL XILSS TO PRINT SYMBOLIC STRING                                                              XILSD
857   3       ELSE                                                                                                XILSD
858   4          DO UNTIL ALL ELEMENTS PROCESSED                                                                  XILSD
859   5             DO UNTIL A BUFFER OF DATA HAS BEEN GENERATED, OR                                              XILSD
860   6                UNTIL ALL ELEMENTS PROCESSED                                                               XILSD
861   6                COMPUTE AND CONVERT THE SUBSCRIPT                                                          XILSD
862   6                IF MASK FOR ELEMENT INDICATES NO DATA, THEN                                                XILSD
863   7                   PUT "0," INTO BUFFER                                                                    XILSD
864   6                ELSE                                                                                       XILSD
865   7                   CONVERT THE DATA USING XRO6, XRE14, OR XRI6                                             XILSD
866   7                   PUT DATA AND "," INTO BUFFER                                                            XILSD
867   6                ENDIF                                                                                      XILSD
868   5             ENDDO                                                                                         XILSD
869   5             IF ALL ELEMENTS OF THIS ARGUMENT HAVE BEEN PROCESSED, THEN                                    XILSD
870   6                REMOVE THE TRAILING COMMA IN THE PRINT BUFFER                                              XILSD
871   5             ENDIF                                                                                         XILSD
872   5             WRITE OUT THE PRINT BUFFER BUILT                                                              XILSD
873   4          ENDDO                                                                                            XILSD
874   3       ENDIF                                                                                               XILSD
875   2    ENDIF                                                                                                  XILSD
876   1 END XILSD                                                                                                 XILSD
```

```
878    1 CD0
879    1 CD0       FORTRAN CALLING PROCEEDURE
880    1 CD0
881    1 CD0    CALL XILSS
882    1 CD0
883    1 C********
884    1 CD1
885    1 CD1    XILSS IS CALLED BY XILSC TO LIST SYMBOLIC STRING DATA
886    1 CD1
887    1 C********
888    1 CD2
889    1 CD2       INPUT
890    1 CD2
891    1 CD2    COMMON XE - LU
892    1 CD2    COMMON XB - DEBUG,  LISTLU, WKBUF
893    1 CD2    COMMON XS - BUFFER = PRINT LINE BUFFER ALREADY INITIALIZED WITH
894    1 CD2                        NAME =
895    1 CD2                BUFPTR = INDEX INTO BUFFER OF NEXT POSITIO
896    1 CD2                DATPTR = INDEX INTO WKBUF OF SYMBOLIC STRING DATA
897    1 CD2
898    1 C********
899    1 CD3
900    1 CD3       OUTPUT
901    1 CD3
902    1 CD3    COMMON XS - BUFFER, BUFPTR, DATPTR
903    1 CD3
904    1 C********
```

XILSS

```
1 CD4
1 CD4      INTERNAL VARIABLES                                                    XILSS
1 CD4                                                                            XILSS
1 CD4    CONTRL = CONTROL TABLE DESCRIBING DISPOSITION AND PROCESSING            XILSS
1 CD4             FOR EACH OF THE TOKEN VALUES 1-32. EACH CONTROL TABLE          XILSS
1 CD4             ENTRY IS 3 WORDS :                                             XILSS
1 CD4                   WORD 1 (SIZE) = NO. OF WORDS IN PRINT BUFFER             XILSS
1 CD4                   WORD 2 (FIELD)= CONTENTS TO GO INTO PRINT BUFFER         XILSS
1 CD4                                   OR FLAG DESCRIBING HOW TO COMPUTE        XILSS
1 CD4                                   THEM                                     XILSS
1 CD4                   WORD 3 (TOKSIZ) = NO. OF WORDS IN SYMBOLIC STRING        XILSS
1 CD4                                                                            XILSS
1 CD4                                                                            XILSS
1 CD4 ENTRY  TOKEN        WORD 1            WORD 2              WORD 3           XILSS
1 CD4  NO.                (SIZE)            (FIELD)             (TOKSIZ)         XILSS
1 CD4   1    INTEGER      3                -1=>CALL XRI6         2               XILSS
1 CD4   2    REAL         7                -2=>CALL XRE14        3               XILSS
1 CD4   3    DOUBLE       9                -3=>CALL XRD18        4               XILSS
1 CD4   4    ----         0                 0=>ERROR (INVALID)   0               XILSS
1 CD4   5    NAME         3                -4=>USE 3 WORDS       4               XILSS
1 CD4                                      FOLLOWING TOKEN                       XILSS
1 CD4   6    ----         0                 0=>ERROR (INVALID)   0               XILSS
1 CD4   7    ----         0                 0=>ERROR (INVALID)   0               XILSS
1 CD4   8    CHAR. STR. -1=>USE VALUE      -5=>USE SIZE WORDS  -1=>USE SIZE+2 WORDS  XILSS
1 CD4                   WORD FOLLOWING      BEGINNING 2 PAST                     XILSS
1 CD4                   TOKEN               TOKEN                                XILSS
1 CD4   9    ----         0                 0=>ERROR (INVALID)   0               XILSS
1 CD4  10    +            1                 +                    1               XILSS
1 CD4  11    -            1                 -                    1               XILSS
1 CD4  12    *            1                 *                    1               XILSS
1 CD4  13    /            1                 /                    1               XILSS
1 CD4  14    <            1                 <                    1               XILSS
1 CD4  15    >            1                 >                    1               XILSS
1 CD4  16    #            1                 #                    1               XILSS
1 CD4  17    @            1                 @                    1               XILSS
1 CD4  18    =            1                 =                    1               XILSS
1 CD4  19    &            1                 &                    1               XILSS
1 CD4  20    ?            1                 ?                    1               XILSS
1 CD4  21    (            1                 (                    1               XILSS
1 CD4  22    )            1                 )                    1               XILSS
1 CD4  23    ----         0                 0=>ERROR (INVALID)   0               XILSS
1 CD4  24    END SYM.STR. 0                -7=>END SYMB. STR.    0               XILSS
1 CD4  25    %            1                 %                    1               XILSS
1 CD4  26    BACKSLASH    1                 BACKSLASH            1               XILSS
1 CD4  27    $            1                 $                    1               XILSS
1 CD4  28    .            1                 .                    1               XILSS
1 CD4  29 LEFT BRACKET    1                LEFT BRACKET          1               XILSS
1 CD4  30 RIGHT BRACKET   1                RIGHT BRACKET         1               XILSS
1 CD4  31 REPEAT          4                -6=>CALL XRI6 AND     2               XILSS
1 CD4                                       APPEND "R "                          XILSS
1 CD4  32    ,            1                 ,                    1               XILSS
1 CD4                                                                            XILSS
```

```
959   1 C********                                                                                    XILSS
960   1 CD5                                                                                          XILSS
961   1 CD5       NOTES                                                                              XILSS
962   1 CD5                                                                                          XILSS
963   1 CD5    USES ROUTINES                                                                         XILSS
964   1 CD5                                                                                          XILSS
965   1 CD5       EXEC                                                                               XILSS
966   1 CD5       XRI6                                                                               XILSS
967   1 CD5       XRE14                                                                              XILSS
968   1 CD5       XRD18                                                                              XILSS
969   1 CD5       XRMSG                                                                              XILSS
970   1 CD5       XRMOV                                                                              XILSS
971   1 CD5                                                                                          XILSS
972   1 C********                                                                                    XILSS
973   1 BEGIN XILSS                                                                                  XILSS
974   2    MOVE A ' CHARACTER INTO PRINT BUFFER AND INCREMENT BUFFER POINTER                          XILSS
975   2    DO UNTIL ALL TOKENS OF SYMBOLIC STRING HAVE BEEN PROCESSED                                XILSS
976   2    EXIT TO ERROR 1 IF TOKEN VALUE IS < 1 OR > 32                                             XILSS
977   3       USE TOKEN VALUE TO RETRIVE 3 CONTROL WORDS (SIZE, FIELD, TOKSIZ)                       XILSS
978   2    EXIT TO ERROR 2 IF FIELD = 0                                                              XILSS
979   3       IF SIZE < 0, THEN                                                                      XILSS
980   4          SET SIZE TO VALUE IN WORD FOLLOWING THIS TOKEN IN THE SYMB. STRING                   XILSS
981   3       ENDIF                                                                                  XILSS
982   3       IF TOKSIZ < 0, THEN                                                                    XILSS
983   4          SET TOKSIZ TO SIZE + 2                                                              XILSS
984   3       ENDIF                                                                                  XILSS
985   3       IF THERE IS NOT ROOM IN PRINT BUFFER FOR SIZE WORDS, THEN                              XILSS
986   4          CALL EXEC TO WRITE PRINT BUFFER TO INDICATED DEVICE                                 XILSS
987   4          SET BUFFER POINTER TO 1ST POSITION FOR DATA                                         XILSS
988   4          CLEAR PRINT BUFFER TO BLANKS                                                        XILSS
989   3       ENDIF                                                                                  XILSS
990   3       IF FIELD > 0, THEN                                                                     XILSS
991   4          MOVE FIELD INTO CURRENT PRINT BUFFER POSITION                                       XILSS
992   3       ELSE                                                                                   XILSS
993   4          CASE (ONE, TWO, THREE, FOUR, FIVE, SIX, EXIT), -FIELD                               XILSS

994   5             :ONE: CALL XRI6 WITH VALUE IN NEXT WORD OF SYMB. STRING                          XILSS
995   5             AND PUT RESULTS INTO PRINT BUFFER                                                XILSS

996   5             :TWO: CALL XRE14 WITH VALUE IN NEXT 2 WORDS OF SYMB. STRING                      XILSS
997   5             AND PUT RESULTS INTO PRINT BUFFER                                                XILSS

998   5             :THREE: CALL XRD18 WITH VALUE IN NEXT 3 WORDS OF SYMB. STRING                    XILSS
999   5             AND PUT RESULTS INTO PRINT BUFFER                                                XILSS

1000  5             :FOUR: MOVE THE NEXT 3 WORDS OF SYMB. STRING INTO PRINT BUFFER                   XILSS

1001  5             :FIVE: MOVE SIZE WORDS FROM 2ND WORD PAST CURRENT TOKEN THE SYMB. STRING          XILSS

1002  5             :SIX: CALL XRI6 WITH VALUE IN NEXT WORD OF SYMB. STRING AND PUT RESULTS           XILSS
1003  5             INTO PRINT BUFFER FOLLOWED BY AN "R"                                             XILSS

1004  5             :EXIT: PUT A ' CHARACTER INTO THE PRINT BUFFER, INCREMENT THE BUFFER              XILSS
1005  5             INDEX BY 1                                                                       XILSS
1006  5             CALL EXEC TO WRITE THE PRINT BUFFER TO INDICATED DEVICE                          XILSS
1007  4          EXIT XILSS                                                                          XILSS
1008  4          ENDCASE                                                                             XILSS
1009  3       ENDIF                                                                                  XILSS
1010  3       INCREMENT PRINT BUFFER INDEX BY SIZE                                                   XILSS
```

```
1011    3       INCREMENT THE SYMBOLIC STRING INDEX BY TOKSIZ                    XILSS
1012    2     ENDDO                                                              XILSS
1013    1 END XILSS                                                              XILSS
```

```
1015    1 CDO                                                                   XICHR
1016    1 CDO       FORTRAN CALLING PROCEEDURE                                  XICHR
1017    1 CDO                                                                   XICHR
1018    1 CDO   CALL XICHR                                                      XICHR
1019    1 CDO                                                                   XICHR
1020    1 C*********                                                            XICHR
1021    1 CD1                                                                   XICHR
1022    1 CD1   PRINT THE CHARACTERISTICS OF AN ARGUMENT                        XICHR
1023    1 CD1                                                                   XICHR
1024    1 C*********                                                            XICHR
1025    1 CD2                                                                   XICHR
1026    1 CD2       INPUT                                                       XICHR
1027    1 CD2                                                                   XICHR
1028    1 CD2   COMMON XE - LU                                                  XICHR
1029    1 CD2                                                                   XICHR
1030    1 CD2   COMMON XB - ARGNO,  IOFLAG, ISIZE,  ITYPE,                      XICHR
1031    1 CD2               LENEFF, NARG                                        XICHR
1032    1 CD2                                                                   XICHR
1033    1 C*********                                                            XICHR
1034    1 CD5                                                                   XICHR
1035    1 CD5       NOTES                                                       XICHR
1036    1 CD5                                                                   XICHR
1037    1 CD5   USES ROUTINES                                                   XICHR
1038    1 CD5                                                                   XICHR
1039    1 CD5      EXEC                                                         XICHR
1040    1 CD5      XRIO6                                                        XICHR
1041    1 CD5      XRMOV                                                        XICHR
1042    1 CD5      XRPCK                                                        XICHR
1043    1 CD5      XRUPK                                                        XICHR
1044    1 CD5                                                                   XICHR
1045    1 C*********                                                            XICHR
1046    1 * WRITE ARGUMENT CHARACTERISTICS                                      XICHR
1047    1 BEGIN XICHR                                                           XICHR
1048    2    BUILD PRINT BUFFER WITH ARGUMENT NAME, SUBSCRIPTS, I/O TYPE AND    XICHR
1049    2    DATA TYPE                                                          XICHR
1050    2    WRITE OUT THE PRINT BUFFER                                         XICHR
1051    1 END XICHR                                                             XICHR
```

```
1053    1 CD0                                                                   XIEXT
1054    1 CD0       FORTRAN CALLING PROCEEDURE                                  XIEXT
1055    1 CD0                                                                   XIEXT
1056    1 CD0   CALL XIEXT                                                      XIEXT
1057    1 CD0                                                                   XIEXT
1058    1 CD0                                                                   XIEXT
1059    1 C*********                                                            XIEXT
1060    1 CD1                                                                   XIEXT
1061    1 CD1   EXTRACT VARIOUS FIELDS OF AN ARGUMENTS CHARACTERISTICS          XIEXT
1062    1 CD1   AND PUT VALUES INTO COMMON                                      XIEXT
1063    1 CD1                                                                   XIEXT
1064    1 C*********                                                            XIEXT
1065    1 CD2                                                                   XIEXT
1066    1 CD2       INPUT                                                       XIEXT
1067    1 CD2                                                                   XIEXT
1068    1 CD2   COMMON XB - ARGNO, ISIZES, WKBUF                                XIEXT
1069    1 CD2                                                                   XIEXT
1070    1 C*********                                                            XIEXT
1071    1 CD3                                                                   XIEXT
1072    1 CD3       OUTPUT                                                      XIEXT
1073    1 CD3                                                                   XIEXT
1074    1 CD3   COMMON XB - CFLAG,  DFLAG,  IARG,   IARG4,                      XIEXT
1075    1 CD3               ICLASS, IDIM,   IOFLAG, ISIZE,                      XIEXT
1076    1 CD3               ISUB,   ITYPE,  LENEFF, LITDSP,                     XIEXT
1077    1 CD3               LITSIZ, NDXBTM, NOBITM, SFLAG                       XIEXT
1078    1 CD3                                                                   XIEXT
1079    1 C*********                                                            XIEXT
1080    1 CD5                                                                   XIEXT
1081    1 CD5       NOTES                                                       XIEXT
1082    1 CD5                                                                   XIEXT
1083    1 CD5   USES ROUTINES                                                   XIEXT
1084    1 CD5                                                                   XIEXT
1085    1 CD5       IAND                                                        XIEXT
1086    1 CD5       XREXT                                                       XIEXT
1087    1 CD5                                                                   XIEXT
1088    1 C*********                                                            XIEXT
1089    1 * EXTRACT THE VARIOUS VALUES AND FLAGS ASSOCIATED WITH THIS           XIEXT
1090    1 * ARGUMENT                                                            XIEXT
1091    1 BEGIN XIEXT                                                           XIEXT
1092    2    USING THE ARGUMENT NO. (ARGNO),LOCATE THIS ARGUMENT'S CHARACTERISTICS XIEXT
1093    2    IN THE WORKING BUFFER                                              XIEXT
1094    2    EXTRACT EACH OF THE FIELDS INTO A WORD OF COMMON FOR GENERAL USEAGE XIEXT
1095    1 END XIEXT                                                             XIEXT
```

```
1097    1 CD0                                                                   XILIT
1098    1 CD0       FORTRAN CALLING PROCEEDURE                                  XILIT
1099    1 CD0                                                                   XILIT
1100    1 CD0   CALL XILIT                                                      XILIT
1101    1 CD0                                                                   XILIT
1102    1 C*********                                                            XILIT
1103    1 CD1                                                                   XILIT
1104    1 CD1   PACK LITERAL ENTRIES INTO FORMAT FOR STORAGE OF INTERFACE       XILIT
1105    1 CD1   TABLE INTO AWA                                                  XILIT
1106    1 CD1                                                                   XILIT
1107    1 C*********                                                            XILIT
1108    1 CD2                                                                   XILIT
1109    1 CD2       INPUTS                                                      XILIT
1110    1 CD2                                                                   XILIT
1111    1 CD2   COMMON XB - ARGNO,  DFLAG,  IARG4,  ISIZE,                      XILIT
1112    1 CD2               LENEFF, LITDSP, LITDWN, LITPTR,                     XILIT
1113    1 CD2               LITSIZ, NARG,   NDXBTM, NOBITM                      XILIT
1114    1 CD2               NUMARG, SFLAG,  WXBLNG, WKBUF                       XILIT
1115    1 CD2                                                                   XILIT
1116    1 C*********                                                            XILIT
1117    1 CD3                                                                   XILIT
1118    1 CD3       OUTPUTS                                                     XILIT
1119    1 CD3                                                                   XILIT
1120    1 CD3   COMMON XB - LITDWN, LITLEN, LITPTR, WKBUF                       XILIT
1121    1 CD3                                                                   XILIT
1122    1 C*********                                                            XILIT
1123    1 CD4                                                                   XILIT
1124    1 CD4       INTERNAL VARIABLES                                          XILIT
1125    1 CD4                                                                   XILIT
1126    1 CD4   COMMON XS - LITUP  = INDEX INTO WKBUF OF AREA FOR NEXT LITERAL  XILIT
1127    1 CD4                        ENTRY TO BE MOVED INTO                     XILIT
1128    1 CD4               LITUP1 = INDEX INTO WKBUF OF LITERAL ENTRY TO BE    XILIT
1129    1 CD4                        COMPRESSED AND MOVED                       XILIT
1130    1 CD4                                                                   XILIT
1131    1 C*********                                                            XILIT
1132    1 CD5                                                                   XILIT
1133    1 CD5       NOTES                                                       XILIT
1134    1 CD5                                                                   XILIT
1135    1 CD5   USES ROUTINES                                                   XILIT
1136    1 CD5                                                                   XILIT
1137    1 CD5       XIEXT                                                       XILIT
1138    1 CD5       XRBIT                                                       XILIT
1139    1 CD5       XRMOV                                                       XILIT
1140    1 CD5       XRNXB                                                       XILIT
1141    1 CD5       XRSET                                                       XILIT
1142    1 CD5                                                                   XILIT
1143    1 C*********                                                            XILIT
```

```
1145    1 * PACK LITERAL AREA INTO FORMAT FOR STORAGE OF INTERFACE TABLE                    XILIT
1146    1 BEGIN XILIT                                                                      XILIT
1147    2    DO UNTIL ALL LITERAL AREAS PROCESSED                                          XILIT
1148    3       DO UNTIL ALL ARGUMENTS SEARCHED                                            XILIT
1149    4          IF THIS LITERAL ENTRY BELONGS TO THIS ARGUMENT, THEN                    XILIT
1150    5             IF ALL ELEMENTS OF THIS ARGUMENT ARE COMPLETE, THEN                  XILIT
1151    6                MOVE ALL DATA FOR LITERAL ENTRY UP IN WORKING BUFFER               XILIT
1152    5             ELSE                                                                 XILIT
1153    6                DO UNTIL ALL BITS OF BIT MASK PROCESSED                           XILIT
1154    7                   IF THE BIT IS ON, THEN                                         XILIT
1155    8                      MOVE CORRECT NUMBER OF WORDS (LENEFF) OF LITERAL            XILIT
1156    8                      UP IN THE WORKING BUFFER                                    XILIT
1157    7                   ENDIF                                                          XILIT
1158    6                ENDDO                                                             XILIT
1159    5             ENDIF                                                                XILIT
1160    5             CALCULATE NEW DISPLACEMENT AND SET IN LITDSP                         XILIT
1161    4          ENDIF                                                                   XILIT
1162    3       ENDDO                                                                      XILIT
1163    2    ENDDO                                                                         XILIT
1164    1 END XILIT                                                                        XILIT
```

```
 SYMBOL DEFINITION TABLE
 :ERR02 :: 292
 :ERR02 :: 616
 :ERR02 :: 782
 :ERR02 :: 379
 :ERR02 :: 709
 :ERR07 :: 617
 :ERR08 :: 710
 :ERR10 :: 618
 :ERR14 :: 783
 :ERR14 :: 432
 :ERR15 :: 433
 :ERR16 :: 434
 :ERR18 :: 784
 :EXIT  :: 1004
 :EXIT  :: 685
 :FIVE  :: 1001
 :FOUR  :: 1000
 :LIST  :: 687
 :ONE   :: 994
 :PROMPT: 686
 :RETURN: 619
 :RETURN: 435
 :RETURN: 380
 :RETURN: 785
 :RETURN: 293
 :RETURN: 711
 :SIX   :: 1002
 :THREE :: 998
 :TWO   :: 996
  XICHR  1047
  XIDAT  502
  XIEXT  1091
  XILIT  1146
  XILSD  835
  XILSS  973
  XILST  338
  XINIX  113
  XINPT  664
  XINTE  35
  XIPAR  752
  XIPMT  276
  XIPRM  193
  XISUB  415

@XQT F.PDLIST
```

```
2    1 CD************                                                                  XLMAN
3    1 CDO                                                                          XLMAN
4    1 CDO      RTE RUN PROCEDURE FOR LIBRARY MAINTENANCE:                          XLMAN
5    1 CDO                                                                          XLMAN
6    1 CDO           :RU,XLMAN,LU,0,ISECU,FLAGS                                     XLMAN
7    1 CDO                                                                          XLMAN
8    1 CD************                                                               XLMAN
9    1 CD1                                                                          XLMAN
10   1 CD1      OFFLINE MAINTENANCE PROGRAM, XLMAN CREATES, DELETES AND             XLMAN
11   1 CD1      MODIFIES FDS FILES.                                                 XLMAN
12   1 CD1                                                                          XLMAN
13   1 CD************                                                               XLMAN
14   1 CD2                                                                          XLMAN
15   1 CD2      INPUTS FROM RUN SEQUENCE                                            XLMAN
16   1 CD2                                                                          XLMAN
17   1 CD2           LU  - LOGICAL UNIT NUMBER WHERE USER DESIRES                   XLMAN
18   1 CD2                 HIS INPUTS/OUTPUTS                                       XLMAN
19   1 CD2          ISECU - SECURITY CODE OF FDS LIBRARY FILES                      XLMAN
20   1 CD2          FLAGS - DEBUG FLAG:  0 - OFF                                    XLMAN
21   1 CD2                               4 - ON                                     XLMAN
22   1 CD2                                                                          XLMAN
23   1 CD************                                                               XLMAN
24   1 CD4                                                                          XLMAN
25   1 CD4      INTERNAL VARIABLES:                                                 XLMAN
26   1 CD4                                                                          XLMAN
27   1 CD4           PROM1 - (INTEGER, 31 WORDS) PROMPT FOR OPTION                  XLMAN
28   1 CD4                   TO BE EXECUTED                                         XLMAN
29   1 CD4           TEMP  - (INTEGER, 1 WORD) TEMPORARY USED FOR RESPONSE          XLMAN
30   1 CD4          TOKENZ - (INTEGER, 29 WORDS) THE 29 TOKENS TO BE                XLMAN
31   1 CD4                   INITIALIZED INTO COMMON                                XLMAN
32   1 CD4                                                                          XLMAN
33   1 CD************                                                               XLMAN
34   1 CD5                                                                          XLMAN
35   1 CD5      RTE FUNCTIONS AND SUBROUTINES USED:                                 XLMAN
36   1 CD5                                                                          XLMAN
37   1 CD5           EXEC,RMPAR                                                     XLMAN
38   1 CD5                                                                          XLMAN
39   1 CD5      FDS FUNCTIONS AND ROUTINES USED:                                    XLMAN
40   1 CD5                                                                          XLMAN
41   1 CD5           XELDS, XLDBF, XLDEL, XLINT, XLMOD, XLMSG,                      XLMAN
42   1 CD5           XLPRM, XLPRO, XTCOM, XRMOV, XRMSG                              XLMAN
43   1 CD5                                                                          XLMAN
44   1 CD5      XE COMMON USED:                                                     XLMAN
45   1 CD5                                                                          XLMAN
46   1 CD5           EQUIVALENCE       (XE(1),  LU    ),                          XLMAN
47   1 CD5          +(XE(2),   ICLASS),(XE(3),  ISECU ),                          XLMAN
48   1 CD5          +(XE(4),   FLAGS ),(XE(5),  MASSTA),                          XLMAN
49   1 CD5          +(XE(6),   SUBSTA),(XE(7),  NUMOR ),                          XLMAN
50   1 CD5          +(XE(7),   VALFLG),(XE(65), TOKENS),                          XLMAN
51   1 CD5          +(XE(142), ICR   ),(XE(145),COMBUF)                           XLMAN
52   1 CD5                                                                          XLMAN
53   1 CD************                                                               XLMAN
```

```
55    1 BEGIN XLMAN                                                                  XLMAN
56    2    CALL RMPAR TO GET INPUT PARAMETERS                                        XLMAN
57    2    INITIALIZE COMMON TO ZERO                                                 XLMAN
58    2    SET CLASS NUMBER TO ZERO                                                  XLMAN
59    2    CALL EXEC TO GET A CLASS NUMBER                                           XLMAN
60    1 EXIT XLMAN IF SECURITY CODE IS NOT VALID                                     XLMAN
61    2    CALL XRMOV TO INITIALIZE TOKENS IN COMMON                                 XLMAN
62    2    DO FOREVER                                                                XLMAN

63    3       :PROMPT:                                                               XLMAN
64    3       INITIALIZE MASTER AND SUBSTATE FLAGS                                   XLMAN
65    3       CALL XTCOM TO PROMPT FOR OPTION                                        XLMAN
66    3       IF XTCOM RETURN CODE IS NOT ZERO OR                                    XLMAN
67    4          FIRST TOKEN IS NOT AN INTEGER OR                                    XLMAN
68    4          INTEGER > 7 ;HEN                                                    XLMAN
69    4          CALL XRMSG TO WRITE INVALID RESPONSE                                XLMAN
70    4          GO TO :PROMPT:                                                      XLMAN
71    3       ENDIF                                                                  XLMAN
72    3       CASE INTEGER (:XLPRM:,:XLPRM:,:XLPRM:,:XLPRO:,:XLDEL:,:XLMOD:,         XLMAN
73    4 *                    :XLMSG:,:XLDBF:,:XLDBF:,:EXIT:)                         XLMAN

74    4          :XLPRM:                                                             XLMAN
75    4          SET NUMBR TO INTEGER                                                XLMAN
76    4          CALL XELDS TO LOAD XLPRM TO CREATE SYSTEM PROMPT FILE               XLMAN

77    4          :XLPRO:                                                             XLMAN
78    4          SET VALFLG TO SAY ORIGINAL XLPRO REQUEST                            XLMAN
79    4          CALL XELDS TO LOAD XLPRO TO ADD A PROCESSOR                         XLMAN
80    4          DO UNTIL VALFLG SAYS EXIT (X)                                       XLMAN
81    5             CALL XELDS TO LOAD XLINT TO ENTER DEFAULT VALUES                 XLMAN
82    5             CALL XELDS TO LOAD XLPRO TO COMPLETE PROCESSING                  XLMAN
83    4          ENDDO                                                               XLMAN

84    4          :XLDEL:                                                             XLMAN
85    4          CALL XELDS TO LOAD XLDEL TO DELETE A PROCESSOR                      XLMAN

86    4          :XLMOD:                                                             XLMAN
87    4          SET VALFLG TO SAY ORIGINAL XLMOD REQUEST                            XLMAN
88    4          CALL XELDS TO LOAD XLMOD TO MODIFY A PROCESSOR                      XLMAN
89    4          DO UNTIL VALFLG SAYS EXIT (X)                                       XLMAN
90    5             CALL XELDS TO LOAD XLINT TO ENTER DEFAULT VALUES                 XLMAN
91    5             CALL XELDS TO LOAD XLMOD TO COMPLETE PROCESSING                  XLMAN
92    4          ENDDO                                                               XLMAN

93    4          :XLMSG:                                                             XLMAN
94    4          CALL XELDS TO LOAD XLMSG TO ADD A MESSAGE                           XLMAN

95    4          :XLDBF:                                                             XLMAN
96    4          CALL XELDS TO LOAD XLDBF TO HANDLE DATA BASE FILES                  XLMAN
97    3       ENDCASE                                                                XLMAN
98    2    ENDDO                                                                     XLMAN

99    2    :EXIT:                                                                    XLMAN
100   2    CALL EXEC TO RELEASE CLASS NUMBER                                         XLMAN
101   1 END XLMAN                                                                    XLMAN
```

```
103   1 CD*************                                                                                    XLPRM
104   1 CD0                                                                                               XLPRM
105   1 CD0       FORTRAN CALLING PROCEDURE:                                                              XLPRM
106   1 CD0                                                                                               XLPRM
107   1 CD0            CALL XELDS ('XLPRM')                                                               XLPRM
108   1 CD0                                                                                               XLPRM
109   1 CD*************                                                                                    XLPRM
110   1 CD1                                                                                               XLPRM
111   1 CD1       CREATES ONE OF THE SYSTEM PROMPT FILES DEPENDING ON "NUMBER"                            XLPRM
112   1 CD1                                                                                               XLPRM
113   1 CD*************                                                                                    XLPRM
114   1 CD2                                                                                               XLPRM
115   1 CD2       INPUT FROM COMMON:                                                                      XLPRM
116   1 CD2                                                                                               XLPRM
117   1 CD2            NUMBER - (INTEGER, 1 WORD) USERS RESPONSE INDICATING                               XLPRM
118   1 CD2                     WHICH SYSTEM PROMPT FILE TO CREATE:                                       XLPRM
119   1 CD2                                                                                               XLPRM
120   1 CD2                      1 -  >XDPRM                                                              XLPRM
121   1 CD2                      2 -  >XIPRM                                                              XLPRM
122   1 CD2                      3 -  >XSPRM                                                              XLPRM
123   1 CD2                                                                                               XLPRM
124   1 CD*************                                                                                    XLPRM
125   1 CD4                                                                                               XLPRM
126   1 CD4       INTERNAL VARIABLES:                                                                     XLPRM
127   1 CD4                                                                                               XLPRM
128   1 CD4            COM3  - (INTEGER, 1 WORD) FIRST TOKEN IN COMBUF                                    XLPRM
129   1 CD4            COM4  - (INTEGER, 1 WORD) FIRST DATA IN COMBUF                                     XLPRM
130   1 CD4            MSG   - (INTEGER, 1 WORD) CONTAINS APPROPRIATE MESSAGE NUMBER                      XLPRM
131   1 CD4                     WITH WHICH TO CALL XRMSG                                                  XLPRM
132   1 CD4            NAME  - (INTEGER, 9 WORDS) 3 ELEMENT ARRAY, EACH ELEMENT                           XLPRM
133   1 CD4                     IS A SYSTEM PROMPT FILE NAME                                              XLPRM
134   1 CD4            NODIR - (INTEGER, 1 WORD) NUMBER OF DIRECTIVES                                     XLPRM
135   1 CD4            NOTOK - (INTEGER, 1 WORD) NUMBER OF TOKENS)                                        XLPRM
136   1 CD4                                                                                               XLPRM
137   1 CD4            PROMS - (INTEGERS) PROM2,PROM3,PROM4 ARE ALL USER PROMPT ARRAYS                    XLPRM
138   1 CD4                                                                                               XLPRM
139   1 CD*************                                                                                    XLPRM
140   1 CD5                                                                                               XLPRM
141   1 CD5       RTE FUNCTIONS AND SUBROUTINES USED:                                                     XLPRM
142   1 CD5                                                                                               XLPRM
143   1 CD5            KCVT,CLOSE,CREAT,PURGE,WRITF                                                       XLPRM
144   1 CD5                                                                                               XLPRM
145   1 CD5       FDS FUNCTIONS AND ROUTINES USED:                                                        XLPRM
146   1 CD5                                                                                               XLPRM
147   1 CD5            XRMOV,XRMSG,XTCOM                                                                  XLPRM
148   1 CD5                                                                                               XLPRM
149   1 CD5       COMMON USED:                                                                            XLPRM
150   1 CD5                                                                                               XLPRM
151   1 CD5            EQUIVALENCE                                                                        XLPRM
152   1 CD5           +(XE(3),   ISECU ),(XE(7),  NUMBR ),                                               XLPRM
153   1 CD5           +(XE(142), ICR   ),(XE(145),COMBUF),                                               XLPRM
154   1 CD5           +(XE(145), NOTOK), (XE(147),COM3  ),                                               XLPRM
155   1 CD5           +(XE(148), COM4  ),(XB(1),  IBUF  ),                                               XLPRM
156   1 CD5           +(XB(128), NODIR)                                                                  XLPRM
157   1 CD5                                                                                              XLPRM
158   1 CD*************                                                                                    XLPRM
```

```
160   1 BEGIN XLPRM                                                                    XLPRM

161   2    :PRM1:                                                                      XLPRM
162   2    CALL XTCOM TO PROMPT USER FOR # DIRECTIVES                                  XLPRM
163   2    ERREXIT IF RESPONSE NOT INTEGER 1-63 TO :ERR1:                              XLPRM
164   2    CALL CREAT TO CREATE PROMPT FILE                                            XLPRM
165   2    IF FILE ALREADY EXISTS THEN                                                 XLPRM
166   3       CALL XTCOM TO PROMPT USER FOR SCRATCH/CANCEL                             XLPRM
167   2    EXIT XLPRM IF RESPONSE IS CANCEL                                            XLPRM
168   3       CALL PURGE TO PURGE PROMPT FILE                                          XLPRM
169   3       ERREXIT IF PURGE ERROR TO :FILERR:                                       XLPRM
170   3       GO TO :PRM1:                                                             XLPRM
171   2    ELSE   (CREATE NEW FILE)                                                    XLPRM
172   3       ERREXIT IF CREAT ERROR TO :FILERR:                                       XLPRM
173   3       CALL XTCOM TO PROMPT USER FOR LIST OF DIRECTIVES                         XLPRM
174   3       ERREXIT IF LIST IS INCONSISTENT WITH # OF DIRECTIVES TO :ERR1:           XLPRM
175   3       DO FOR EACH DIRECTIVE                                                    XLPRM
176   4          ERREXIT IF RESPONSE IS NOT VALID DIRECTIVE TO :ERR1:                  XLPRM
177   4          CALL XRMOV TO MOVE DIRECTIVE INTO BUFFER                              XLPRM
178   3       ENDDO                                                                    XLPRM
179   3       CALL WRITF TO WRITE LIST OF DIRECTIVES TO FILE                           XLPRM
180   3       ERREXIT IF WRITF ERROR TO :FILERR:                                       XLPRM
181   3       DO FOR EACH DIRECTIVE                                                    XLPRM
182   4          CALL XTCOM TO PROMPT USER FOR DEFINITION                              XLPRM
183   4          DO UNTIL EOS IS REACHED IN RESPONSE                                   XLPRM
184   5             ERREXIT IF RESPONSE IS NOT A CHARACTER STRING TO :ERR1:            XLPRM
185   5             ERREXIT IF RESPONSE IS TOO LONG (>128) TO :ERR1:                   XLPRM
186   5             MOVE RESPONSE INTO BUFFER                                          XLPRM
187   5             SET CONTROL CHARACTERS IN BUFFER                                   XLPRM
188   5             ERREXIT IF NEXT RESPONSE IS NOT A COMMA TO :ERR1:                  XLPRM
189   4          ENDDO                                                                 XLPRM
190   4          SET REMAINDER OF BUFFER TO NULL                                       XLPRM
191   4          CALL WRITF TO WRITE DEFINITION                                        XLPRM
192   4          ERREXIT IF WRITF ERROR TO :FILERR:                                    XLPRM
193   3       ENDDO                                                                    XLPRM
194   3       CALL CLOSE TO CLOSE FILE                                                 XLPRM
195   3       ERREXIT IF CLOSE ERROR TO :FILERR:                                       XLPRM
196   3       CALL XRMSG TO DISPLAY FILE CREATED MESSAGE                               XLPRM
197   2    ENDIF                                                                       XLPRM
198   1 EXIT XLPRM                                                                     XLPRM

199   2    :ERR1:  (ERROR IN RESPONSE)                                                 XLPRM
200   2    CALL XRMSG TO DISPLAY ERROR IN RESPONSE                                     XLPRM
201   2    RETURN TO REPROMPT USER FOR ANOTHER RESPONSE                                XLPRM

202   2    :FILERR:   (FILE ACCESS ERROR)                                              XLPRM
203   2    CALL XRMSG TO DISPLAY ERROR                                                 XLPRM
204   2    CALL CLOSE TO CLOSE FILE                                                    XLPRM
205   2    CALL PURGE TO PURGE FILE                                                    XLPRM
206   1 END XLPRM                                                                      XLPRM
```

```
208    1 CD**************
209    1 CD0
210    1 CD0       FORTRAN CALLING SEQUENCE:
211    1 CD0
212    1 CD0             CALL KELDS ('XLPRO')
213    1 CD0
214    1 CD**************
215    1 CD1
216    1 CD1       XLPRO WILL ADD A PROCESSOR TO THE LIBRARY DIRECTORY. IF
217    1 CD1       THE PROCESSOR HAS A DEFAULT INTERFACE TABLE, IT WILL ALSO
218    1 CD1       CREATE A PROMPT FILE AND INTERFACE TABLE FILE
219    1 CD1
220    1 CD**************
221    1 CD2
222    1 CD2       INPUT FROM COMMON:
223    1 CD2
224    1 CD2             VALFLG - PROCESS CONTROL
225    1 CD2                           0 - ORIGINAL REQUEST TO ADD
226    1 CD2                           1 - COMPLETE PROCESSING
227    1 CD2
228    1 CD**************
229    1 CD4
230    1 CD4       INTERNAL VARIABLES:
231    1 CD4
232    1 CD4             ABSTR - (INTEGER, 128 WORDS) CONTAINS THE ABSTRACT OF THE
233    1 CD4                     PROCESSOR IN LATER BUILDS. PRESENTLY IS A NULL RECORD.
234    1 CD4            BLOCKS - (INTEGER , 1 WORD) NUMBER OF BLOCKS TO ALLOCATE
235    1 CD4                     TO INTERFACE TABLE FILE
236    1 CD4            DIRECT - (INTEGER, 6 WORDS) 3 NAME ARRAY CONTAINING VALID
237    1 CD4                     DIRECTIVES FOR THE INTERFACE TABLE EDITOR
238    1 CD4             NAMEV - (INTEGER, 3 WORDS) COMBINATION OF PROCESSOR
239    1 CD4                     NAME AND VERSION NUMBER
240    1 CD4             PRNAM - (INTEGER, 3 WORDS) PROCESSOR NAME
241    1 CD4             PROMS - (INTEGERS) PROMPTS FOR USER TO BE PROMPTED WITH
242    1 CD4              TYPE - (INTEGER, 1 WORD) TYPE OF PARAMETER USED IN
243    1 CD4                      CALCULATING SIZE
244    1 CD4              VERS - (INTEGER, 1 WORD) VERSION NUMBER OF PROCESSOR
245    1 CD4
246    1 CD**************
247    1 CD5
248    1 CD5       RTE AND FMGR ROUTINES USED
249    1 CD5
250    1 CD5             EXEC, IAND, KCVT, CREAT, OPEN,
251    1 CD5             READF, WRITF, CLOSE, PURGE
252    1 CD5
253    1 CD5       FDS ROUTINES USED:
254    1 CD5
255    1 CD5             XLIFL, XLINS, XLPFL, XLSPS
256    1 CD5             XRCPR, XRMOV, XRMSG, XRPCK, XRSET, XRUPK, XTCOM
257    1 CD5
258    1 CD5       COMMON USED:
259    1 CD5             EQUIVALENCE            (XE(3),   ISECU ),
260    1 CD5            *  (XE(5),   MASSTA),    (XE(6),   SUBSTA),
261    1 CD5            +  (XE(7),   VALFLG),    (XE(16),  PRCNAM),
262    1 CD5            +  (XE(85),  EOSTOK),    (XE(86),  INTTOK),
263    1 CD5            +  (XE(89),  NAMTOK),    (XE(113), CONTOK),
264    1 CD5            +  (XE(142), ICR   ),    (XE(145), COMBUF),
265    1 CD5            +  (XB(26),  VERS  ),    (XB(37),  ARGNO ),
266    1 CD5            +  (XB(38),  IT    ),    (XB(90),  LITPTR),
```

```
      1 CD5            + (XB(91),  LITLEN),                                XLPRO
      1 CD5            + (XB(96),  NOPARM),     (XB(101), MEDR  ),         XLPRO
      1 CD5            + (XB(101), LIBD1 ),     (XB(104), LIBD2 ),         XLPRO
      1 CD5            + (XB(108), PARMS ),                                XLPRO
      1 CD5            + (XS(1),   IERR  ),     (XS(2),   IDCB  )          XLPRO
      1 CD5                                                                XLPRO
      1 CD5***********                                                     XLPRO
```

```
275    1 BEGIN XLPRO                                                                    XLPRO
276    2    IF THIS ENTRY IS THE ORIGINAL XLPRO ENTRY FOR THIS PROCESSOR THEN           XLPRO
277    3       INITIALIZE MASTER AND SUB STATES                                         XLPRO
278    3       PERFORM LIBD TO UPDATE LIBRARY DIRECTORY                                 XLPRO
279    3       IF PROCESSOR HAS AN INTERFACE TABLE THEN                                 XLPRO
280    4          SET RETN = 2                                                          XLPRO

281    4          :PRMPT2:                                                              XLPRO
282    4          CALL XTCOM TO PROMPT FOR # PARAMETERS                                 XLPRO
283    4          ERREXIT IF XTCOM RETURN CODE IS NOT ZERO OR                           XLPRO
284    4          ERREXIT IF # PARAMETERS NOT INTEGER 1-63 TO :PRMERR:                  XLPRO
285    4          CREATE HEADER ENTRY WITH #PARAMETERS AND PROCESSOR NAME               XLPRO
286    4          CALL XLINS TO WRITE INSTRUCTIONS FOR ENTERING SPECS                   XLPRO
287    4          DO FOR # PARAMETERS                                                   XLPRO
288    5             CALL XLSPS TO CREATE ONE PARAMETER ENTRY                           XLPRO
289    4          ENDDO                                                                 XLPRO
290    4          SET CODES ARRAY TO ADD ABSTRACT AND PARAMETER DEFINITIONS             XLPRO
291    4          CALL XLPFL TO CREATE PROMPT FILE                                      XLPRO
292    4          SET RETN = 3                                                          XLPRO

293    4          :PRMPT3:                                                              XLPRO
294    4          CALL XTCOM FOR DEFAULT VALUES DECISION                                XLPRO
295    4          ERREXIT IF RETURN CODE IS NOT ZERO OR                                 XLPRO
296    4          ERREXIT IF RESPONSE IS NOT YE OR NO TO :PRMERR:                       XLPRO
297    4          IF RESPONSE WAS YE THEN                                               XLPRO
298    5             SET FLAG TO CALL INTERFACE TABLE EDITOR                            XLPRO
299    4          EXIT XLPRO                                                            XLPRO
300    4          ENDIF                                                                 XLPRO
301    3       ELSE                                                                     XLPRO
302    4          SET CODES ARRAY TO ADD ONLY ABSTRACT                                  XLPRO
303    4          CALL XLPFL TO CREATE PROMPT FILE                                      XLPRO
304    4          PERFORM XLPRO - NO RETURN EXPECTED                                    XLPRO
305    3       ENDIF                                                                    XLPRO
306    2    ENDIF                                                                       XLPRO
307    2    CALL XLIFL TO CREATE THE DEFAULT INTERFACE TABLE FILE                       XLPRO
308    2    SET VALFLAG TO SAY ORIGINAL REQUEST TO ADD A PROCESSOR                      XLPRO
309    2    PERFORM XLPRO - NO RETURN EXPECTED                                          XLPRO

310    2    :PRMERR:                                                                    XLPRO
311    2    CALL XRMSG TO DISPLAY ERROR MESSAGE                                         XLPRO
312    2    GO TO (:PRMPT1:,:PRMPT2:,:PRMPT3:), RETN                                    XLPRO
313    1 END XLPRO                                                                      XLPRO
```

```
315    1 BEGIN  LIBD                                                                    XLPRO
316    2    CALL OPEN TO OPEN LIBRARY DIRECTORY                                         XLPRO
317    2    IF RETURN CODE SAYS FILE NOT FOUND THEN                                     XLPRO
318    3       SET RECORD 1 TO ALL ZEROS                                                XLPRO
319    2    ELSE                                                                        XLPRO
320    3       ERREXIT IF FILE ERROR TO :FILERR:                                        XLPRO
321    3       CALL READF AND CLOSE TO READ IN LIBRARY DIRECTORY                        XLPRO
322    3       ERREXIT IF FILE ERROR TO :FILERR:                                        XLPRO
323    3       IF # PROCESSORS + 1 > 50 THEN                                            XLPRO
324    4          CALL XRMSG TO WRITE ERROR: TOO MANY PROCESSORS                        XLPRO
325    3       EXIT XLPRO                                                               XLPRO
326    3       ENDIF                                                                    XLPRO
327    3       SET RETN = 1                                                             XLPRO

328    3       :PRMPT1:                                                                 XLPRO
329    3       CALL XTCOM TO PROMPT FOR PROCESSOR NAME, VERSION, INT TABLE              XLPRO
330    2    EXIT XLPRO IF RETURN CODE SAYS X ENTERED                                    XLPRO
331    3       ERREXIT IF RETURN CODE > ZERO OR                                         XLPRO
332    3       ERREXIT IF PROCESSOR NAME IS NOT 6-CHAR NAME TO :PRMERR:                 XLPRO
333    3       CALL XRMOV TO MOVE PROCESSOR NAME INTO ENTRY                             XLPRO
334    3       ERREXIT IF VERSION IS NOT INTEGER VALUE 0-127 TO :PRMERR:                XLPRO
335    3       CALL XRSET TO SET VERSION IN ENTRY                                       XLPRO
336    3       ERREXIT IF INTERFACE TABLE OPTION IS NOT YE OR NO TO :PRMERR:            XLPRO
337    3       SET IT BIT = 0                                                           XLPRO
338    3       IF RESPONSE IS YES THEN                                                  XLPRO
339    4          SET IT BIT = 1                                                        XLPRO
340    3       ENDIF                                                                    XLPRO
341    3       CALL XRSET TO SET BIT ON/OFF                                             XLPRO
342    3       ERREXIT IF PROCESSOR NAME ALREADY EXISTS TO :PRMERR:                     XLPRO
343    3       INCREMENT # PROCESSORS BY 1                                              XLPRO
344    3       CALL XRMOV TO MOVE NEW ENTRY INTO  XLIBD                                 XLPRO
345    3       IF # PROCESSORS > 1 THEN                                                 XLPRO
346    4          CALL PURGE TO PURGE OLD FILE                                          XLPRO
347    4          ERREXIT IF RETURN CODE < ZERO TO :FILERR:                             XLPRO
348    3       ENDIF                                                                    XLPRO
349    3       CALL CREAT, WRITE AND CLOSE TO CREATE NEW LIBRARY DIRECTORY              XLPRO
350    3       ERREXIT IF FILE ERROR TO :FILERR:                                        XLPRO
351    2    ENDIF                                                                       XLPRO
352    1 END LIBD                                                                       XLPRO
```

```
1 CD************                                                         XLCDB
1 CD0                                                                    XLCDB
1 CD0       FORTRAN CALLING SEQUENCE:                                    XLCDB
1 CD0                                                                    XLCDB
1 CD0             CALL XLCDB                                             XLCDB
1 CD0                                                                    XLCDB
1 CD************                                                         XLCDB
1 CD1                                                                    XLCDB
1 CD1       XLCDB CREATES A NEW DATA BASE FILE (MDB/PDB) FROM AN OLD     XLCDB
1 CD1       DATA BASE FILE (MDB/PDB) AND DELETES THE OLD FILE            XLCDB
1 CD1                                                                    XLCDB
1 CD************                                                         XLCDB
1 CD2                                                                    XLCDB
1 CD2       INPUTS IN COMMON:                                            XLCDB
1 CD2                                                                    XLCDB
1 CD2             XE(3) QUAL,    XE(7) NUMBR,   XE(8) SECU               XLCDB
1 CD2             XE(142) ICR,   XB(3) OLDFIL,  XB(6) NEWFIL             XLCDB
1 CD2             XB(9) TOTSIZ                                           XLCDB
1 CD2                                                                    XLCDB
1 CD************                                                         XLCDB
1 CD4                                                                    XLCDB
1 CD4       INTERNAL VARIABLES:                                          XLCDB
1 CD4                                                                    XLCDB
1 CD4             XB(17) FREC - FIRST RECORD NUMBER IN DATBUF            XLCDB
1 CD4             XB(18) LREC - LAST RECORD NUMBER IN DATBUF             XLCDB
1 CD4             XB(40) IDCB - DCB FOR OLDFIL                           XLCDB
1 CD4             XB(56) IDCB2- DCB FOR NEWFIL                           XLCDB
1 CD4             XB(200)TOCBUF-BUFFOR FOR COMPLETE DATA BASE TOC        XLCDB
1 CD4                                                                    XLCDB
1 CD************                                                         XLCDB
1 CD5                                                                    XLCDB
1 CD5       RTE ROUTINES USED:                                           XLCDB
1 CD5                                                                    XLCDB
1 CD5             CLOSE, CREAT, KCVT,  OPEN,                             XLCDB
1 CD5             PURGE, READF, WRITF                                    XLCDB
1 CD5                                                                    XLCDB
1 CD5       FDS ROUTINES USED:                                           XLCDB
1 CD5                                                                    XLCDB
1 CD5             XDDBD, XREXT, XRMSG                                    XLCDB
1 CD5                                                                    XLCDB
1 CD5       COMMON USED:                                                 XLCDB
1 CD5                                                                    XLCDB
1 CD5             EQUIVALENCE         (XE(3),  QUAL  ),                  XLCDB
1 CD5            +(XE(7),   NUMBR ),  (XE(8),  SECU  ),                  XLCDB
1 CD5            +(XE(142),ICR    ),  (XB(3),  OLDFIL),                  XLCDB
1 CD5            +(XB(6),   NEWFIL),  (XB(9),  TOTSIZ),                  XLCDB
1 CD5            +(XB(10),  FILCHR),  (XB(12), QUALIF),                  XLCDB
1 CD5            +(XB(13),  NOTOC ),  (XB(14), SIZE  ),                  XLCDB
1 CD5            +(XB(15),  MSG   ),  (XB(16), TOCPTR),                  XLCDB
1 CD5            +(XB(17),  FREC  ),  (XB(18), LREC  ),                  XLCDB
1 CD5            +(XB(39),  IERR  ),  (XB(40), IDCB  ),                  XLCDB
1 CD5            +(XB(56),  IDCB2 ),  (XB(72), IBUF  ),                  XLCDB
1 CD5            +(XB(200),TOCBUF)                                       XLCDB
1 CD5                                                                    XLCDB
1 CD************                                                         XLCDB
```

```
410   1 BEGIN XLCDB                                                                                                                  XLCDB
411   2    CALL OPEN TO OPEN OLDFIL                                                                                                  XLCDB
412   2    ERREXIT IF OPEN ERROR TO :ERR3:                                                                                           XLCDB
413   2    CALL READF TO READ IN HEADER RECORD                                                                                       XLCDB
414   2    ERREXIT IF READF ERROR TO :ERR2:                                                                                          XLCDB
415   2    COMPUTE NUMBER OF TOC RECORDS (NOTOC) AS(# ENTRIES + 16)/ 16                                                              XLCDB
416   2    IF NOTOC > 1 THEN                                                                                                         XLCDB
417   3       CALL READF TO READ IN REMAINING TOC RECORDS                                                                            XLCD3
418   3       ERREXIT IF READF ERROR TO :ERR2:                                                                                       XLCDB
419   2    ENDIF                                                                                                                     XLCDB
420   2    SET # RECORDS LEFT TO READ (SIZE) AS TOTAL SIZE - NOTOC                                                                   XLCDB
421   2    CALL CREAT TO CREATE NEWFIL                                                                                               XLCDB
422   2    ERREXIT  IF CREAT ERROR TO :ERR2:                                                                                         XLCDB
423   2    DO FOR EACH TOC ENTRY                                                                                                     XLCDB
424   3       IF REQUEST WAS POB TO MOB THEN                                                                                         XLCDB
425   4          ERREXIT IF NAME > 4 CHARS OR                                                                                        XLCDB
426   4          ERREXIT IF CLASS IS DRDE AND NAME > 2 CHARS TO :ERR1:                                                               XLCDB
427   4          APPEND MDB CONVENTION TO FRONT OF NAME                                                                              XLCDB
428   3       ELSE                                                                                                                   XLCDB
429   4          REMOVE MDB CONVENTION FROM NAME                                                                                     XLCDB
430   3       ENDIF                                                                                                                  XLCDB
431   2    ENDDO                                                                                                                     XLCDB
432   2    CALL WRITE TO WRITE NEW TOC RECORDS TO NEWFIL                                                                             XLCDB
433   2    ERREXIT IF WRITF ERROR TO :ERR1:                                                                                          XLCDB
434   2    SET TOCPTR TO FIRST TOC ENTRY                                                                                             XLCDB
435   2    SET FIRST RECORD NUMBER AND LAST RECORD NUMBER                                                                            XLCDB
436   2    CONVERT SIZE FROM BLOCKS TO WORDS                                                                                         XLCDB
437   2    DO UNTIL ALL RECORDS ARE COPIED (SIZE = 0)                                                                                XLCDB
438   3       SET LENGTH TO MAXIMUM SIZE OF 1 READ/WRITE (1024)                                                                      XLCDB
439   3       IF SIZE < LENGTH THEN                                                                                                  XLCDB
440   4          SET LENGTH TO SIZE                                                                                                  XLCDB
441   3       ENDIF                                                                                                                  XLCDB
442   3       CALL READF TO READ LENGTH WORDS OF DATA                                                                                XLCDB
443   3       ERREXIT IF READF ERROR TO :ERR1:                                                                                       XLCDB
444   3       DECREMENT SIZE BY LENGTH READ                                                                                          XLCDB
445   3       UPDATE FIRST AND LAST RECORD NUMBERS                                                                                   XLCDB
446   3       STARTSEARCH FOR TOCPTR = TOCPTR TO LAST TOC ENTRY                                                                      XLCDB
447   3       EXIT IF RECORD # IN TOC ENTRY > LAST RECORD NUMBER                                                                     XLCDB
448   4          IF CLASS IS AN INTERFACE TABLE THEN                                                                                 XLCDB
449   5             COMPUTE INDEX INTO DATBUF FROM FIRST RECORD #, RECORD # IN TOC ENTRY                                             XLCDB
450   5             AND INDEX IN TOC ENTRY                                                                                           XLCDB
451   5             SET NAME IN INTERFACE TABLE TO NAME IN TOC ENTRY                                                                 XLCDB
452   4          ENDIF                                                                                                               XLCDB
453   3       ENDSEARCH                                                                                                              XLCDB
454   3       CALL WRITF TO WRITE LENGTH WORDS OF DATA TO NEWFIL                                                                     XLCDB
455   3       ERREXIT IF WRITF ERROR TO :ERR1:                                                                                       XLCDB
456   2    ENDDO                                                                                                                     XLCDB
```

```
458   2    CALL CLOSE TO CLOSE NEWFIL                                          XLCDB
459   2    CALL CLOSE TO CLOSE OLDFIL                                          XLCDB
460   2    CALL XDDBD TO DELETE OLDFIL FROM PDB LOG FILE                       XLCDB
461   2    CALL PURGE TO PURGE OLDFIL FROM SYSTEM                              XLCDB
462   1 EXIT XLCDB                                                             XLCDB

463   2    :ERR1:                                                              XLCDB
464   2    CALL CLOSE TO CLOSE NEWFIL                                          XLCDB
465   2    CALL PURGE TO PURGE NEWFIL                                          XLCDB

466   2    :ERR2:                                                              XLCDB
467   2    CALL CLOSE TO CLOSE OLDFIL                                          XLCDB

468   2    :ERR3:                                                              XLCDB
469   2    IF REQUEST WAS PDB TO MDB THEN                                      XLCDB
470   3       SET QUAL TO  SAY DELETE MDB FILE                                 XLCDB
471   2    ELSE  ( REQUEST WAS MDB TO PDB)                                     XLCDB
472   3       SET QUAL TO  SAY DELETE PDB FILE                                 XLCDB
473   2    ENDIF                                                               XLCDB
474   2    CALL XDDBD TO DELETE MDB/PDB FROM LOG FILE                          XLCDB
475   2    IF ERROR WAS FILE MANAGER THEN                                      XLCDB
476   3       CALL XRMSG TO DISPLAY  ERROR AND RETURN CODE                     XLCDB
477   2    ELSE                                                                XLCDB
478   3       CALL XRMSG TO DISPLAY ERROR                                      XLCDB
479   2    ENDIF                                                               XLCDB
480   1 END  XLCDB                                                             XLCDB
```

```
1 CD************                                                          XLDBF
1 CD0                                                                     XLDBF
1 CD0      FORTRAN CALLING SEQUENCE:                                      XLDBF
1 CD0                                                                     XLDBF
1 CD0            CALL XELDS ('XLDBF')                                     XLDBF
1 CD0                                                                     XLDBF
1 CD************                                                          XLDBF
1 CD1                                                                     XLDBF
1 CD1      XLDBF IS THE DATA BASE FILES MAIN ROUTINE. IT DETERMINES WHICH XLDBF
1 CD1      DATA BASE ROUTINE IS NEEDED AND GATHERS INPUTS NEEDED BY THAT  XLDBF
1 CD1      ROUTINE. HANDLES ALL PDB LOG FILE RELATED TRANSACTIONS.        XLDBF
1 CD1                                                                     XLDBF
1 CD************                                                          XLDBF
1 CD2                                                                     XLDBF
1 CD2      INPUTS IN COMMON:                                              XLDBF
1 CD2                                                                     XLDBF
1 CD2            XE(3) ISECU,   XE(7) NUMBR,                              XLDBF
1 CD2            XE(85)TOKENS,  XE(142) ICR                               XLDBF
1 CD2                                                                     XLDBF
1 CD************                                                          XLDBF
1 CD4                                                                     XLDBF
1 CD4      INTERNAL VARIABLES IN COMMON                                   XLDBF
1 CD4                                                                     XLDBF
1 CD4            XE(3) QUAL - USER QUALIFIER REQUIRED BY XDDB ROUTINES AND XRQFN XLDBF
1 CD4            XE(8) SECU - TEMPORARY SLOT FOR ISECU DURING THIS OVERLAY XLDBF
1 CD4            XB(3) OLDFIL - OLD FILE NAME                             XLDBF
1 CD4            XB(6) NEWFIL - NEW FILE NAME                             XLDBF
1 CD4            XB(9) TOTSIZ - SIZE OF OLDFIL/NEWFIL IN BLOCKS           XLDBF
1 CD4            XB(10)FILCHR - 4 BASE CHARACTERS OF FILE NAME            XLDBF
1 CD4            XB(12)QUALIF - QUALIFIER ENTERED BY USER                 XLDBF
1 CD4                                                                     XLDBF
1 CD************                                                          XLDBF
1 CD5                                                                     XLDBF
1 CD5      RTE ROUTINES USED:                                             XLDBF
1 CD5                                                                     XLDBF
1 CD5            KCVT, OPEN                                               XLDBF
1 CD5                                                                     XLDBF
1 CD5      FDS ROUTINES USED:                                             XLDBF
1 CD5                                                                     XLDBF
1 CD5            XDDBA, XDDBV, XERTN, XLCDB, XLPCR                        XLDBF
1 CD5            XLPMC, XREXT, XRMSG, XRQFN, XRSFR, XTCOM                 XLDBF
1 CD5                                                                     XLDBF
1 CD5      COMMON USED:                                                   XLDBF
1 CD5                                                                     XLDBF
1 CD5            EQUIVALENCE          (XE(3),   ISECU ),                  XLDBF
1 CD5           +(XE(3)    QUAL  ),   (XE(7),   NUMBR ),                  XLDBF
1 CD5           +(XE(8)    SECU  ),   (XE(85),  TOKENS),                  XLDBF
1 CD5           +(XE(142),ICR    ),   (XE(145)  COMBUF),                  XLDBF
1 CD5           +(XB(3),   OLDFIL),   (XB(6),   NEWFIL),                  XLDBF
1 CD5           +(XB(9),   TOTSIZ),   (XB(10),  FILCHR),                  XLDBF
1 CD5           +(XB(12),  QUALIF),   (XB(99),  IERR  ),                  XLDBF
1 CD5           +(XB(100),IDCB   )                                        XLDBF
1 CD5                                                                     XLDBF
1 CD************                                                          XLDBF
```

```
537   1 BEGIN XLDBF                                                                                                    XLDBF
538   2    IF REQUEST IS TO CREATE/MODIFY LOG FILE THEN                                                                XLDBF
539   3       CALL OPEN TO OPEN MDB/PDB LOG FILE                                                                       XLDBF
540   3       IF OPEN ERROR SAYS FILE NOT FOUND THEN                                                                   XLDBF
541   4          CALL XLPCR TO CREATE MDB/PDB LOG FILE                                                                 XLDBF
542   3       ELSE                                                                                                     XLDBF
543   4          ERREXIT IF OPEN ERROR TO :FILERR:                                                                     XLDBF
544   4          CALL XLPMO TO MODIFY MDB/PDB LOG FILE                                                                 XLDBF
545   3       ENDIF                                                                                                    XLDBF
546   2    ELSE                                                                                                        XLDBF
547   3       DO UNTIL USER REQUESTS EXIT (%)                                                                          XLDBF
548   4          CALL XTCOM TO PROMPT USER FOR NAME AND USER ID                                                        XLDBF
549   4          IF RESPONSE IS NOT EXIT (PERCENT) THEN                                                                XLDBF
550   5             ERREXIT IF RESPONSE IS INVALID TO :ERR1:                                                           XLDBF
551   5             SAVE 4 CHARACTER NAME AND ID IN COMMON                                                             XLDBF
552   5             IF REQUEST WAS PDB TO MDB THEN                                                                     XLDBF
553   6                SET QUALIFIER TO SEARCH FOR PDB NAME                                                            XLDBF
554   5             ELSE  (REQUEST WAS FOR MDB TO PDB)                                                                 XLDBF
555   6                SET QUALIFIER TO SEARCH FOR MDB NAME                                                            XLDBF
556   5             ENDIF                                                                                              XLDBF
557   5             CALL XDDBV TO VERIFY EXISTANCE OF MDB/PDB DEPENDING ON QUALIFIER                                   XLDBF
558   5             ERREXIT IF NAME WAS NOT FOUND TO :ERR1:                                                            XLDBF
559   5             ERREXIT IF FILE MANAGER ERROR TO :FILERR:                                                          XLDBF
560   5             IF REQUEST WAS PDB TO MDB THEN                                                                     XLDBF
561   6                SET QUALIFIER TO ADD MDB TO LOG FILE                                                            XLDBF
562   5             ELSE  (REQUEST WAS MDB TO PDB)                                                                     XLDBF
563   6                SET QUALIFIER TO ADD PDB TO LOG FILE                                                            XLDBF
564   5             ENDIF                                                                                              XLDBF
565   5             CALL XDDBA TO ADD MDB/PDB NAME TO LOG FILE DEPENDING ON QUALIFIER                                  XLDBF
566   5             ERREXIT IF DUPLICATE NAME OF                                                                       XLDBF
567   5             ERREXIT IF MAXIMUM NUMBER OF ENTRIES EXIST TO :ERR1:                                               XLDBF
568   5             ERREXIT IF FILE MANAGER ERROR TO :FILERR:                                                          XLDBF
569   5             IF REQUEST WAS PDB TO MDB THEN                                                                     XLDBF
570   6                CALL XRQFN TO SET OLDFIL TO PDB NAME                                                            XLDBF
571   6                SET NEWFIL TO MDB NAME                                                                          XLDBF
572   5             ELSE  (REQUEST WAS MDB TO PDB)                                                                     XLDBF
573   6                SET OLDFIL TO MDB NAME                                                                          XLDBF
574   6                CALL XRQFN TO SET NEWFIL TO PDB NAME                                                            XLDBF
575   5             ENDIF                                                                                              XLDBF
576   5             CALL XLCDB TO COPY OLDFIL TO NEWFIL                                                                XLDBF
577   4          ENDIF                                                                                                 XLDBF
578   3       ENDDO                                                                                                    XLDBF
579   2    ENDIF                                                                                                       XLDBF
580   1 EXIT XLDBF                                                                                                     XLDBF

581   2    :ERR1:                                                                                                      XLDBF
582   2    CALL XRMSG TO DISPLAY ERROR                                                                                 XLDBF
583   2    RETURN TO PROMPT FOR ANOTHER 4 CHARACTERS AND USER ID                                                       XLDBF

584   2    :FILERR:                                                                                                    XLDBF
585   2    CALL XRMSG TO DISPLAY FILE ACCESS ERROR                                                                     XLDBF
586   1 END XLDBF                                                                                                      XLDBF
```

```
588   1 CD*************                                                                XLDEL
589   1 CDO                                                                            XLDEL
590   1 CDO       FORTRAN CALLING PROCEDURE FOR DELETE PROCESSOR                       XLDEL
591   1 CDO                                                                            XLDEL
592   1 CDO             CALL XELDS ('XLDEL')                                           XLDEL
593   1 CDO                                                                            XLDEL
594   1 CD*************                                                                XLDEL
595   1 CD1                                                                            XLDEL
596   1 CD1       XLDEL DELETES A PROCESSOR FROM THE LIBRARY DIRECTORY AND             XLDEL
597   1 CD1       THE PROMPT FILE. IF THE PROCESSOR HAS AN INTERFACE TABLE,            XLDEL
598   1 CD1       IT DELETES THE DEFAULT INTERFACE TABLE FILE ALSO.                    XLDEL
599   1 CD1                                                                            XLDEL
600   1 CD*************                                                                XLDEL
601   1 CD4                                                                            XLDEL
602   1 CD4       INTERNAL VARIABLES:                                                  XLDEL
603   1 CD4             COMNAM - (INTEGER, 7 WORDS) IS A TEMPORARY WORK AREA           XLDEL
604   1 CD4                      WHERE ENTRY IN LIBRARY DIRECTORY IS STRIPPED         XLDEL
605   1 CD4                      DOWN TO PROCESSOR NAME                               XLDEL
606   1 CD4             PRNAM -  (INTEGER, 2 WORDS) PROCESSOR NAME TO BE              XLDEL
607   1 CD4                      DELETED.                                             XLDEL
608   1 CD4                                                                           XLDEL
609   1 CD*************                                                               XLDEL
610   1 CD5                                                                           XLDEL
611   1 CD5       RTE/ FMGR ROUTINES USED:                                            XLDEL
612   1 CD5                                                                           XLDEL
613   1 CD5             IAND,KCVT,CREAT,OPEN,READF,WRITF,CLOSE,PURGE                  XLDEL
614   1 CD5                                                                           XLDEL
615   1 CD5       FDS ROUTINES USED:                                                  XLDEL
616   1 CD5                                                                           XLDEL
617   1 CD5             XRCPR, XREXT, XRMOV, XRMSG, XRPCK, XRSET, XRUPK, XTCOM        XLDEL
618   1 CD5                                                                           XLDEL
619   1 CD5       XE AND XB COMMON USED                                               XLDEL
620   1 CD5             EQUIVALENCE              (XE(3),   ISECU ),                   XLDEL
621   1 CD5            +      (XE(142),  ICR   ), (XE(145), COMBUF),                  XLDEL
622   1 CD5            +      (XS(48),   LIBD1 ), (XB(51),  LIBD2 )                   XLDEL
623   1 CD5                                                                           XLDEL
624   1 CD*************                                                               XLDEL
```

```
626   1 BEGIN XLDEL                                                                    XLDEL

627   2    :PRMPT:                                                                     XLDEL
628   2    CALL XTCOM TO PROMPT FOR PROCESSOR NAME                                     XLDEL
629   2    IF XTCOM RETURN CODE IS NOT ZERO OR                                         XLDEL
630   3       RESPONSE IS NOT A VALID PROCESSOR NAME THEN                              XLDEL
631   3       CALL XRMSG TO WRITE ERROR MESSAGE                                        XLDEL
632   3       GO TO :PRMPT:                                                            XLDEL
633   2    EXIT XLDEL IF RETURN CODE SAYS % ENTERED                                    XLDEL
634   2    ENDIF                                                                       XLDEL
635   2    CALL OPEN, READF AND CLOSE TO READ IN LIBRARY DIRECTORY                     XLDEL
636   2    ERREXIT IF FILE ERROR TO :FILERR:                                           XLDEL
637   2    IF PROCESSOR IS NOT IN LIBRARY DIRECTORY THEN                               XLDEL
638   3       CALL XRMSG TO DISPLAY ERROR                                              XLDEL
639   3       GO TO :PRMPT:                                                            XLDEL
640   2    ENDIF                                                                       XLDEL
641   2    DECREMENT # PROCESSORS BY 1                                                 XLDEL
642   2    CALL PURGE TO PURGE OLD LIBRARY DIRECTORY                                   XLDEL
643   2    ERREXIT IF RETURN CODE < ZERO TO :FILERR:                                   XLDEL
644   1 EXIT XLDEL IF #PROCESSORS IS ZERO                                              XLDEL
645   2    CALL CREAT, WRITF AND CLOSE TO RECREATE LIBRARY DIRECTORY                   XLDEL
646   2    ERREXIT IF FILE ERROR TO :FILERR:                                           XLDEL
647   2    CREATE THE PROMPT FILE NAME                                                 XLDEL
648   2    CALL PURGE TO PURGE THE PROMPT FILE                                         XLDEL
649   2    ERREXIT IF PURGE ERROR TO :FILERR:                                          XLDEL
650   2    CALL XRMSG TO SAY FILE PURGED SUCCESSFULLY                                  XLDEL
651   2    IF PROCESSOR HAD AN INTERFACE TABLE THEN                                    XLDEL
652   3       CREATE DEFAULT IT NAME                                                   XLDEL
653   3       CALL PURGE TO PURGE DEFAULT IT                                           XLDEL
654   3       ERREXIT IF RETURN CODE IS NOT ZERO TO :FILERR:                           XLDEL
655   3       CALL XRMSG TO DISPLAY 'FILE PURGED' MESSAGE                              XLDEL
656   2    ENDIF                                                                       XLDEL
657   2    GO TO :PRMPT:                                                               XLDEL

658   2    :FILERR: CALL XRMSG TO WRITE FILE ACCESS ERROR                              XLDEL
659   1 END XLDEL                                                                      XLDEL
```

```
661  1 C0**************                                                    XLIFL
662  1 C00                                                                 XLIFL
663  1 C00      FORTRAN CALLING SEQUENCE:                                  XLIFL
664  1 C00                                                                 XLIFL
665  1 C00          CALL XLIFL                                             XLIFL
666  1 C00                                                                 XLIFL
667  1 C0**************                                                    XLIFL
668  1 C01                                                                 XLIFL
669  1 C01      XLIFL CREATES THE DEFAULT INTERFACE TABLE FILE             XLIFL
670  1 C01                                                                 XLIFL
671  1 C0**************                                                    XLIFL
672  1 C04                                                                 XLIFL
673  1 C04      INTERNAL VARIABLES                                         XLIFL
674  1 C04                                                                 XLIFL
675  1 C04          BLOCKS - (INTEGER, 1 WORD) # BLOCKS TO BE ALLOCATED TC XLIFL
676  1 C04                   THE FILE                                      XLIFL
677  1 C0                                                                  XLIFL
678  1 C0**************                                                    XLIFL
679  1 C05                                                                 XLIFL
680  1 C05      FDS ROUTINES USED:                                         XLIFL
681  1 C05                                                                 XLIFL
682  1 C05          XREXT, XRMSG                                           XLIFL
683  1 C05                                                                 XLIFL
684  1 C05      RTE ROUTINES USED:                                         XLIFL
685  1 C05                                                                 XLIFL
686  1 C05          CLOSE, CREAT, WRITF                                    XLIFL
687  1 C05                                                                 XLIFL
688  1 C05      COMMON USED:                                               XLIFL
689  1 C05                                                                 XLIFL
690  1 C05          EQUIVALENCE (XE(3),   ISECU ),                         XLIFL
691  1 C05         + (XE(142),  ICR   ),  (XB(90),  LITPTR),               XLIFL
692  1 C05         + (XB(91),   LITLEN),                                   XLIFL
693  1 C05         + (XB(96),   NOPARM),  (XB(101), HEDP  ),               XLIFL
694  1 C05         + (XB(108),  PARMS ),  (XS(1),   IERR  ),               XLIFL
695  1 C05         + (XS(2),    IDCB  )                                    XLIFL
696  1 C05                                                                 XLIFL
697  1 C0**************                                                    XLIFL
```

```
699    1 BEGIN XLIFL                                                                    XLIFL
700    2    EXTRACT LITERAL LENGTH FROM HEADER                                          XLIFL
701    2    COMPUTE # BLOCKS FOR THIS FILE                                              XLIFL
702    2    CALL CREAT TO CREATE DEFAULT INT TABLE FILE                                 XLIFL
703    2    ERREXIT IF RETURN CODE < ZERO TO :FILERR:                                   XLIFL
704    2    CALL WRITF TO WRITE HEADER AND SPECS                                        XLIFL
705    2    ERREXIT IF RETURN CODE < ZERO TO :FILERR:                                   XLIFL
706    2    IF THERE IS A LITERAL RECORD THEN                                           XLIFL
707    3       CALL WRITF TO WRITE LITERALS                                             XLIFL
708    3       ERREXIT IF RETURN CODE < ZERO TO :FILERR:                                XLIFL
709    2    ENDIF                                                                       XLIFL
710    2    CALL CLOSE                                                                  XLIFL
711    2    ERREXIT IF RETURN CODE < ZERO TO :FILERR:                                   XLIFL
712    2    CALL XRMSG TO DISPLAY FILE (NAME) CREATED MESSAGE                           XLIFL
713    1 EXIT XLIFL                                                                     XLIFL

714    2    :FILERR:                                                                    XLIFL
715    2    CALL XRMSG TO WRITE FILE ACCESS ERROR                                       XLIFL
716    2    CALL XERTN TO RETURN TO MAIN PROGRAM                                        XLIFL
717    1 END XLIFL                                                                      XLIFL
```

```
1 CD*************                                                                XLINS
1 CDO                                                                           XLINS
1 CDO       FORTRAN CALLING SEQUENCE:                                           XLINS
1 CDO                                                                           XLINS
1 CDO             CALL XLINS                                                    XLINS
1 CDO                                                                           XLINS
1 CD*************                                                               XLINS
1 CD1                                                                           XLINS
1 CD1       XLINS DISPLAYS INSTRUCTIONS FOR ENTERING PARAMETER SPECS             XLINS
1 CD1                                                                           XLINS
1 CD*************                                                               XLINS
1 CD4                                                                           XLINS
1 CD4       INTERNAL VARIABLES:                                                 XLINS
1 CD4                                                                           XLINS
1 CD4             PROMPT - (INTEGER, 120 WORDS) ALL 7 LINES IN AN ARRAY         XLINS
1 CD4                      TO BE DISPLAYED                                      XLINS
1 CD4                                                                           XLINS
1 CD*************                                                               XLINS
1 CD5                                                                           XLINS
1 CD5       RTE ROUTINES USED:                                                  XLINS
1 CD5                                                                           XLINS
1 CD5             EXEC                                                          XLINS
1 CD5                                                                           XLINS
1 CD5       COMMON USED:                                                        XLINS
1 CD5                                                                           XLINS
1 CD5             EQUIVALENCE   (XE(1),  LU    )                              XLINS
1 CD5                                                                           XLINS
1 CD*************                                                               XLINS
1 *                                                                             XLINS
1 *                                                                             XLINS
1 *                                                                             XLINS
1 *                                                                             XLINS
1 BEGIN XLINS                                                                   XLINS
2    CALL EXEC TO DISPLAY ALL 7 LINES                                           XLINS
1 END XLINS                                                                     XLINS
```

```
755    1 CD*************                                                                    XLINT
756    1 CDO                                                                               XLINT
757    1 CDO      FORTRAN CALLING PROCEDURE:                                               XLINT
758    1 CDO                                                                               XLINT
759    1 CDO            CALL  XELDS (XLINT)                                                XLINT
760    1 CDO                                                                               XLINT
761    1 CDO                                                                               XLINT
762    1 CD*************                                                                   XLINT
763    1 CD1                                                                               XLINT
764    1 CD1      XLINT SEGMENT SETS UP COMMON TO CALL THE INTERFACE TABLE TO               XLINT
765    1 CD1      ACCEPT DEFAULT VALUES FOR THE INTERFACE TABLE                            XLINT
766    1 CD1                                                                               XLINT
767    1 CD*************                                                                   XLINT
768    1 CDS                                                                               XLINT
769    1 CDS      FDS FUNCTIONS AND SUBROUTINES USED:                                      XLINT
770    1 CDS                                                                               XLINT
771    1 CDS            XEINT, XERTN, XINIX, XRMOV                                         XLINT
772    1 CDS                                                                               XLINT
773    1 CDS      COMMON USED:                                                             XLINT
774    1 CDS                                                                               XLINT
775    1 CDS            EQUIVALENCE          (XE(5),  MASSTA),                             XLINT
776    1 CDS           +(XE(6),  SUBSTA)     (XB(1),  NUMDIR),                             XLINT
777    1 CDS           +(XB(2),  DIRECT),    (XB(25), LSTFLG),                             XLINT
778    1 CDS           +(XE(37), ARGNO ),    (XB(41), PRMTMD),                             XLINT
779    1 CDS           +(XB(73), ISIZES),    (XB(89), LITDWN),                             XLINT
780    1 CDS           +(XB(90), LITPTR),    (XB(91), LITLEN),                             XLINT
781    1 CDS           +(XB(92), MAPE  ),                                                  XLINT
782    1 CDS           +(XB(96), NUMARG),    (XB(97), NEWTAB),                             XLINT
783    1 CDS           +(XB(100),WKBLNG),    (XB(101),WKBUF ),                             XLINT
784    1 CDS           +(XB(1400),END  )                                                   XLINT
785    1 CDS                                                                               XLINT
786    1 CD*************                                                                   XLINT
```

```
788  1 BEGIN XLINT                                                    XLINT
789  1   SET MASTER STATE AS EXEC                                     XLINT
790  2   SET SUBSTATE  AS INTERFACE TABLE EDITOR                      XLINT
791  2   SET UP LIST OF VALID DIRECTIVES                              XLINT
792  2   SET UP LIST FLAG TO GET ENTIRE LIST                          XLINT
793  2   SET CURRENT ARGUMENT TO ZERO                                 XLINT
794  2   SET PROMPT MODE TO ALL                                       XLINT
795  2   SET UP INDEX INTO PARMS OF SHORT PROMPTS                     XLINT
796  2   SET NEW TABLE NAME TO ZERO                                   XLINT
797  2   INITIALIZE ARRAY OF TYPE LENGTHS                             XLINT
798  2   IF LITERAL LENGTH IS ZERO THEN                               XLINT
799  3     SET LITERAL POINTER TO FIRST AVAILABLE WORD                XLINT
800  2   ELSE                                                         XLINT
801  3     SET LITERAL POINTER TO FIRST WORD OF LITERALS              XLINT
802  3     CALL XEINT TO UNPACK LITERALS                              XLINT
803  2   ENDIF                                                        XLINT
804  2   CALL XINIX TO ACCEPT DEFAULT VALUES                          XLINT
805  2   CALL XERTN TO RETURN                                         XLINT
806  1 END XLINT                                                      XLINT
```

```
808   1 CD************                                                                                    XLMOD
809   1 CD0                                                                                              XLMOD
810   1 CD0       FORTRAN CALLING SEQUENCE:                                                              XLMOD
811   1 CD0                                                                                              XLMOD
812   1 CD0             CALL XELDS ('XLMOD')                                                             XLMOD
813   1 CD0                                                                                              XLMOD
814   1 CD************                                                                                   XLMOD
815   1 CD1                                                                                              XLMOD
816   1 CD1       XLMOD ALLOWS A USER TO MODIFY THE VERSION, SPECS, AND PROMPTS FOR                      XLMOD
817   1 CD1       A PROCESSOR AS WELL AS ADD/DELETE PARAMETERS FROM AN INTERFACE                         XLMOD
818   1 CD1       TABLE.                                                                                 XLMOD
819   1 CD1                                                                                              XLMOD
820   1 CD************                                                                                   XLMOD
821   1 CD5                                                                                              XLMOD
822   1 CD5       RTE ROUTINES USED:                                                                     XLMOD
823   1 CD5                                                                                              XLMOD
824   1 CD5             CLOSE, IAND, OPEN, POSNT, READF, WRITF                                           XLMOD
825   1 CD5                                                                                              XLMOD
826   1 CD5       FDS ROUTINES USED:                                                                     XLMOD
827   1 CD5                                                                                              XLMOD
828   1 CD5             XLIFL, XLINS, XLPFL, XLSPS, XRCPR, XREXT,                                        XLMOD
829   1 CD5             XRMOV, XRMSG, XRPCK, XRSET, XRUPK, XTCOM                                         XLMOD
830   1 CD5                                                                                              XLMOD
831   1 CD5       COMMON USED:                                                                           XLMOD
832   1 CD5                                                                                              XLMOD
833   1 CD5             EQUIVALENCE          (XE(3),   ISECU ),                                          XLMOD
834   1 CD5           + (XE(5),   MASSTA), (XE(6),   SUBSTA),                                          XLMOD
835   1 CD5           + (XE(7),   VALFLG), (XE(13),  PRMNAM),                                          XLMOD
836   1 CD5           + (XE(16),  PRGNAM), (XE(85),  EOSTOK),                                          XLMOD
837   1 CD5           + (XE(86),  INTTOK), (XE(89),  NAMTOK),                                          XLMOD
838   1 CD5           + (XE(92),  NEGTOK), (XE(142), ICR   ),                                          XLMOD
839   1 CD5           + (XE(145), COMBUF), (XB(26),  VERS  ),                                          XLMOD
840   1 CD5           + (XB(33),  SFLAG ), (XB(35),  CFLAG ),                                          XLMOD
841   1 CD5           + (XB(37),  ARGNO ), (XB(38),  IT    ),                                          XLMOD
842   1 CD5           + (XB(43),  LEN   ), (XB(44),  NEGFLG),                                          XLMOD
843   1 CD5           + (XB(88),  LITDSP), (XB(90),  LITPTR),                                          XLMOD
844   1 CD5           + (XB(92),  INDX  ), (XB(96),  NOPARM),                                          XLMOD
845   1 CD5           + (XB(101), NEGR  ), (XB(101), LIB01 ),                                          XLMOD
846   1 CD5           + (XB(104), LIB02 ), (XB(108), PARMS )                                           XLMOD
847   1 CD5                                                                                              XLMOD
848   1 CD************                                                                                   XLMOD
```

```
850   1 BEGIN XLMOD                                                                    XLMOD
851   2    IF VALFLAG SAYS THIS IS AN ORIGINAL REQUEST TO MODIFY THEN                  XLMOD
852   3       SET RTN = 1                                                              XLMOD

853   3       :PROM1:                                                                  XLMOD
854   3       CALL XLINS TO DISPLAY SPEC INSTRUCTIONS                                  XLMOD
855   3       INITIALIZE MASTER AND SUB STATE FLAGS                                    XLMOD
856   3       CALL XTCOM TO PROMPT FOR PROCESSOR NAME                                  XLMOD
857   2    EXIT XLMOD IF RETURN CODE SAYS % ENTERED                                    XLMOD
858   3       ERREXIT IF XTCOM RETURN CODE NON-ZERO OR                                 XLMOD
859   3       ERREXIT IF INVALID PROCESSOR NAME (NOT 6CHAR NAME) TO :PRMERR:           XLMOD
860   3       CALL OPEN, READF AND CLOSE TO READ IN LIBRARY DIRECTORY                  XLMOD
861   3       ERREXIT IF THERE IS A FILE ERROR TO :FILERR:                             XLMOD
862   3       ERREXIT IF PROCESSOR IS NOT IN LIBRARY DIRECTORY TO :PRMERR:             XLMOD
863   3       SAVE INTERFACE TABLE BIT AND VERSION NUMBER                              XLMOD
864   3       PERFORM VERSION TO UPDATE VERSION NUMBER                                 XLMOD
865   3       IF THE PROCESSOR HAD AN INTERFACE TABLE THEN                             XLMOD
866   4          SET NEW VERSION NUMBER IN INTERFACE TABLE                             XLMOD
867   4          CREATE DEFAULT INTERFACE TABLE NAME                                   XLMOD
868   4          CALL OPEN AND READF TO READ IN HEDR AND SPECS                         XLMOD
869   4          IF THERE ARE LITERALS THEN                                            XLMOD
870   5             CALL READF TO READ IN LITERALS                                     XLMOD
871   4          ENDIF                                                                 XLMOD
872   4          CALL CLOSE TO CLOSE FILE                                              XLMOD
873   4          ERREXIT IF THERE WAS A FILE ERROR TO :FILERR:                         XLMOD
874   4          CREATE PROMPT TABLE NAME                                              XLMOD
875   4          CALL OPEN, READF AND CLOSE TO READ IN SHORT PROMPTS                   XLMOD
876   4          ERREXIT IF THERE WAS A FILE ERROR TO :FILERR:                         XLMOD
877   4          CALL NAMF TO RENAME PROMPT FILE >XLTMP                                XLMOD
878   4          ERREXIT IF NAMF ERROR TO :FILERR:                                     XLMOD
879   4          SET CODES ARRAY TO MODIFY-ABSTRACT AND NO CHANGES TO PARAMETER SPECS   XLMOD
880   4          PERFORM DELPRM TO DELETE PARAMETERS                                   XLMOD
881   4          PERFORM MODPRM TO MODIFY PARAMETERS                                   XLMOD
882   4          PERFORM ADDPRM TO ADD PARAMETERS                                      XLMOD
883   4          CALL XLPFL TO CREATE NEW PROMPT FILE                                  XLMOD
884   4          PERFORM DEFALT TO ADD/MODIFY/DELETE ANY DEFAULT VALUES                XLMOD
885   3       ELSE                                                                     XLMOD
886   4          CALL NAMF TO RENAME PROMPT FILE >XLTMP                                XLMOD
887   4          ERREXIT IF NAMF ERROR TO :FILERR:                                     XLMOD
888   4          SET CODES ARRAY TO MODIFY ABSTRACT ONLY                               XLMOD
889   4          CALL XLPFL TO CREAT NEW PROMPT FILE                                   XLMOD
890   4          PERFORM XLMOD - NO RETURN EXPECTED                                    XLMOD
891   3       ENDIF                                                                    XLMOD
892   2    ENDIF                                                                       XLMOD
893   2    CALL PURGE TO PURGE OLD DEFAULT INTERFACE TABLE FILE                        XLMOD
894   2    ERREXIT IF FILE ERROR TO :FILERR:                                           XLMOD
895   2    CALL XLIFL TO CREATE NEW DEFAULT INTERFACE TABLE FILE                       XLMOD
896   2    SET VALFLG TO SAY ORIGINAL REQUEST TO MODIFY                                XLMOD
897   2    PERFORM XLMOD - NO RETURN EXPECTED                                          XLMOD

898   2    :PRMERR:                                                                    XLMOD
899   2    CALL XRMSG TO DISPLAY ERROR MESSAGE                                         XLMOD
900   2    GO TO (:PROM1:,:PROM2:,:PROM3:,:PROM4:,:PROM5:,:PROM6:,:PROM7:),RTN         XLMOD

901   2    :FILERR:                                                                    XLMOD
902   2    CALL XRMSG TO DISPLAY FILE ERROR                                            XLMOD
903   1 END XLMOD                                                                      XLMOD
```

```
905    1 BEGIN VERSION                                                                    XLMOD
906    2    SET RTN = 2                                                                   XLMOD

907    2    :PROM2:                                                                       XLMOD
908    2    CALL XTCOM TO PROMPT USER FOR VERSION NUMBER                                  XLMOD
909    2    IF RETURN CODE IS NOT CR THEN                                                 XLMOD
910    3       ERREXIT IF RETURN CODE IS NON-ZERO OR                                      XLMOD
911    3       ERREXIT IF VERSION IS INVALID TO :PRMERR:                                  XLMOD
912    3       IF INPUT VERSION IS NOT EQUAL TO OLD VERSION THEN                          XLMOD
913    4          CALL XRSET TO PUT NEW VERSION IN ENTRY                                  XLMOD
914    4          CALL OPEN, WRITF, CLOSE TO UPDATE LIBRARY DIRECTORY                     XLMOD
915    4          ERREXIT IF FILE ERROR TO :FILERR:                                       XLMOD
916    3       ENDIF                                                                      XLMOD
917    2    ENDIF                                                                         XLMOD
918    1 END VERSION                                                                      XLMOD
919    1 *                                                                                XLMOD
920    1 *                                                                                XLMOD
921    1 *                                                                                XLMOD
922    1 BEGIN DELPRM                                                                     XLMOD
923    2    SET RTN = 3                                                                   XLMOD

924    2    :PROM3:                                                                       XLMOD
925    2    DO UNTIL RETURN CODE IS CR ENTERED                                            XLMOD
926    3       IF NUMBER OF PARAMETERS > 1 THEN                                           XLMOD
927    4          CALL XTCOM TO PROMPT FOR DELETE PARAMETER NAME                          XLMOD
928    4          IF RETURN CODE IS NOT CR ENTERED THEN                                   XLMOD
929    5             PERFORM RSPND TO INTERPRET RESPONSE                                  XLMOD
930    5             PERFORM CHDAT TO CHECK FOR EXISTING DATA                             XLMOD
931    5             SET ARGNO TH WORD IN CODES TO SAY 'DELETED'                          XLMOD
932    5             CALL XRMOV TO MOVE DATA TO DELETE PARAMETER                          XLMOD
933    5             DECREMENT # PARAMETERS BY 1                                          XLMOD
934    4          ENDIF                                                                   XLMOD
935    3       ELSE                                                                       XLMOD
936    4          CALL XRMSG TO DISPLAY NO PARAMETERS CAN BE DELETED                      XLMOD
937    3       EXIT DELPRM                                                                XLMOD
938    3       ENDIF                                                                      XLMOD
939    2    ENDDO                                                                         XLMOD
940    1 END DELPRM                                                                       XLMOD
941    1 *                                                                                XLMOD
942    1 *                                                                                XLMOD
943    1 *                                                                                XLMOD
944    1 BEGIN MODPRM                                                                     XLMOD
945    2    SET RTN = 4                                                                   XLMOD

946    2    :PROM4:                                                                       XLMOD
947    2    DO UNTIL RETURN CODE IS CR ENTERED                                            XLMOD
948    3       CALL XTCOM TO PROMPT FOR MODIFY PARAMETER NAME                             XLMOD
949    3       IF RETURN CODE IS NOT CR ENTERED THEN                                      XLMOD
950    4          PERFORM RSPND TO INTERPRET RESPONSE                                     XLMOD
951    4          PERFORM CHDAT TO CHECK FOR EXISTING DATA                                XLMOD
952    4          SET ARGNO TH NON-DELETED WORD IN CODES TO SAY 'MODIFIED'                XLMOD
953    4          CALL XLSPS TO PROMPT USER FOR SPECIFICATIONS                            XLMOD
954    4          SET IT COMPLETE BIT OFF                                                 XLMOD
955    3       ENDIF                                                                      XLMOD
956    2    ENDDO                                                                         XLMOD
957    1 END MODPRM                                                                       XLMOD
```

```
959   1 BEGIN ADDPRM                                                                                                    XLMOD
960   2    SET RTN = 5                                                                                                  XLMOD

961   2    :PROM5:                                                                                                      XLMOD
962   2    DO UNTIL RETURN CODE IS CR ENTERED                                                                           XLMOD
963   3       IF NUMBER OF PARAMETERS < 63 THEN                                                                         XLMOD
964   4          CALL XTCOM TO PAROMPT FOR ADD PARAMETER BEFORE/AFTER PARAMETER NAME                                    XLMOD
965   4          IF RETURN CODE IS NOT CR ENTERED THEN                                                                  XLMOD
966   5             PERFORM RSPND TO INTERPRET RESPONSE                                                                 XLMOD
967   5             SET ARGNO TH NON-DELETED WORD IN CODES TO SAY 'ADDED'                                               XLMOD
968   5             INCREMENT NUMBER OF PARAMETERS BY 1                                                                 XLMOD
969   5             CALL XRMOV TO MOVE DATA TO MAKE SPACE FOR NEW PARIMETER                                             XLMOD
970   5             CALL XLSPS TO GET NEW SPECS FOR THIS PARAMETER                                                      XLMOD
971   5             SET IT COMPLETE BIT OFF                                                                             XLMOD
972   4          ENDIF                                                                                                  XLMOD
973   3       ELSE                                                                                                      XLMOD
974   4          CALL XRMSG TO DISPLAY NO MORE PARAMETERS CAN BE ADDED                                                  XLMOD
975   3       EXIT ADDPRM                                                                                               XLMOD
976   3       ENDIF                                                                                                     XLMOD
977   2    ENDDO                                                                                                        XLMOD
978   1 END ADDPRM                                                                                                      XLMOD
979   1 *                                                                                                               XLMOD
980   1 *                                                                                                               XLMOD
981   1 *                                                                                                               XLMOD
982   1 BEGIN DEFALT                                                                                                    XLMOD
983   2    SET RTN = 7                                                                                                  XLMOD

984   2    :PROM7:                                                                                                      XLMOD
985   2    CALL XTCON TO PROMPT FOR ADD/MODIFY/DELETE DEFAULT VALUES                                                    XLMOD
986   2    CALL XTCOM TO PROMPT FOR ADD/MODIFY/DELETE DEFAULT VALUES                                                    XLMOD
987   2    ERREXIT IF RETURN CODE IS NON-ZERO TO :PRMERR:                                                               XLMOD
988   2    IF RESPONSE IS YES THEN                                                                                      XLMOD
989   3       SET VALFLAG TO SAY CALL INTERFACE TABLE EDITOR                                                            XLMOD
990   2    EXIT XLMOD                                                                                                   XLMOD
991   2    ENDIF                                                                                                        XLMOD
992   1 END DEFALT                                                                                                      XLMOD
```

```
 994    1 BEGIN RSPND                                                                    XLMOD
 995    2    ERREXIT IF RETURN CODE IS NON-ZERO TO :PRMERR:                              XLMOD
 996    2    SET NEGATIVE FLAG OFF                                                       XLMOD
 997    2    IF NEGATIVE IS REQUESTED THEN                                               XLMOD
 998    3       SET NEGATIVE FLAG ON                                                     XLMOD
 999    2    ENDIF                                                                       XLMOD
1000    2    ERREXIT IF PARAMETER NAME NOT IN INTERFACE TABLE TO :PRMERR:                XLMOD
1001    2    CONVERT PARAMETER NAME TO ARGUMENT NUMBER                                   XLMOD
1002    1 END RSPND                                                                      XLMOD
1003    1 *                                                                              XLMOD
1004    1 *                                                                              XLMOD
1005    1 *                                                                              XLMOD
1006    1 BEGIN CHDAT                                                                    XLMOD
1007    2    EXTRACT SFLAG, CFLAG AND DISP FROM SPEC ENTRY                               XLMOD
1008    2    IF DOUBLE SUBSCRIPT FLAG IS ON THEN                                         XLMOD
1009    3       SET LITERAL ENTRY LENGTH TO 2                                            XLMOD
1010    2    ELSE                                                                        XLMOD
1011    3       IF THERE IS COMPLETE LITERAL DATA THEN                                   XLMOD
1012    4          SET LITERAL ENTRY LENGTH TO SIZE                                      XLMOD
1013    3       ENDIF                                                                    XLMOD
1014    2    ENDIF                                                                       XLMOD
1015    1 END CHDAT                                                                      XLMOD
```

```
1017    1 CD*************                                                    XLMSG
1018    1 CDO                                                               XLMSG
1019    1 CDO       FORTRAN CALLING PROCEDURE                               XLMSG
1020    1 CDO                                                               XLMSG
1021    1 CDO    CALL XELDS ('XLMSG ')                                      XLMSG
1022    1 CDO                                                               XLMSG
1023    1 C*********                                                        XLMSG
1024    1 CD1                                                               XLMSG
1025    1 CD1       XLMSG PROVIDES MAINTENANCE OF THE FDS MESSAGE FILE  XRMSG XLMSG
1026    1 CD1                                                               XLMSG
1027    1 C*********                                                        XLMSG
1028    1 CD2       INPUT                                                   XLMSG
1029    1 CD2    XE COMMON - LU, ISEQU, FLAGS, TOKENS, ICR                  XLMSG
1030    1 CD2                                                               XLMSG
1031    1 CD2    TERMINAL  - CREATION MODE, AREA AND MAXIMUM NUMBER OF MESSAGES XLMSG
1032    1 CD2                UPDATE MODE, MESSAGE NUMBER AND TEXT           XLMSG
1033    1 CD2                                                               XLMSG
1034    1 CD2    MESSAGE FILE - DIRECTORY AND OLD TEXT                      XLMSG
1035    1 CD2                                                               XLMSG
1036    1 C*********                                                        XLMSG
1037    1 CD3       OUTPUT                                                  XLMSG
1038    1 CD3    XE COMMON - COMBUF                                         XLMSG
1039    1 CD3                                                               XLMSG
1040    1 CD3    MESSAGE FILE - DIRECTORY AND TEXT UPDATES                  XLMSG
1041    1 CD3                                                               XLMSG
1042    1 C*********                                                        XLMSG
1043    1 CD4       LOCAL                                                   XLMSG
1044    1 CD4    AREA   - NUMERICAL AREA INDICATOR FOR MESSAGE              XLMSG
1045    1 CD4    DIRECT - MESSAGE DIRECTORY (SEE SDD 6.2.4.12)              XLMSG
1046    1 CD4    I      - INDEX TO BEGINNING OF CURRENT DIRECTORY ENTRY     XLMSG
1047    1 CD4    IDCB   - FILE MANAGER DATA CONTROL BLOCK                   XLMSG
1048    1 CD4    IERR   - FILE MANAGER & XTCOM RETURN CODE                  XLMSG
1049    1 CD4    NUMB   - MESSAGE NUMBER WITHIN MESSAGE AREA                XLMSG
1050    1 CD4    NXTBLK - BLOCK NUMBER WITHIN FILE                          XLMSG
1051    1 CD4             CREATE MODE - NEXT BLOCK AVAILABLE FOR ALLOCATION XLMSG
1052    1 CD4             UPDATE MODE - NUMBER OF BLOCK CONTAINING MESSAGE  XLMSG
1053    1 CD4    RECPOS - MESSAGE LOCATION WITHIN 128 WORD BLOCK (1, 33, 65 OR 97) XLMSG
1054    1 CD4                                                               XLMSG
1055    1 C*********                                                        XLMSG
1056    1 CD5       NOTES                                                   XLMSG
1057    1 CD5    USES APOSN, CLOSE, CREAT, EXEC, IAND, KCVT, OPEN, READF, WRITF, XLMSG
1058    1 CD5         XERTN, XRI6, XRMOV, XRMSG, XTCOM, XUDBG               XLMSG
1059    1 CD5                                                               XLMSG
1060    1 CD5    WHEN REPLACING AN EXISTING MESSAGE, A NULL RESPONSE WILL LEAVE THE XLMSG
1061    1 CD5    EXISTING TEXT IN PLACE.                                    XLMSG
1062    1 CD5                                                               XLMSG
1063    1 CD5    MESSAGE UPDATING MAY BE TERMINATED AT ANY TIME BY ENTERING A X XLMSG
1064    1 CD5                                                               XLMSG
1065    1 C*********                                                        XLMSG
```

```
1067    1 BEGIN XLMSG                                                                                                    XLMSG
1068    2    OPEN  XRMSG                                                                                                 XLMSG
1069    2    IF FILE NOT FOUND                                                                                           XLMSG
1070    2    THEN                                                                                                        XLMSG
1071    3       OUTPUT 'MESSAGE FILE CREATION'                                                                           XLMSG
1072    3       DO FOR EACH OF THE 32 DIRECTORY ENTRIES                                                                  XLMSG
1073    4          PROMPT FOR AREA ID AND MAXIMUM NUMBER OF MESSAGES FOR THIS ENTRY NUMBER                               XLMSG
1074    4          IF RESPONSE WAS NULL                                                                                  XLMSG
1075    4          THEN                                                                                                  XLMSG
1076    5             CLEAR ENTRY                                                                                        XLMSG
1077    4          ELSE                                                                                                  XLMSG
1078    5             STORE ID                                                                                           XLMSG
1079    5             COMPUTE AREA ORIGIN AND STORE                                                                      XLMSG
1080    5             CLEAR LAST MESSAGE NUMBER                                                                          XLMSG
1081    5             COMPUTE NUMBER OF BLOCKS AND STORE                                                                 XLMSG
1082    4          ENDIF                                                                                                 XLMSG
1083    3       ENDDO                                                                                                    XLMSG
1084    3       CREATE A CLEARED FILE OF TOTAL REQUIRED SIZE                                                             XLMSG
1085    2    ELSE                                                                                                        XLMSG
1086    3       READ DIRECTORY                                                                                           XLMSG
1087    2    ENDIF                                                                                                       XLMSG
1088    2    DO UNTIL USER INPUTS X                                                                                      XLMSC
1089    3       PROMPT FOR MESSAGE NUMBER                                                                                XLMSG
1090    3       SEPARATE AREA AND MESSAGE NUMBER AND COMPUTE BLOCK NUMBER AND MESSAGE LOC                                 XLMSG
1091    3       IF VALID AREA AND BLOCK NUMBER <= NUMBER OF BLOCKS                                                       XLMSG
1092    3       THEN                                                                                                     XLMSG
1093    4          READ BLOCK                                                                                            XLMSG
1094    4          IF FIRST WORD OF MESSAGE IS NOT NULL (MESSAGE ALREADY EXIST)                                          XLMSG
1095    4          THEN                                                                                                  XLMSG
1096    5             DISPLAY OLD MESSAGE TEXT                                                                           XLMSG
1097    4          ENDIF                                                                                                 XLMSG
1098    4          PROMPT FOR TEXT                                                                                       XLMSG
1099    4          IF NON-NULL RESPONSE                                                                                  XLMSG
1100    4          THEN                                                                                                  XLMSG
1101    5             STORE TEXT IN BLOCK                                                                                XLMSG
1102    5             REWRITE BLOCK                                                                                      XLMSG
1103    5             IF MESSAGE NUMBER > LAST MESSAGE NUMBER                                                            XLMSG
1104    5             THEN                                                                                               XLMSG
1105    6                REPLACE LAST MESSAGE NUMBER WITH NEW NUMBER                                                     XLMSG
1106    5             ENDIF                                                                                              XLMSG
1107    4          ENDIF                                                                                                 XLMSG
1108    3       ELSE                                                                                                     XLMSG
1109    4          OUTPUT 'XL29 AREA INVALID OR NUMBER TOO LARGE'                                                        XLMSG
1110    3       ENDIF                                                                                                    XLMSG
1111    2    ENDDO                                                                                                       XLMSG
1112    2    REWRITE DIRECTORY BLOCK                                                                                     XLMSG
1113    1 END XLMSG                                                                                                      XLMSG
```

```
1115    1 CD************                                                        XLPCR
1116    1 CDO                                                                 XLPCR
1117    1 CDO       FORTRAN CALLING SEQUENCE:                                 XLPCR
1118    1 CDO                                                                 XLPCR
1119    1 CDO             CALL XLPCR                                          XLPCR
1120    1 CDO                                                                 XLPCR
1121    1 CD************                                                        XLPCR
1122    1 CD1                                                                 XLPCR
1123    1 CD1       XLPCR CREATES AND INITIALIZES THE PDB LOG FILE            XLPCR
1124    1 CD1                                                                 XLPCR
1125    1 CD************                                                        XLPCR
1126    1 CD2                                                                 XLPCR
1127    1 CD2       INPUTS FROM COMMON:                                       XLPCR
1128    1 CD2                                                                 XLPCR
1129    1 CD2             XE(3) ISECU,   XE(142) ICR                          XLPCR
1130    1 CD2                                                                 XLPCR
1131    1 CD************                                                        XLPCR
1132    1 CD5                                                                 XLPCR
1133    1 CD5       RTE ROUTINES USED:                                        XLPCR
1134    1 CD5                                                                 XLPCR
1135    1 CD5             CLOSE, CREAT, WRITF                                 XLPCR
1136    1 CD5                                                                 XLPCR
1137    1 CD5       FDS ROUTINES USED:                                        XLPCR
1138    1 CD5             XRMOV, XRMSG, XTCOM                                 XLPCR
1139    1 CD5                                                                 XLPCR
1140    1 CD5       COMMON USED:                                              XLPCR
1141    1 CD5                                                                 XLPCR
1142    1 CD5             EQUIVALENCE          (XE(3),  ISECU ),              XLPCR
1143    1 CD5            +(XE(85), TOKENS),    (XE(142),ICR   ),              XLPCR
1144    1 CD5            +(XE(145),COMBUF),    (XB(99), IERR  ),              XLPCR
1145    1 CD5            +(XB(100),IDCB  ),    (XB(116),IBUF  ),              XLPCR
1146    1 CD5                                                                 XLPCR
1147    1 CD************                                                        XLPCR
```

```
1149    1 BEGIN XL.CR                                                                   XLPCR
1150    2    CALL CREAT TO CREATE MDB/PDB LOG FILE                                      XLPCR
1151    2    ERREXIT IF CREATE ERROR TO :FILERR:                                        XLPCR
1152    2    INITIALIZE LOG RECORD BUFFER TO ZEROS                                      XLPCR
1153    2    SET # MDB FILES CURRENTLY USED TO ZERO                                     XLPCR
1154    2    SET MAXIMUM NUMBER MDBS TO 20                                              XLPCR
1155    2    CALL WRITE TO WRITE MDB RECORD TO LOG FILE                                 XLPCR
1156    2    ERREXIT IF WRITE ERROR TO :FILERR:                                         XLPCR
1157    2    DO FOR EACH REMAINING LOG RECORD                                           XLPCR
1158    3       DO FOR THIS PAIR OF USER ID'S                                           XLPCR
1159    4          CALL XTCOM TO PROMPT FOR MAXIMUM ALLOWED # PDB'S                     XLPCR
1160    4          SET MAXIMUM # PDB FILES TO RESPONSE                                  XLPCR
1161    4          SET # PDB FILES CURRENTLY USED TO ZERO                               XLPCR
1162    3       ENDDO                                                                   XLPCR
1163    3       CALL WRITF TO WRITE 1 PDB RECORD TO LOG FILE                            XLPCR
1164    3       ERREXIT IF WRITF ERROR TO :FILERR:                                      XLPCR
1165    2    ENDDO                                                                      XLPCR
1166    2    CALL CLOSE TO CLOSE FILE                                                   XLPCR
1167    2    ERREXIT IF CLOSE ERROR TO :FILERR:                                         XLPCR
1168    1 EXIT XLPCR                                                                    XLPCR

1169    2    :FILERR:                                                                   XLPCR
1170    2    CALL XRMSG TO DISPLAY FILE ERROR                                           XLPCR
1171    2    CALL CLOSE TO CLOSE FILE                                                   XLPCR
1172    1 END XLPCR                                                                     XLPCR
```

```
1174    1 CD**************                                                                  XLPFL
1175    1 CD0                                                                               XLPFL
1176    1 CD0       FORTRAN CALLING SEQUENCE:                                               XLPFL
1177    1 CD0                                                                               XLPFL
1178    1 CD0             CALL XLPFL  (NOCOD, CODES)                                        XLPFL
1179    1 CD0                                                                               XLPFL
1180    1 CD**************                                                                  XLPFL
1181    1 CD1                                                                               XLPFL
1182    1 CD1       XLPFL CREATES PROMPT FILE FOR PROCESSORS. IT ALSO RE-CREATES            XLPFL
1183    1 CD1       IT WHEN A PROCESSOR IS MODIFIED.                                        XLPFL
1184    1 CD1                                                                               XLPFL
1185    1 CD**************                                                                  XLPFL
1186    1 CD2                                                                               XLPFL
1187    1 CD2       INPUTS IN CALLING SEQUENCE:                                             XLPFL
1188    1 CD2                                                                               XLPFL
1189    1 CD2             NOCOD - NUMBER OF CODES IN THE CODE ARRAY                         XLPFL
1190    1 CD2             CODES - ARRAY OF CODES THAT REPRESENT:                            XLPFL
1191    1 CD2                         CODES(1)  PROCESSOR ABSTRACT                        XLPFL
1192    1 CD2                         CODES(2) THRU CODES(NOCOD) PARAMETERS 1 THRU N      XLPFL
1193    1 CD2                       WITH VALUES OF:                                       XLPFL
1194    1 CD2                         0 - NO CHANGE                                       XLPFL
1195    1 CD2                         1 - MODIFIED                                        XLPFL
1196    1 CD2                         2 - DELETED                                         XLPFL
1197    1 CD2                         3 - ADDED                                           XLPFL
1198    1 CD2                                                                             XLPFL
1199    1 CD2       INPUTS IN COMMON:                                                     XLPFL
1200    1 CD2                                                                             XLPFL
1201    1 CD2             XE(1) LU,     XE(3) ISECU,  XE(16) PRCNAM,                      XLPFL
1202    1 CD2             XE(142) ICR,  XB(96)NOPARM, XB(108)PARMS                        XLPFL
1203    1 CD2                                                                             XLPFL
1204    1 CD**************                                                                XLPFL
1205    1 CD5                                                                             XLPFL
1206    1 CD5       RTE FUNCTIONS USED:                                                   XLPFL
1207    1 CD5                                                                             XLPFL
1208    1 CD5             CLOSE, CREAT, EXEC,  OPEN,                                      XLPFL
1209    1 CD5             POSNT, PURGE, READF, WRITF                                      XLPFL
1210    1 CD5                                                                             XLPFL
1211    1 CD5       FDS FUNCTIONS USED:                                                   XLPFL
1212    1 CD5                                                                             XLPFL
1213    1 CD5             XERTN, XRCPR, XPMOV, XRMSG,                                     XLPFL
1214    1 CD5             XRPCK, XRUPK, XTCOM                                             XLPFL
1215    1 CD5                                                                             XLPFL
1216    1 CD5       COMMON USED:                                                          XLPFL
1217    1 CD5                                                                             XLPFL
1218    1 CD5             EQUIVALENCE         (XE(1),   LU    ),                        XLPFL
1219    1 CD5           + (XE(3),   ISECU ), (XE(7),   VALFLG),                         XLPFL
1220    1 CD5           + (XE(16),  PRCNAM), (XE(85),  TOKENS),                         XLPFL
1221    1 CD5           + (XE(142), ICR   ), (XE(144), COMPTR),                         XLPFL
1222    1 CD5           + (XE(145), COMBUF), (XB(96),  NOPARM),                         XLPFL
1223    1 CD5           + (XB(108),PARMS )                                              XLPFL
1224    1 CD5                                                                           XLPFL
1225    1 CD**************                                                              XLPFL
```

```
1227    1 BEGIN XLPFL                                                                                                   XLPFL
1228    2    COMPUTE SIZE OF FILE AS # PARAMETERS +3                                                                    XLPFL
1229    2    CALL CREAT TO CREATE PROMPT FILE                                                                           XLPFL
1230    2    ERREXIT IF CREAT ERROR TO :FILERR:                                                                         XLPFL
1231    2    STUFF SYNTAX RECORD # AND # PARAMETERS INTO LIST OF SHORT PROMPTS                                          XLPFL
1232    2    CALL WRITF TO WRITE SHORT PROMPT RECORD(S)                                                                 XLPFL
1233    2    ERREXIT IF WRITF ERROR TO :FILERR:                                                                         XLPFL
1234    2    IF ABSTRACT CODE IS MODIFY THEN                                                                            XLPFL
1235    3       CALL OPEN TO OPEN OLD PROMPT FILE >XLTMP                                                                XLPFL
1236    3       ERREXIT IF OPEN ERROR TO :FILERR:                                                                       XLPFL
1237    3       CALL READF TO READ EXISTING ABSTRACT                                                                    XLPFL
1238    3       ERREXIT IF READF ERROR TO :FILERR:                                                                      XLPFL
1239    3       CALL EXEC TO DISPLAY EXISTING ABSTRACT                                                                  XLPFL
1240    3       CALL XTCOM TO PROMPT USER TO MODIFY ABSTRACT                                                            XLPFL
1241    3       IF RESPONSE IS CR (NO MODIFICATION) THEN                                                                XLPFL
1242    4          CALL WRITF TO WRITE EXISTING ABSTRACT TO NEW FILE                                                    XLPFL
1243    4          ERREXIT IF WRITF ERROR TO :FILERR:                                                                   XLPFL
1244    3       ELSE   (NEW ABSTRACT WAS ENTERED)                                                                       XLPFL
1245    4          PERFORM FORMAT TO FORMAT RESPONSE INTO FILE                                                          XLPFL
1246    3       ENDIF                                                                                                   XLPFL
1247    2    ELSE  (ABSTRACT CODE IS ADD)                                                                               XLPFL
1248    3       CALL XTCOM TO PROMPT USER TO ENTER NEW ABSTRACT                                                         XLPFL
1249    3       PERFORM FORMAT TO FORMAT RESPONSE INTO FILE                                                             XLPFL
1250    2    ENDIF                                                                                                      XLPFL
1251    2    IF #PARAMETERS IS NOT ZERO THEN                                                                            XLPFL
1252    3       PERFORM EXTPRM TO GET DEFINITIONS FOR EACH PARAMETER                                                    XLPFL
1253    2    ENDIF                                                                                                      XLPFL
1254    2    CALL CLOSE TO CLOSE PROMPT FILE                                                                            XLPFL
1255    2    ERREXIT IF CLOSE ERROR TO :FILERR:                                                                         XLPFL
1256    2    IF THERE WAS AN OLD FILE THEN                                                                              XLPFL
1257    3       CALL CLOSE TO CLOSE OLD FILE >XLTMP                                                                     XLPFL
1258    3       CALL PURGE TO PURGE OLD FILE >XLTMP                                                                     XLPFL
1259    2    ENDIF                                                                                                      XLPFL
1260    1 EXIT XLPFL                                                                                                    XLPFL

1261    2    :FILERR:                                                                                                   XLPFL
1262    2    CALL XRMSG TO DISPLAY ERROR CODE                                                                           XLPFL
1263    2    CALL CLOSE TO CLOSE NEW FILE                                                                               XLPFL
1264    2    CALL CLOSE TO CLOSE OLD FILE >XLTMP                                                                        XLPFL
1265    2    CALL PURGE TO PURGE OLD FILE >XLTMP                                                                        XLPFL
1266    2    SET VALFLG =0                                                                                              XLPFL
1267    2    CALL XERTN TO RETURN TO MAIN  ***NO RETURN TO HERE***                                                      XLPFL
1268    1 END XLPFL                                                                                                     XLPFL
```

```
1270    1 BEGIN FORMAT                                                                  XLPFL
1271    2    INITIALIZE TOKEN POINTER AND  TOTAL WORD COUNT                             XLPFL
1272    2    DO UNTIL EOS IS DETECTED IN RESPONSE                                       XLPFL
1273    3       ERREXIT IF RESPONSE IS NOT CHARACTER STRING TO :ERR1:                   XLPFL
1274    3       ERREXIT IF RESPONSE IS TOO LONG (>128 WORDS) TO :ERR1:                  XLPFL
1275    3       CALL XRMOV TO MOVE RESPONSE TO BUFFER                                   XLPFL
1276    3       SET CONTROL CHARACTERS IN BUFFER                                        XLPFL
1277    3       INCREMENT TOTAL WORD COUNT BY THIS RESPONSE                             XLPFL
1278    3       ERREXIT IF NEXT RESPONSE IS NOT A COMMA TO :ERR1:                       XLPFL
1279    3       INCREMENT TOKEN POINTER TO NEXT CHARACTER STRING                        XLPFL
1280    2    ENDDO                                                                      XLPFL
1281    2    SET REMAINDER OF BUFFER TO NULL                                            XLPFL
1282    2    CALL WRITF TO WRITE NEW RESPONSE TO PROMPT FILE                            XLPFL
1283    2    ERREXIT IF WRITF ERROR TO :FILERR:                                         XLPFL
1284    1 END FORMAT                                                                    XLPFL

1285    1 :ERR1:                                                                        XLPFL
1286    1 CALL XRMSG TO DISPLAY INVALID RESPONSE                                        XLPFL
1287    1 RETURN TO REPROMPT USER FOR ANOTHER RESPONSE                                  XLPFL
```

```
1289    1 BEGIN EXTPRM                                                                                  XLPFL
1290    2    DO FOR EACH ENTRY IN CODES ARRAY                                                           XLPFL
1291    3       BEGIN CASE  (:NOCHNG:,:MOD:,:DEL:,:ADD:), CODES+1                                       XLPFL

1292    4          :NOCHNG:   (SPECS HAD NO CHANGES)                                                    XLPFL
1293    4          CALL READF TO READ EXISTING DEFINITION FROM >XLTMP                                   XLPFL
1294    4          ERREXIT IF READF ERROR TO :FILERR:                                                   XLPFL
1295    4          CALL WRITF TO WRITE EXISTING DEFINITION TO NEW FILE                                  XLPFL
1296    4          ERREXIT IF WRITF ERROR TO :FILERR:                                                   XLPFL

1297    4          :MOD:      (SPECS WERE MODIFIED)                                                     XLPFL
1298    4          CALL READF TO READ EXISTING DEFINITION FROM >XLTMP                                   XLPFL
1299    4          ERREXIT IF READF ERROR TO :FILERR:                                                   XLPFL
1300    4          CALL EXEC TO DISPLAY EXISTING DEFINITION                                             XLPFL
1301    4          CALL XTCOM TO PROMPT USER TO MODIFY DEFINITION                                       XLPFL
1302    4          IF RESPONSE IS CR (NO RESPONSE) THEN                                                 XLPFL
1303    5             CALL WRITF TO WRITE EXISTING DEFINITION TO NEW FILE                               XLPFL
1304    5             ERREXIT IF WRITF ERROR TO :FILERR:                                                XLPFL
1305    4          ELSE   (NEW DEFINITION WAS ENTERED)                                                  XLPFL
1306    5             PERFORM FORMAT TO FORMAT DEFINITION INTO FILE                                     XLPFL
1307    4          ENDIF                                                                                XLPFL

1308    4          :DEL:      (SPECS WERE DELETED)                                                      XLPFL
1309    4          CALL POSNT TO POSITION >XLTMP OVER THIS ENTRY                                        XLPFL
1310    4          ERREXIT IF POSNT ERROR TO :FILERR:                                                   XLPFL

1311    4          :ADD:      (SPECS WERE ADDED)                                                        XLPFL
1312    4          CALL XTCOM TO PROMPT USER TO ENTER NEW DEFINITION                                    XLPFL
1313    4          PERFORM FORMAT TO FORMAT RESPONSE INTO FILE                                          XLPFL
1314    3       END CASE                                                                                XLPFL
1315    3       DO UNTIL RESPONSE IS CR (EXIT)                                                          XLPFL
1316    4          CALL XTCOM TO PROMPT USER FOR SHORT PROMPT TO MODIFY DEFINITION                      XLPFL
1317    3       EXIT EXTPRM IF RESPONSE IS CR                                                           XLPFL
1318    4          SEARCH SHOPT PROMPTS FOR RESPONSE                                                    XLPFL
1319    4          ERREXIT IF NOT FOUND TO :ERR1:                                                       XLPFL
1320    4          CALL READF TO READ EXISTING DEFINITION FROM NEW FILE                                 XLPFL
1321    4          ERREXIT IF READF ERROR TO :FILERR:                                                   XLPFL
1322    4          CALL EXEC TO DISPLAY EXISTING DEFINITION                                             XLPFL
1323    4          CALL XTCOM TO PROMPT USER TO MODIFY DEFINITION                                       XLPFL
1324    4          IF RESPONSE IS POSITIVE                                                              XLPFL
1325    5             CALL POSNT TO REPOSITION NEW FILE BACK TO PREVIOUS RECORD                         XLPFL
1326    5             EPREXIT IF POSNT ERROR TO :FILERR                                                 XLPFL
1327    5             PERFORM FORMAT TO FORMAT NEW DEFINITION INTO FILE                                 XLPFL
1328    4          ENDIF                                                                                XLPFL
1329    3       ENDDO                                                                                   XLPFL
1330    2    ENDDO                                                                                      XLPFL
1331    1 END EXTPRM                                                                                    XLPFL
```

```
1 CD*************                                                  XLPMO
1 CD0                                                              XLPMO
1 CD0      FORTRAN CALLING SEQUENCE:                               XLPMO
1 CD0                                                              XLPMO
1 CD0           CALL XLPMO                                         XLPMO
1 CD0                                                              XLPMO
1 CD*************                                                  XLPMO
1 CD1                                                              XLPMO
1 CD1      XLPMO MODIFIES THE EXISTING PDB LOG FILE                XLPMO
1 CD1                                                              XLPMO
1 CD*************                                                  XLPMO
1 CD2                                                              XLPMO
1 CD2      INPUTS FROM COMMON:                                     XLPMO
1 CD2                                                              XLPMO
1 CD2           XE(1) LU,      XE(3) QUAL,     XE(8) SECU,         XLPMO
1 CD2           XE(142) ICR,   XB(99)IERR,     XB(100) IDCB,       XLPMO
1 CD2                                                              XLPMO
1 CD*************                                                  XLPMO
1 CD4                                                              XLPMO
1 CD4      INTERNAL VARIABLES IN COMMON:                           XLPMO
1 CD4                                                              XLPMO
1 CD4           XB(50) UNO   - NUMBER OF USER ID A-Z (1-26)        XLPMO
1 CD4           XB(51) RECNO- RECORD # WHERE UNO'S PDB LIST IS (2-14)  XLPMO
1 CD4           XB(52) FNAM - FILE NAME OF USER'S FDB (FULL 6 CHARS)   XLPMO
1 CD4           XB(55) DIFF - NUMBER OF FILES THAT MUST BE DELETETED   XLPMO
1 CD4                         WHEN NEW MAY < CURRENTLY IN USE      XLPMO
1 CD4           XB(244)OUTBUF-AREA WHERE DISPLAY OF CURRENT FILES IS BUILT.  XLPMO
1 CD4                                                              XLPMO
1 CD*************                                                  XLPMO
1 CD5                                                              XLPMO
1 CD5      RTE ROUTINES USED:                                      XLPMO
1 CD5                                                              XLPMO
1 CD5           CLOSE, EXEC, KCVT, PURGE, READF, WRITF             XLPMO
1 CD5                                                              XLPMO
1 CD5      FDS ROUTINES USED:                                      XLPMO
1 CD5                                                              XLPMO
1 CD5           XRCPR, XREXT, XRMOV, XRMSG,                        XLPMO
1 CD5           XRQFN, XRSFR, XTCOM                                XLPMO
1 CD5                                                              XLPMO
1 CD5      COMMON USED:                                            XLPMO
1 CD5                                                              XLPMO
1 CD5           EQUIVALENCE           (XE(1),  LU     ),           XLPMO
1 CD5          +(XE(3),   QUAL   ),   (XE(8),   SECU  ),           XLPMO
1 CD5          +(XE(85), TOKENS),     (XE(142),ICR    ),           XLPMO
1 CD5          +(XE(144),COMPTR),     (XE(145),COMBUF),            XLPMO
1 CD5          +(XB(50), UNO    ),    (XB(51), RECNO ),            XLPMO
1 CD5          +(XB(52), FNAM   ),    (XB(55), DIFF  ),            XLPMO
1 CD5          +(XB(56), RTN    ),    (XB(57), MSG   ),            XLPMO
1 CD5          +(XB(99), IERR   ),    (XB(100),IDCB  ),            XLPMO
1 CD5          +(XB(116),IBUF   ),    (XB(244),OUTBUF),            XLPMO
1 CD5          +(XB(304),IDCB2 )                                   XLPMO
1 CD5                                                              XLPMO
1 CD*************                                                  XLPMO
```

```
1387    1 BEGIN XLPMO                                                                                                   XLPMO
1388    2    DO UNTIL USER RESPONDS EXIT (%)                                                                            XLPMO
1389    3       CALL XTCOM TO PROMPT USER FOR USER ID                                                                   XLPMO
1390    3       ERREXIT IF RESPONSE IS INVALID TO :ERR1:                                                                XLPMO
1391    3       SAVE USER ID AND NUMBER (1-26) IN COMMON                                                                XLPMO
1392    3       CALL READF TO READ IN THE RECORD CONTAINING THIS ID                                                     XLPMO
1393    3       ERREXIT IF READF ERROR TO :FILERR:                                                                      XLPMO
1394    2       DISPLAY MAX # ALLOWED FOR THIS USER AND # CURRENTLY USED                                                XLPMO
1395    3       CALL XTCOM TO PROMPT USER FOR NEW MAXIMUM                                                               XLPMO
1396    3       IF RESPONSE IS A MODIFICATION (NOT CR) THEN                                                             XLPMO
1397    4          ERREXIT IF RESPONSE IS NOT VALID TO :ERR1:                                                           XLPMO
1398    4          COMPUTE DIFFERENCE AS #CURRENTLY USED - NEW MAXIMUM                                                  XLPMO
1399    4          IF DIFFERENCE <= 0 THEN                                                                              XLPMO
1400    5             CALL WRITF TO REWRITE UPDATED RECORD TO FILE                                                      XLPMO
1401    5             ERREXIT IF WRITF ERROR TO :FILERR:                                                                XLPMO
1402    4          ELSE                                                                                                 XLPMO
1403    5             DISPLAY LIST OF CURRENT FILES                                                                     XLPMO
1404    5             SET #PURGED = 0                                                                                   XLPMO
1405    5             DO UNTIL DIFFERENCE <= 0 OR RESPONSE IS EXIT (PERCENT)                                            XLPMO
1406    6                CALL XTCOM TO PROMPT USER TO DELETE #OVER FILES                                                XLPMO
1407    6                DO FOR EACH FILE NAME IN RESPONSE                                                              XLPMO
1408    7                   SEARCH LIST FOR FILE NAME                                                                   XLPMO
1409    7                   ERREXIT IF INVALID NAME TO :ERR1:                                                           XLPMO
1410    7                   MOVE FILE NAME TO PURGE LIST                                                                XLPMO
1411    7                   INCREMENT #PURGED BY 1                                                                      XLPMO
1412    7                   DECREMENT DIFFERENCE BY 1                                                                   XLPMO
1413    7                   COMPRESS OLD FILE NAME OUT OF LIST                                                          XLPMO
1414    6                ENDDO                                                                                          XLPMO
1415    5             ENDDO                                                                                             XLPMO
1416    5             IF RESPONSE WAS NOT EXIT THEN                                                                     XLPMO
1417    6                CALL WRITF TO REWRITE UPDATED LOG RECORD                                                       XLPMO
1418    6                ERREXIT IF WRITF ERROR TO :FILERR:                                                             XLPMO
1419    6                DO FOR #PURGED FILES                                                                           XLPMO
1420    7                   CALL XRQFN TO CREATE FILE NAME                                                              XLPMO
1421    7                   CALL PURGE TO PURGE FILE                                                                    XLPMO
1422    6                ENDDO                                                                                          XLPMO
1423    5             ENDIF                                                                                             XLPMO
1424    4          ENDIF                                                                                                XLPMO
1425    3       ENDIF                                                                                                   XLPMO
1426    2    ENDDO                                                                                                      XLPMO
1427    2    CALL CLOSE TO CLOSE MDB/PDB LOG FILE                                                                       XLPMO
1428    1 EXIT XLPMO                                                                                                    XLPMO

1429    2    :ERR1:                                                                                                     XLPMO
1430    2    CALL XRMSG TO DISPLAY ERROR IN RESPONSE                                                                    XLPMO
1431    2    RETURN TO REISSUE LAST PROMPT                                                                              XLPMO

1432    2    :FILERR:                                                                                                   XLPMO
1433    2    CALL XRMSG TO DISPLAY FILE ERROR                                                                           XLPMO
1434    2    CALL CLOSE TO CLOSE MDB/PDB FILE                                                                           XLPMO
1435    1 END XLPMO                                                                                                     XLPMO
```

```
1 CD************                                                        XLSPS
1 CD0                                                                   XLSPS
1 CD0      FORTRAN CALLING SEQUENCE FOR SPEC PROCESSOR:                 XLSPS
1 CD0                                                                   XLSPS
1 CD0            CALL XLSPS                                             XLSPS
1 CD0                                                                   XLSPS
1 CD************                                                        XLSPS
1 CD1                                                                   XLSPS
1 CD1      XLSPS PROCESSES INPUTS FOR ONE PARAMETER.                    XLSPS
1 CD1                                                                   XLSPS
1 CD************                                                        XLSPS
1 CD4                                                                   XLSPS
1 CD4      INTERNAL VARIABLES:                                          XLSPS
1 CD4                                                                   XLSPS
1 CD4            CLASS - (INTEGER, 1 WORD, XE(10)) IS THE PARAMETER CLASS XLSPS
1 CD4            IPRMT - (INTEGER, 1 WORD ) IS THE INDEX INTO PARMS WHERE XLSPS
1 CD4                    THE SHORT PROMPT IS TO BE STORED               XLSPS
1 CD4            ISPEC - (INTEGER, 1 WORD ) IS THE INDEX INTO PARMS WHERE XLSPS
1 CD4                    THE SPECS ARE TO BE STORED                     XLSPS
1 CD4              MSG - (INTEGER, 1 WORD) IS THE CURRENT ERROR CONDITION XLSPS
1 CD4             TYPE - (INTEGER, 1 WORD, XE(11)) IS THE PARAMETER TYPE XLSPS
1 CD4                                                                   XLSPS
1 CD************                                                        XLSPS
1 CD5                                                                   XLSPS
1 CD5      ROUTINES USED:                                               XLSPS
1 CD5                                                                   XLSPS
1 CD5            KCVT, XRCPR, XRMOV, XRMSG, XRSET, XTCOM                XLSPS
1 CD5                                                                   XLSPS
1 CD5      XE COMMON USED:                                              XLSPS
1 CD5                                                                   XLSPS
1 CD5            EQUIVALENCE (XE(8),   RETC  ), (XE(9),   RETN  ),     XLSPS
1 CD5           +            (XE(10).  CLASS ), (XE(11),  TYPE  ),     XLSPS
1 CD5           +            (XE(145), COMBUF), (XB(37),  ARGNO ),     XLSPS
1 CD5           +            (XB(96),  NOPARM), (XB(108), PARMS )      XLSPS
1 CD5                                                                   XLSPS
1 CD************                                                        XLSPS
```

```
1474   1 BEGIN XLSPS                                                                                                    XLSPS
1475   2    CALL KCVT TO CONVERT PARAMETER NUMBER TO ASKII                                                              XLSPS

1476   2    :PRMPT1:                                                                                                    XLSPS
1477   2    SET RETRN TO 1                                                                                              XLSPS
1478   2    CALL XTCOM TO PROMPT FOR PROMPT,CLASS,TYPE AND I/O FLAGS                                                    XLSPS
1479   2    ERREXIT IF RETURN CODE IS NOT ZERO TO :PRMERR:                                                              XLSPS
1480   2    INITIALIZE THIS SHORT PROMPT AND ENTRY                                                                      XLSPS
1481   2    ERREXIT IF SHORT PROMPT IS NOT VALID (6 CHAR NAME) TO :PRMERR:                                              XLSPS
1482   2    ERREXIT IF THIS IS A DUPLICATE SHORT PROMPT                                                                 XLSPS
1483   2    CALL XRMOV TO MOVE SHORT PROMPT INTO SPECS                                                                  XLSPS
1484   2    ERREXIT IF CLASS IS NOT VALID TO :PRMERR:                                                                   XLSPS
1485   2    CALL XRSET TO SET CLASS IN ENTRY                                                                            XLSPS
***    2    ERREXIT IF TYPE IS NOT VALID TO :PRMERR:                                                                    XLSPS
       2    ERREXIT IF CLASS IS DRDE AND TYPE IS SYMBOLIC STRING TO :PRMERR:                                            XLSPS
  88   2    CALL XRSET TO SET TYPE IN ENTRY                                                                             XLSPS
1489   2    ERREXIT IF I/O FLAGS ARE NOT VALID TO :PRMERR:                                                              XLSPS
1490   2    ERREXIT IF TYPE IS SYMBOLIC STRING AND I/O FLAGS ARE NOT INPUT TO :PRMERR:                                  XLSPS
1491   2    CALL XRSET TO SET I/O FLAGS IN ENTRY                                                                        XLSPS
1492   2    IF CLASS IS DATa ELEMENT THEN                                                                               XLSPS
1493   3       IF TYPE IS NOT SYMBOLIC STRING THEN                                                                      XLSPS
1494   4          SET RETRN TO 2

1495   4          :PRMPT2:                                                                                              XLSPS
1496   4          CALL XTCOM TO PROMPT FOR I AND J DIMENSIONS                                                           XLSPS
1497   4          ERREXIT IF RETURN CODE IS NOT NORMAL TO :PRMERR:                                                      XLSPS
1498   4          ERREXIT IF I DIMENSION IF NOT VALID TO :PRMERR:                                                       XLSPS
1499   4          IF J DIMENSION IS NOT ENTERED THEN                                                                    XLSPS
1500   5             SET J DIMENSION TO 1                                                                               XLSPS
1501   4          ELSE                                                                                                  XLSPS
1502   5             ERREXIT IF J DIMENSION IS NOT VALID TO :PRMERR:                                                    XLSPS
1503   5             IF J DIMENSION < 2 THEN                                                                            XLSPS
1504   6                SET J DIMENSION TO 1                                                                            XLSPS
1505   5             ELSE                                                                                               XLSPS
1506   6                STORE I DIMENSION IN ENTRY                                                                      XLSPS
1507   5             ENDIF                                                                                              XLSPS
1508   4          ENDIF                                                                                                 XLSPS
1509   4          SET SIZE =I DIMENSION * J DIMENSION * TYPE LENGTH                                                     XLSPS
1510   4          ERREXIT IF SIZE IS NOT VALID TO :PRMERR:                                                              XLSPS
1511   3       ELSE                                                                                                     XLSPS
1512   4          SET RETRN TO 3

1513   4                T3:                                                                                           XLSPS
1514   4          CA    XTCOM TO PROMPT FOR MAXIMUM SIZE                                                              XLSPS
1515   4          ERREXIT IF RETURN CODE IS NOT NORMAL TO :PRMERR:                                                      XLSPS
1516   4          ERREXIT IF MAXIMUM SIZE IS NOT VALID TO :PRMERR:                                                      XLSPS
1517   4          SET MAXIMUM SIZE INTO IDIM FIELD OF ENTRY                                                             XLSPS
1518   3       ENDIF                                                                                                    XLSPS
1519   2    ENDIF                                                                                                       XLSPS
1520   1 EXIT XLSPS

1521   2    :PRMERR:  CALL XRMSG TO DISPLAY ERROR MESSAGE                                                               XLSPS
1522   2    GO TO (:PRMPT1:,:PRMPT2:,:PRMPT3:), RETRN                                                                   XLSPS
1523   1 END XLSPS                                                                                                      XLSPS
```

```
SYMBOL DEFINITION TABLE
:ADD   :  1311
 ADDPRM    959
 CASE     1291
 CHDAT    1006
 DEFALT    982
:DEL   :  1308
 DELPRM    922
:ERR1  :  1429
:ERR1  :   463
:ERR1  :   199
:ERR1  :  1285
:ERR1  :   581
:ERR2  :   466
:ERR3  :   468
:EXIT  :    99
 EXTPRM   1289
:FILERR:   202
:FILERR:   584
:FILERR:  1261
:FILERR:   901
:FILERR:   658
:FILERR:  1432
:FILERR:  1169
:FILERR:   714
 FORMAT   1270
 LIBD     315
:MOD   :  1297
 MODPRM    944
:NOCHNG:  1292
:PRMERR:  1521
:PRMERR:   898
:PRMERR:   310
:PRMPT :   627
:PRMPT1:   328
:PRMPT1:  1476
:PRMPT2:   281
:PRMPT2:  1495
:PRMPT3:  1513
:PRMPT3:   293
:PRM1  :   161
:PROMPT:    53
:PROM1 :   853
:PROM2 :   907
:PROM3 :   924
:PROM4 :   946
:PROM5 :   961
:PROM7 :   984
 RSPND     994
 VERSIO    905
 XLCOB     410
:XLDBF  :   95
 XLDBF    537
:XLDEL  :   84
 XLDEL     626
 XLIFL     699
 XLINS     751
 XLINT     788
 XLMAN      55
```

```
 XLMOD     850
:XLMOD :    86
 XLMSG    1067
:XLMSG :    93
 XLPCR    1149
 XLPFL    1227
 XLPMO    1387
 XLPRM     160
:XLPRM :    74
 XLPRO     275
:XLPRO :    77
 XLSPS    1474

@XQT F.PDLIST
```

```
2    1 *D1   FDS MANAGER.  SCHEDULED BY FDS CONFIGURATION MANAGER.  ONE PROGRAM                XMGR
3    1 *D1   PER SIGNED ON USER.                                                              XMGR
4    1 *D1                                                                                    XMGR
5    1 **********                                                                             XMGR
6    1 *D2       INPUT                                                                        XMGR
7    1 *D2   FDS STATUS TABLE (SEE SDD 6.2.2)                                                 XMGR
8    1 *D2   FDS MANAGEMENT REQUESTS (SEE SDD 6.2.6.2)                                        XMGR
9    1 *D2                                                                                    XMGR
10   1 **********                                                                             XMGR
11   1 *D3       OUTPUT                                                                       XMGR
12   1 *D3   FDS MANAGER RESPONSES (SEE SDD 6.2.6.3)                                          XMGR
13   1 *D3                                                                                    XMGR
14   1 **********                                                                             XMGR
15   1 *D5       NOTES                                                                        XMGR
16   1 *D5   USES XMAMG, XMAWA, XMPAW, XMSCH, XMXQT, XUDMP                                    XMGR
17   1 *D5        RMPAR,$LIBX,$LIBR,RNRQ,CNUMD,$XEQ                                           XMGR
18   1 *D5                                                                                    XMGR
19   1 *D5   XMAWA IS A MANAGER GLOBAL DATA AREA CONTAINING THE AWA AND VARIOUS               XMGR
20   1 *D5   CONTROL VALUES                                                                   XMGR
21   1 *D5                                                                                    XMGR
22   1 **********                                                                             XMGR
23   1 *                                                                                      XMGR
24   1 *                                                                                      XMGR
25   1 *                                                                                      XMGR
26   1 *                                                                                      XMGR
27   1 BEGIN XMGR                                                                             XMGR
28   2 *    INPUTS: P1= LU                                                                    XMGR
29   2 *            P2= XVSTB ENTRY ADDRESS                                                   XMGR
30   2 *            P3= USER ID                                                               XMGR
31   2 *            P4= OPTIONS                                                               XMGR
32   2 *            P5= NUMBER OF DWA TRACKS                                                  XMGR
33   2    CALL  RMPAR                                                                         XMGR
34   2    LOCK THE MANAGER INTO THE PARTITION                                                 XMGR
35   2    SET XVSTA( FDS STATUS TABLE ENTRY ADDRESS)                                          XMGR
36   2    RESOLVE XVSTB ADDRESS                                                               XMGR
37   2    CALL XMDIN  INITIALIZE DWA FUNCTIONS                                                XMGR
38   2    IF ERROR RETURN THEN                                                                XMGR
39   2    EXIT TO :XMEND                                                                      XMGR
40   2    ENDIF                                                                               XMGR
41   2    GET EXEC'S CLASS NUMBER                                                             XMGR
42   2    GET PROCESSOR'S CLASS NUMBER                                                        XMGR
43   2    CALL  $LIBR     GET PRIVILEDGE                                                      XMGR
44   2    SET EXEC'S CLASS NUMBER IN STBEC                                                    XMGR
45   2    SET PROCESSOR'S CLASS NUMBER IN STBPC                                               XMGR
46   2    SET CURRENT TASK IN STBAT FROM EXEC ID STBEX                                        XMGR
47   2    CALL  $LIBX     ENABLE                                                              XMGR
48   2    FORM EXEC'S NAME FROM 'XEY' & ASCII LU(STBLA)                                       XMGR
49   2    SCHEDULE EXEC WITH WAIT                                                             XMGR
50   2    CALL  XMSCH                                                                         XMGR
51   2    DO UNTIL EXEC REQUESTS TERMINATION OR ABENDS(P1=9 OR -32768)                        XMGR
52   3 *       REQUEST IS IN P1                                                               XMGR
53   3       IF REQUEST IS FOR AWA MANAGEMENT (P1=1) THEN                                     XMGR
54   4          PERFORM XMAMG                                                                 XMGR
55   3       ELSE                                                                             XMGR
56   4          IF REQUEST IS FOR SEQUENCE TABLE EXECUTION (P1=2) THEN                        XMGR
57   5             PERFORM XMXQT                                                              XMGR
58   4          ELSE                                                                          XMGR
59   5             IF REQUEST NOT TERMINATION (P1=9 OR -32768) THEN                           XMGR
60   6                ISSUE MESSAGE '*** XM01 INVALID REQUEST (I1) FROM XEXNN'                XMGR
```

```
61    6 *                      II=REQUEST & NN IS LU                                XMGR
62    6 *  SET UP FOR FDS TERMINATION                                               XMGR
63    5              EXIT TO :XMEND                                                 XMGR
64    5              ENDIF                                                          XMGR
65    4           ENDIF                                                             XMGR
66    3        ENDIF                                                                XMGR
67    2     ENDDO                                                                   XMGR

68    2     :XMEND                                                                  XMGR
69    2     IF DUMP OPTION THEN                                                     XMGR
70    3        CALL  XUDMP TO DUMP OUR PARTITION                                    XMGR
71    2     ENDIF                                                                   XMGR
72    2     RELEASE DVA TRACK SPACE                                                 XMGR
73    2     FREE EXEC'S & PROCESSOR'S CLASS NUMBERS                                 XMGR
74    2     CALL RNRQ SET LOCK ON STATUS TABLE                                      XMGR
75    2     UNLOCK THE MANAGER FROM THE PARTITION                                   XMGR
76    2     CALL  $LIBR     DISABLE                                                 XMGR
77    2     CLEAR EXEC'S ID SEGMENT                                                 XMGR
78    2     CALCULATE EQT ADDRESS FOR ATTENTION ID                                  XMGR
79    2     CLEAR THE STATUS TABLE ENTRY                                            XMGR
80    2     CALL $LIBX       ENABLE                                                 XMGR
81    2     CALL RNRQ  CLEAR THE LOCK ON THE STATUS TABLE                           XMGR
82    2     ISSUE MESSAGE '***XM02 SIGN OFF FOR LU "NN" '                           XMGR
83    2     CALL $LIBR     GET PRIVLEDGED AND DISABLED                              XMGR
84    2     CALL $LIST   TO MAKE US DORMANT                                         XMGR
85    2     CLEAR OUR OWN ID                                                        XMGR
86    2     RESET EQT TO REMOVE ATTENTION ID                                        XMGR
87    1  EXIT :$XEQ    TO THE RTE DISPATCHER                                        XMGR
88    1  END      XMGR                                                             XMGR
```

```
 90    1 *D0      CALLING PROCEDURE                                                XMXQT
 91    1 *D0   JSB  XMXQT                                                          XMXQT
 92    1 *D0                                                                       XMXQT
 93    1 **********                                                                XMXQT
 94    1 *D1   SEQUENCE TABLE EXECUTION FROM &SEQTB                                XMXQT
 95    1 *D1                                                                       XMXQT
 96    1 **********                                                                XMXQT
 97    1 *D2      INPUT                                                            XMXQT
 98    1 *D2   XMAWA, XMPRM, XVSTA                                                 XMXQT
 99    1 *D2                                                                       XMXQT
100    1 **********                                                                XMXQT
101    1 *D3      OUTPUT                                                           XMXQT
102    1 *D3   XMCTP, XMPRM, XVSTA                                                 XMXQT
103    1 *D3                                                                       XMXQT
104    1 **********                                                                XMXQT
105    1 *D5      ROUTINES USED                                                    XMXQT
106    1 *D5   CNUMD, EXEC, XMAMG, XMKIL, XMPAW (XMSCH),                           XMXQT
107    1 *D5   XMSST, XMTFN, $LIBR, $LIBX                                          XMXQT
108    1 *D5                                                                       XMXQT
109    1 **********                                                                XMXQT
```

```
1 BEGIN XMXQT                                                                                    XMXQT
2    CALL XMTFN TO FIND &SEQTB TOC ENTRY                                                          XMXQT
2    FIND ADDRESS OF &SEQTB                                                                       XMXQT
2    CALL XMSST TO CONVERT ENDING SEQUENCE NUMBER INTO ENDING DISPLACEMENT                        XMXQT
2    CALL XMSST TO CONVERT STARTING SEQUENCE NUMBER INTO CURRENT DISPLACEMENT                     XMXQT
2    COMPUTE CURRENT ENTRY ADDRESS                                                                XMXQT
2    DO UNTIL THE LAST TABLE ENTRY IS EXECUTED OR                                                 XMXQT
3       UNTIL THE TERMINATION ENTRY IS EXECUTED                                                   XMXQT
3       IF PROCESSOR REQUIRES AN INTERFACE TABLE (WORD 3 BIT 8 IS SET) THEN                       XMXQT
3       EXIT TO :ERR08: IF INTERFACE TABLE NOT SPECIFIED (WORD 4 = 0) (PARMS = 1)                 XMXQT
4          CALL XMTFN TO SEARCH AWA FOR INTERFACE TABLE (CHAIN 4)                                 XMXQT
3       EXIT TO :ERR08: IF TABLE NOT FOUND (PARMS = 2)                                            XMXQT
4          IF TABLE NOT IN  AWA, THEN                                                             XMXQT
5             CALL XMDRT TO RETRIEVE FROM DWA                                                     XMXQT
4          EXIT TO :ERR08: IF SPACE IS NOT FOUND                                                  XMXQT
4          ENDIF                                                                                  XMXQT
3       EXIT TO :ERR08: IF INTERFACE TABLE NOT COMPLETE (WORD 3 BIT 8 CLEAR) (PARM5 = 3)          XMXQT
3       EXIT TO :ERR08: IF PROCESSOR NAME FIELDS DIFFERENT (BYTES 3-7) (PARM5 = 4)                XMXQT
3       EXIT TO :ERR08: IF VERSION FIELDS NOT EQUAL (WORD 3 BITS 9-15) (PARM5 = 5)                XMXQT
3       ENDIF                                                                                     XMXQT
3       SEARCH SEGMENT TABLE FOR PROCESSOR TO BE SCHEDULED                                        XMXQT
3       ERREXIT TO :ERR08: IF NOT FOUND                                                           XMXQT
3       CALL $LIBR TO BE PRIVLEDGED                                                               XMXQT
3       SET CURRENT TASK TO THIS PROCESSOR                                                        XMXQT
3       CALL $LIBX TO BE UN-PRIVLEDGED                                                            XMXQT
3       IF PROCESSOR REQUIRES AN INTERFACE TABLE THEN                                             XMXQT
4          CALCULATE LENGTH OF INTERFACE TABLE HEADER AND SPECS                                   XMXQT
4          CALL EXEC TO CLASS I/O WRITE HEADER AND SPECS                                          XMXQT
3       ENDIF                                                                                     XMXQT
3       CALL EXEC TO SCHEDULE PROCESSOR WITH WAIT                                                 XMXQT
3       CALL XMSCH TO RETRIEVE PARAMETERS FROM PROCESSOR                                          XMXQT
3       CALL XMTFN TO FIND &SEQTB TOC ENTRY                                                       XMXQT
3       FIND ADDRESS OF &SEQTB                                                                    XMXQT
3       COMPUTE CURRENT ENTRY ADDRESS                                                             XMXQT
3       DO WHILE PROCESSOR REQUESTS AWA MANAGEMENT (PARM1 = 1)                                    XMXQT
4          CALL XMAMG TO HONOR AWA REQUEST                                                        XMXQT
3       ENDDO                                                                                     XMXQT
3       CLEAR OUT CLASS BUFFERS FROM LAST PROCESSOR                                               XMXQT
3       ERREXIT TO :ERR08: IF PROCESSOR REQUESTED TERMINATION (PARM1 = 8)                         XMXQT
3       ERREXIT TO :ERR08: IF PROCESSOR ABENDED (PARM1 = -32768)                                  XMXQT
3       IF REQUEST IS TO RESET CURRENT SEQUENCE ENTRY (PARM1 = 3) THEN                            XMXQT
4          ERREXIT TO :ERR08: IF RESET NUMBER IS ZERO                                             XMXQT
4          CALL XMSST TO CONVERT SEQUENCE NUMBER INTO CURRENT ENTRY DISPLACEMENT                  XMXQT
4          ERREXIT TO :ERR08: IF RESET SEQUENCE NUMBER IS NOT FOUND                               XMXQT
4          IF TERMINAL ENTRY WAS JUST EXECUTED THEN                                               XMXQT
5             SET UP PARMS TO SHOW RESET SEQUENCE NUMBER                                          XMXQT
5             PERFORM :END: - **NO RETURN EXPECTED**                                              XMXQT
4          ENDIF                                                                                  XMXQT
3       ELSE                                                                                      XMXQT
4          IF REQUEST IS NOT NORMAL COMPLETION (PARM1 = 0) THEN                                   XMXQT
5             DISPLAY ERROR MESSAGE - INVALID REQUEST                                             XMXQT
5             ERREXIT TO :ERR08: IF CURRENT AT IS DORMANT                                         XMXQT
5             CALL XMKIL TO SET PARAMETERS TO ABEND ASSOCIATED TASK                               XMXQT
5             CALL XMPAW TO RESCHEDULE PROCESSOR                                                  XMXQT
5             ERREXIT TO :ERR08: TO TERMINATE SEQUENCE                                            XMXQT
4          ENDIF                                                                                  XMXQT
3       ENDIF                                                                                     XMXQT
3       INCREMENT CURRENT DISPLACEMENT TO NEXT ENTRY                                              XMXQT
2    ENDDO                                                                                        XMXQT
```

```
171  2    SET PARM1 = 0 (NORMAL COMPLETION)                    XMXQT
172  2    PERFORM :END: - NO RETURN EXPECTED                   XMXQT
173  2    :ERR01:  SET PARM1 = 1                               XMXQT
174  2    :ERR08:  SET PARM1 = 8 AND PARMS TO APPROPRIATE REASON CODE  XMXQT
175  2    :END:                                                XMXQT
176  2    CALL SLIBR TO BECOME PRIVLEDGED                      XMXQT
177  2    SET CURRENT TASK IN MGR, STATUS TABLE, AND AVA TO EXEC  XMXQT
178  2    CALL SLIBX TO BECOME UN-PRIVLEDGED                   XMXQT
179  2    CALL XMPAW TO POST EXEC AND WAIT FOR NEXT REQUEST    XMXQT
180  1  END XMXQT
```

```
182   1 *D0       CALLING PROCEDURE                                                                  XMAFR
183   1 *D0                                                                                          XMAFR
184   1 *D0    JSB  XMAFR                                                                            XMAFR
185   1 *D0    DEF  *+3                                                                              XMAFR
186   1 *D0    DEF  ADDR                                                                             XMAFR
187   1 *D0    DEF  SIZE                                                                             XMAFR
188   1 *D0                                                                                          XMAFR
189   1 **********                                                                                   XMAFR
190   1 *D1                                                                                          XMAFR
191   1 *D1    PLACE A FE ON THE FE CHAINS AND MERGE WITH ANY ADJACENT FES                           XMAFR
192   1 *D1                                                                                          XMAFR
193   1 **********                                                                                   XMAFR
194   1 *D2       INPUT                                                                              XMAFR
195   1 *D2    ADDR   - ADDRESS OF AREA BEING FREED                                                  XMAFR
196   1 *D2    SIZE   - SIZE OF AREA BEING FREED.  IF LESS THAN 3 THE FOLLOWING                      XMAFR
197   1 *D2             WORD(S) WILL ALSO BE FREED SUCH THAT THE MINIMUM FE SIZE                     XMAFR
198   1 *D2             OF 3 WORDS IS MAINTAINED.                                                    XMAFR
199   1 *D2                                                                                          XMAFR
200   1 *D2       EXTERNAL SYMBOLS FROM XMAWA                                                        XMAFR
201   1 *D2    XMBCP, XMFCP, XMFNC, XMFRE                                                            XMAFR
202   1 *D2                                                                                          XMAFR
203   1 **********                                                                                   XMAFR
204   1 *D3       OUTPUT (EXTERNAL SYMBOLS FROM XMAWA)                                               XMAFR
205   1 *D3    XMBCP, XMFCP, XMFNC, XMFRE                                                            XMAFR
206   1 *D3                                                                                          XMAFR
207   1 **********                                                                                   XMAFR
208   1 *D5       NOTES                                                                              XMAFR
209   1 *D5    USES .ENTR                                                                            XMAFR
210   1 *D5                                                                                          XMAFR
211   1 **********                                                                                   XMAFR
```

```
213   1 BEGIN XMAFR                                                                                    XMAFR
214   2    SET NEW FE SIZE FIELD TO MAX(SIZE,3)                                                        XMAFR
215   2    INCREMENT TOTAL FREE SPACE BY NEW FE SIZE                                                   XMAFR
216   2    INDEX TO FORWARD CHAIN POINTER (FCP) HEAD                                                   XMAFR
217   2    START SEARCH WHILE FCP NOT = END-OF-CHAIN (-32768)                                          XMAFR
218   2    EXIT IF 'ADDR' < FCP VALUE                                                                  XMAFR
219   3       SET NEW FE FCP TO CURRENT FCP VALUE                                                      XMAFR
220   3       SET CURRENT FCP VALUE TO 'ADDR'                                                          XMAFR
221   3       SET NEW FE BACKWARD CHAIN POINTER (BCP) TO NEXT FE BCP VALUE                             XMAFR
222   3       SET NEXT FE'S BCP VALUE TO 'ADDR'                                                        XMAFR
223   3       PERFORM MERGE TO ATTEMPT COMBINATION OF NEW FE AND NEXT FE                               XMAFR
224   2    OR ELSE                                                                                     XMAFR
225   3       INDEX TO NEXT FE FCP                                                                     XMAFR
226   2    END LOOP                                                                                    XMAFR
227   3       SET NEW FE'S FCP VALUE TO CURRENT FE'S FCP VALUE (-32768)                                XMAFR
228   3       SET CURRENT FE'S FCP VALUE TO 'ADDR'                                                     XMAFR
229   3       SET NEW FE BCP TO BCP HEAD VALUE                                                         XMAFR
230   3       SET BCP HEAD TO 'ADDR'                                                                   XMAFR
231   2    END SEARCH                                                                                  XMAFR
232   2    IF NEW FE BCP NOT = END-OF-CHAIN                                                            XMAFR
233   2    THEN                                                                                        XMAFR
234   3       PERFORM MERGE TO ATTEMPT COMBINATION OF PREVIOUS FE AND NEW FE                           XMAFR
235   2    END IF                                                                                      XMAFR
236   2    IF TOC SPACE FENCE IS WITHIN BOUNDARIES OF THE NEW                                          XMAFR
237   3       (OR CONSOLIDATED) FE, THEN                                                               XMAFR
238   3       MOVE THE TOC SPACE FENCE TO BE ORIGIN OF THIS FE                                         XMAFR
239   2    ENDIF                                                                                       XMAFR
240   1 END XMAFR                                                                                      XMAFR
241   1 BEGIN MERGE                                                                                    XMAFR
242   2    IF FE 1 IS ADJACENT TO FE 2                                                                 XMAFR
243   2    THEN                                                                                        XMAFR
244   3       INCREMENT FE 1 SIZE FIELD BY FE 2 SIZE FIELD                                             XMAFR
245   3       SET FE 1 FCP TO VALUE OF FE 2 FCP                                                        XMAFR
246   3       IF FE 1 FCP NOT = END-OF-CHAIN                                                           XMAFR
247   3       THEN                                                                                     XMAFR
248   4          SET FE 3 BCP TO ADDRESS OF FE 1                                                       XMAFR
249   3       ELSE                                                                                     XMAFR
250   4          SET BCP HEAD TO ADDRESS OF FE 1                                                       XMAFR
251   3       ENDIF                                                                                    XMAFR
252   2    ENDIF                                                                                       XMAFR
253   1 END MERGE                                                                                      XMAFR
```

```
255   1 *D0      CALLING PROCEDURE                                                    XMAGT
256   1 *D0                                                                           XMAGT
257   1 *D0   JSB  XMAGT                                                              XMAGT
258   1 *D0   DEF  *+3                                                                XMAGT
259   1 *D0   DEF  OPTN                                                               XMAGT
260   1 *D0   DEF  SIZE                                                               XMAGT
261   1 *D0                                                                           XMAGT
262   1 **********                                                                    XMAGT
263   1 *D1                                                                           XMAGT
264   1 *D1   FIND A BLOCK OF FREE SPACE IN THE AWA AT LEAST 'SIZE' WORDS LARGE       XMAGT
265   1 *D1                                                                           XMAGT
266   1 **********                                                                    XMAGT
267   1 *D2      INPUT                                                                XMAGT
268   1 *D2   OPTN   - INDICATOR OF WHICH FREE CHAIN TO SEARCH                        XMAGT
269   1 *D2            0 = FORWARD POINTER CHAIN (FOR TOC SPACE)                      XMAGT
270   1 *D2            1 = BACKWARD POINTER CHAIN (FOR DATA SPACE)                    XMAGT
271   1 *D2   SIZE   - NUMBER OF WORDS NEEDED (A MINIMUM OF 3 WORDS WILL BE           XMAGT
272   1 *D2            ALLOCATED EVEN IF 'SIZE' IS 1 OR 2)                            XMAGT
273   1 *D2                                                                          XMAGT
274   1 *D2      EXTERNAL SYMBOLS FROM XMAWA                                         XMAGT
275   1 *D2   XMFCP, XMFNC, XMFRE                                                    XMAGT
276   1 *D2                                                                          XMAGT
277   1 **********                                                                    XMAGT
278   1 *D3      OUTPUT                                                              XMAGT
279   1 *D3   A-REG  - ADDRESS OF ALLOCATED BLOCK OR -32768 (OCTAL 100000)           XMAGT
280   1 *D3            INDICATING NONE AVAILABLE                                     XMAGT
281   1 *D3                                                                          XMAGT
282   1 *D3      EXTERNAL SYMBOLS FROM XMAWA                                         XMAGT
283   1 *D3   XMFCP, XMFNC, XMFRE                                                    XMAGT
284   1 *D3   XMPX1,XMAPK                                                            XMAGT
285   1 *D3                                                                          XMAGT
286   1 **********                                                                    XMAGT
287   1 *D5      NOTES                                                               XMAGT
288   1 *D5   IF SOME FE EXACTLY 'SIZE' WORDS OR >= SIZE+3 WORDS IS NOT FOUND AN     XMAGT
289   1 *D5   ERROR RETURN (A-REG = -32768) IS TAKEN                                 XMAGT
290   1 *D5                                                                          XMAGT
291   1 *D5   USES .ENTR                                                             XMAGT
292   1 *D5                                                                          XMAGT
293   1 **********                                                                    XMAGT
```

```
295    1 BEGIN XMAGT                                                          XMAGT
296    2    IF TOTAL FREE SPACE < MAX(SIZE,3)                                 XMAGT
297    2    THEN                                                              XMAGT
298    3       CALL XMPK1   PURGE DWA ELEMENTS FROM AWA                       XMAGT
299    3       SET PHASE1 FLAG                                                XMAGT
300    3       IF TOTAL FREE SPACE < MAX(3,SIZE), THEN                        XMAGT
301    4          RETURN VALUE= -32768                                        XMAGT
302    3       EXIT TO :XMAGR                                                 XMAGT
303    3       ENDIF                                                          XMAGT
304    2    ENDIF                                                             XMAGT
305    2    PERFORM XMSRC(OPTN,SIZE)                                          XMAGT
306    2    IF RETURN CODE IS NOT FOUND, THEN                                 XMAGT
307    3       CALL XMAPK                                                     XMAGT
308    3       IF RETURN CODE IS NOT FOUND, THEN                              XMAGT
309    4          RETURN VALUE= -32768                                        XMAGT
310    3       EXIT TO :XMAGR                                                 XMAGT
311    3       ENDIF                                                          XMAGT
312    3       RETURN VALUE= AREA                                             XMAGT

313    3       :XMAGR                                                         XMAGT
314    2    ENDIF                                                             XMAGT
315    2    CLEAR PHASE1 FLAG                                                 XMAGT
316    1 END XMAGT                                                            XMAGT
```

```
318    1 BEGIN  XMSRC                                                                  XMAGT
319    2    SET INDEX TO APPROPRIATE CHAIN HEAD, I.E., FEHEAD(OPTN)                    XMAGT
320    2    START SEARCH WHILE POINTER NOT = END-OF-CHAIN (-32768), AND                XMAGT
321    3       WHILE TOC SPACE FENCE HAS NOT BEEN CROSSED                              XMAGT
322    2    EXIT IF FE SIZE = MAX('SIZE', 3)                                           XMAGT
323    3       DECREMENT TOTAL FREE SPACE BY MAX('SIZE', 3)                            XMAGT
324    3       DECHAIN FE                                                              XMAGT
325    3       RETURN ADDRESS OF AREA                                                  XMAGT
326    2    EXIT IF FE SIZE >= MAX('SIZE', 3) + 3                                      XMAGT
327    3       DECREMENT TOTAL FREE SPACE BY MAX(SIZE,3)                               XMAGT
328    3       IF ALLOCATING FROM HEAD OF SPACE (OPTN = 0)                             XMAGT
329    3       THEN                                                                    XMAGT
330    4          CREATE CHAIN POINTERS AND SIZE FIELDS IN BOTTOM OF SPACE             XMAGT
331    4          RECHAIN NEW FE                                                       XMAGT
332    4          RETURN ADDRESS OF AREA                                               XMAGT
333    3       ELSE                                                                    XMAGT
334    4          CHANGE SIZE FIELD TO FE SIZE - MAX(SIZE,3)                           XMAGT
335    4          COMPUTE AND RETURN ADDRESS OF AREA                                   XMAGT
336    3       ENDIF                                                                   XMAGT
337    2    OR ELSE                                                                    XMAGT
338    3       INDEX TO NEXT FE                                                        XMAGT
339    2    END LOOP                                                                   XMAGT
340    3       SET RETURN CODE TO 0(NOT FOUND)                                         XMAGT
341    2    END SEARCH                                                                 XMAGT
342    2    IF TOC SPACE WAS FOUND AT THE TOC SPACE FENCE, THEN                        XMAGT
343    3       INCREMENT TOC SPACE FENCE BY 'SIZE'                                     XMAGT
344    2    ENDIF                                                                      XMAGT
345    1 END XMSRC                                                                     XMAGT
```

```
347   1 *D0      CALLING PROCEDURE                                                    XMAMG
348   1 *D0                                                                           XMAMG
349   1 *D0    JSB  XMAMG                                                            XMAMG
350   1 *D0    DEF  CLSNO                                                            XMAMG
351   1 *D0                                                                          XMAMG
352   1 **********                                                                   XMAMG
353   1 *D1                                                                          XMAMG
354   1 *D1    PROVIDE AWA MANAGEMENT BASED ON REQUEST LIST (SEE FDS SDD             XMAMG
355   1 *D1    TABLE 6.2 - III)                                                      XMAMG
356   1 *D1                                                                          XMAMG
357   1 **********                                                                   XMAMG
358   1 *D2      INPUT                                                               XMAMG
359   1 *D2    CLSNO  - CLASS I/O NUMBER CONTAINING REQUEST LIST                     XMAMG
360   1 *D2                                                                          XMAMG
361   1 *D2    REQUEST LIST (SEE TABLE 6.2 - III)                                    XMAMG
362   1 *D2                                                                          XMAMG
363   1 **********                                                                   XMAMG
364   1 *D3      OUTPUT                                                              XMAMG
365   1 *D3    ID SEGMENT PARAMETERS (SEE TABLE 6.2 - IV)                            XMAMG
366   1 *D3    REQUEST LIST FIELD 8                                                  XMAMG
367   1 *D3                                                                          XMAMG
368   1 **********                                                                   XMAMG
369   1 *D5      NOTES                                                               XMAMG
370   1 *D5    USES EXEC,XMAFR,XMAGT,XMAPK,XMTFN                                     XMAMG
371   1 *D5         XMDIN,XMDAL,XMDDA,XMDST,XMDRT                                    XMAMG
372   1 *D5                                                                          XMAMG
373   1 *D5    NEITHER XEXEC NOR THE PROCESSORS WILL MAKE AWA MANAGEMENT REQUESTS    XMAMG
374   1 *D5    FOR DWA DATA (CLASS 3 & 5).  ONLY THE MANAGER IS AWARE OF THE DWA     XMAMG
375   1 *D5    AND IT WILL DUPLICATE INTERFACE TABLES AND SEQUENCE TABLES IN TO       XMAMG
376   1 *D5    THE DWA AND COPY THEM BACK TO THE AWA AS NECESSARY.                   XMAMG
377   1 *D5                                                                          XMAMG
378   1 **********                                                                   XMAMG
```


C-3

```
380   1 BEGIN XMAMG                                                                                                    XMAMG
381   2    RETRIEVE AWA MANAGEMENT REQUEST LIST FROM CLASS I/O NUMBER                                                  XMAMG
382   2    CLEAR RETURN PARM1                                                                                          XMAMG
383   2    CLEAR REWRITE FLAG                                                                                          XMAMG
384   2    INITIALIZE TO FIRST REQUEST CODE                                                                            XMAMG
385   2    DO UNTIL END OF LIST (0), EIGHT REQUESTS PROCESSED OR PARM1 > ZERO                                          XMAMG
386   3       IF REQUEST FOR TOC (CODE 16)                                                                             XMAMG
387   3       THEN                                                                                                     XMAMG
388   4          IF REQUESTED SIZE .GT. TOC SIZE, THEN                                                                 XMAMG
389   5             CALL XMAPK TO COLLAPSE AWA IF TOC TOO SCATTERED FOR XEXEC BUFFER SIZE                               XMAMG
390   4          ENDIF                                                                                                 XMAMG
391   4          WRITE CHAIN HEADS, TOTAL FREE SPACE AND TOC TO CLASS I/O                                              XMAMG
392   4          STORE CLASS I/O NUMBER IN REQUEST WORD EIGHT                                                          XMAMG
393   4          SET REQUESTED SIZE FROM TOC SIZE                                                                      XMAMG
394   4          SET REWRITE FLAG                                                                                      XMAMG
395   3       ELSE                                                                                                     XMAMG
396   4          IF REQUES TO CLEAR (CODE 17)                                                                          XMAMG
397   4          THEN                                                                                                  XMAMG
398   5             GET A(&DWA) FROM XMDWA                                                                             XMAMG
399   5             SAVE HEADER AND DIRECTORY-SIZE                                                                     XMAMG
400   5             CLEAR XMHD0 THRU XMHD8                                                                             XMAMG
401   5             BUILD AN FE AT XMAWA FOR AWA SIZE                                                                  XMAMG
402   5             CALL XMAGT TO  ALLOCATE A TOC ENTRY FOR &DWA                                                       XMAMG
403   5             CHAIN IN TOC ENTRY TO XMHD0                                                                        XMAMG
404   5             CALL XMAGT TO  ALLOCATE SPACE FOR &DWA                                                             XMAMG
405   5             SET LOCATION, SIZE, & KEY IN THE TOC                                                               XMAMG
406   5             SET DIRECTORY ADDRESS AT XMDWA                                                                     XMAMG
407   5             CLEAR THE DIRECTORY                                                                                XMAMG
408   5             SET LU, TRACK NUMBER, & NUMBER OF TRACKS IN THE DIRECTORY                                          XMAMG
409   4          ELSE                                                                                                  XMAMG
410   5             CALL XMTFN TO SEARCH TOC FOR INDICATED ENTRY                                                       XMAMG
411   5             CASE (:VERIFY:, :VERALO:, :VERALO:, :RENAME:, :DELVER:, :DELVER:, :STORE:,                         XMAMG
412   6                (:RETRVE:, :RETRVE:) REQUEST CODE                                                               XMAMG

413   6                :VERIFY:                                                                                        XMAMG
414   6                IF ENTRY NOT FOUND                                                                              XMAMG
415   6                THEN                                                                                            XMAMG
416   7                   SET RETURN PARM1 AND PARM2 (2 & INDEX)                                                       XMAMG
417   6                ENDIF                                                                                           XMAMG

418   6                :VERALO:                                                                                        XMAMG
419   6                IF ENTRY ALREADY EXISTS                                                                         XMAMG
420   6                THEN                                                                                            XMAMG
421   7                   IF ALLOCATE REQUEST (3)                                                                      XMAMG
422   7                   THEN                                                                                         XMAMG
423   8                      SET RETURN PARM1 AND PARM2 (3 & INDEX)                                                    XMAMG
424   7                   ELSE                                                                                         XMAMG
425   8                      IF TYPE, SIZE AND I-DIM FIELDS DO NO MATCH                                                XMAMG
426   8                      THEN                                                                                      XMAMG
427   9                         SET RETURN PARM1 AND PARM2 (4 & INDEX)                                                 XMAMG
428   8                      ENDIF                                                                                     XMAMG
429   7                   ENDIF                                                                                        XMAMG
430   7                ELSE                                                                                            XMAMG
431   7                   CALL XMAGT TO ALLOCATE TOC SPACE                                                             XMAMG
432   7                   IF CLASS EQ 3 OR 8, THEN                                                                     XMAMG
433   8                      CHAIN IN NEW TOC ENTRY                                                                    XMAMG
434   8                      SET DATA SPACE ADDRESS TO ZERO                                                            XMAMG
435   7                   ELSE                                                                                         XMAMG
436   8                      CALL XMAGT TO ALLOCATE DATA SPACE                                                         XMAMG
```

```
437   8              IF SPACE NOT AVAILABLE                                                                XMAMG
438   8              THEN                                                                                  XMAMG
439   9                 SET RETURN PARM1 AND PARM2 (1 & INDEX)                                             XMAMG
440   8              ELSE                                                                                  XMAMG
441   9                 IF DATA ELEMENT (CLASS 2)                                                          XMAMG
442   9                 THEN                                                                               XMAMG
443  10                    IF CHARACTER STRING (TYPE 4 - 8)                                                XMAMG
444  10                    THEN                                                                            XMAMG
445  11                       INITIALIZE AREA TO BLANKS                                                    XMAMG
446  10                    ELSE                                                                            XMAMG
447  11                       INITIALIZE AREA TO ZEROS                                                     XMAMG
448  10                    ENDIF                                                                           XMAMG
449   9                 ENDIF                                                                              XMAMG
450   9                 CHAIN IN NEW TOC ENTRY                                                             XMAMG
451   8              ENDIF                                                                                 XMAMG
452   8              IF CLASS EQ 4 OR 6  AN SEQUENCE TABLE OR INTERFACE TABLE                              XMAMG
453   8              THEN  CALL XMDAL   DWA ALLOCATION                                                     XMAMG
454   8              ENDIF                                                                                 XMAMG
455   7           ENDIF                                                                                    XMAMG
456   6        ENDIF                                                                                       XMAMG

457   6        :RENAME:                                                                                    XMAMG
458   6        IF ENTRY NOT FOUND                                                                          XMAMG
459   6        THEN                                                                                        XMAMG
460   7           SET RETURN PARM1 AND PARM2 (2 & INDEX)                                                   XMAMG
461   6        ELSE                                                                                        XMAMG
462   7           CALL XMTFN TO SEARCH TOC FOR NEW ENTRY AND DETERMINE CHAIN POSITION                      XMAMG
463   7           IF ENTRY FOUND                                                                           XMAMG
464   7           THEN                                                                                     XMAMG
465   8              SET RETURN PARM1 AND PARM2 (3 & INDEX)                                                XMAMG
466   7           ELSE                                                                                     XMAMG
467   8              CALL XMAGT TO ALLOCATE NEW TOC ENTRY                                                  XMAMG
468   8              IF SPACE NOT AVAILABLE                                                                XMAMG
469   8              THEN                                                                                  XMAMG
470   9                 SET RETURN PARM1 AND PARM2 (1 & INDEX)                                             XMAMG
471   8              ELSE                                                                                  XMAMG
472   9                 COPY OLD ENTRY ATTRIBUTES INTO NEW ENTRY AND CHAIN IN TO TOC                       XMAMG
473   9                 DECHAIN OLD ENTRY                                                                  XMAMG
474   9                 CALL XMAFR TO RETURN OLD ENTRY TOC SPACE TO FE POOL                                XMAMG
475   8              ENDIF                                                                                 XMAMG
476   7           ENDIF                                                                                    XMAMG
477   6        ENDIF                                                                                       XMAMG

478   6        :DELVER:                                                                                    XMAMG
479   6        IF ENTRY FOUND                                                                              XMAMG
480   6        THEN                                                                                        XMAMG
481   7           GENERATE KEY 1 LESS THAN FOUND KEY                                                       XMAMG
482   7           CALL XMTFN FOR GENERATED KEY                                                             XMAMG
483   7           DECHAIN TOC ENTRY                                                                        XMAMG
484   7           CALL XMAFR TO RETURN TOC ENTRY SPACE TO FE POOL                                          XMAMG
485   7           CALL XMAFR TO RETURN DATA SPACE TO FE POOL                                               XMAMG
486   7           IF CLASS EQ 4 OR 6, THEN                                                                 XMAMG
487   8              CALL XMDDA  DWA DEALLOCATION                                                          XMAMG
488   7           ENDIF                                                                                    XMAMG
489   6        ELSE                                                                                        XMAMG
490   7           IF DELETE REQUEST (S)                                                                    XMAMG
491   7           THEN                                                                                     XMAMG
492   8              SET RETURN PARM1 AND PARM2 (2 & INDEX)                                                XMAMG
493   7           ENDIF                                                                                    XMAMG
```

```
494   6                ENDIF                                                                                                  XMAMG

495   6                :STORE:                                                                                                XMAMG
496   6                IF ENTRY NOT FOUND                                                                                     XMAMG
497   6                THEN                                                                                                   XMAMG
498   7                   SET RETURN PARM1 AND PARM2 (2 & INDEX)                                                              XMAMG
499   6                ELSE                                                                                                   XMAMG
500   7                   IF (TOC TYPE > 0 AND INCONSISTENT WITH REQUEST TYPE) OR                                             XMAMG
501   8                      DISPLACEMENT OR SPECIFIED SIZE < 0, OR                                                           XMAMG
502   8                      DISPLACEMENT + REQUESTED SIZE > ALLOCATED SIZE                                                   XMAMG
503   7                   THEN                                                                                                XMAMG
504   8                      SET RETURN PARM1 AND PARM2 (4 & INDEX)                                                           XMAMG
505   7                   ELSE                                                                                                XMAMG
506   8                      GET DATA FROM INDICATED CLASS I/O; STORE INTO AWA                                                XMAMG
507   8                      FREE CLASS NUMBER                                                                                XMAMG
508   8                      IF CLASS EQ 4 OR 6, THEN                                                                         XMAMG
509   9                         CALL XMDST  DWA STORE DATA                                                                    XMAMG
510   8                      ENDIF                                                                                            XMAMG
511   7                   ENDIF                                                                                               XMAMG
512   6                ENDIF                                                                                                  XMAMG

513   6                :RETRVE:                                                                                               XMAMG
514   6                IF ENTRY NOT FOUND                                                                                     XMAMG
515   6                THEN                                                                                                   XMAMG
516   7                   SET RETURN PARM1 AND PARM2 (2 & INDEX)                                                              XMAMG
517   6                ELSE                                                                                                   XMAMG
518   7                   IF VALUES REQUESTED (8)                                                                             XMAMG
519   7                   THEN                                                                                                XMAMG
520   8                      IF (TOC-TYPE .NE. 0 AND .NE. REQUESTED-TYPE) , OR                                                XMAMG
521   9                         DISPLACEMENT OR SPECIFIED SIZE < 0, OR                                                        XMAMG
522   9                         DISPLACEMENT + SPECIFIED SIZE > ALLOCATED SIZE                                                XMAMG
523   8                      THEN                                                                                             XMAMG
524   9                         SET RETURN PARM1 AND PARM2 (4 & INDEX)                                                        XMAMG
525   8                      ELSE                                                                                             XMAMG
526   9                         IF REQUESTED SIZE = ZERO                                                                      XMAMG
527   9                         THEN                                                                                          XMAMG
528  10                            CALCULATE AMOUNT OF DATA TO RETRIEVE AS ACTUAL SIZE MINUS DISPLACEMENT                     XMAMG
529  10                            STORE COMPUTED SIZE IN REQUEST WORD SIX                                                    XMAMG
530   9                         ENDIF                                                                                         XMAMG
531   9                         IF CLASS EQ 4 OR 6 AND TOC ADDRESS EQ 0, THEN                                                 XMAMG
532  10 *     THE ELEMENT EXISTS ONLY ON THE DWA                                                                              XMAMG
533  10                            CALL XMDRT    MOVE INTO AWA                                                                XMAMG
534  10                            IF NO SPACE THEN                                                                           XMAMG
535  11                               SET RETURN PARM1 AND PARM2 TO(1, INDEX)                                                 XMAMG
536  10                            EXIT TO :XMREX                                                                             XMAMG
537  10                            ENDIF                                                                                      XMAMG
538   9                         ENDIF                                                                                         XMAMG
539   9                         WRITE VALUES TO CLASS I/O                                                                     XMAMG
540   9                         STORE TYPE IN LOW BYTE OF REQUEST WORD 1                                                      XMAMG
541   9                         STORE CLASS NUMBER IN REQUEST WORD EIGHT                                                      XMAMG
542   9                         SET REWRITE FLAG                                                                              XMAMG
543   8                      ENDIF                                                                                            XMAMG
544   7                   ELSE                                                                                                XMAMG
545   8                      WRITE TOC ENTRY TO CLASS I/O                                                                     XMAMG
546   8                      STORE CLASS NUMBER IN REQUEST WORD EIGHT                                                         XMAMG
547   8                      SET REWRITE FLAG                                                                                 XMAMG

548   8                      :XMREX                                                                                           XMAMG
549   7                   ENDIF                                                                                               XMAMG
```

```
550   6                ENDIF                                                                                    XMAMG
551   5              END CASE                                                                                   XMAMG
552   4           ENDIF                                                                                         XMAMG
553   3        ENDIF                                                                                            XMAMG
554   3        INCREMENT TO NEXT REQUEST CODE                                                                   XMAMG
555   2     ENDDO                                                                                               XMAMG
556   2     STORE REWRITE FLAG IN RETURN PARMS                                                                  XMAMG
557   2     IF REWRITE FLAG SET                                                                                 XMAMG
558   2     THEN                                                                                                XMAMG
559   3        WRITE REQUEST LIST BACK TO COMMUNICATIONS CLASS I/O NUMBER                                       XMAMG
560   2     ENDIF                                                                                               XMAMG
561   2     PAW ASSOCIATED TASK WITH RETURN PARMS                                                               XMAMG
562   1 END XMAMG                                                                                               XMAMG
```

```
564   1 *D0       CALLING PROCEDURE                                                                  XMTFN
565   1 *D0                                                                                          XMTFN
566   1 *D0    JSB  XMTFN                                                                            XMTFN
567   1 *D0    DEF  *+2                                                                              XMTFN
568   1 *D0    DEF  KEYS                                                                             XMTFN
569   1 *D0                                                                                          XMTFN
570   1 **********                                                                                   XMTFN
571   1 *D1                                                                                          XMTFN
572   1 *D1    EXAMINE THE TOC FOR AN ENTRY EQUAL TO 'KEYS'                                          XMTFN
573   1 *D1                                                                                          XMTFN
574   1 **********                                                                                   XMTFN
575   1 *D2       INPUT                                                                              XMTFN
576   1 *D2    KEYS   - ADDRESS OF FOUR WORD KEY TO BE LOCATED IN THE TOC                            XMTFN
577   1 *D2                                                                                          XMTFN
578   1 *D2       EXTERNAL SYMBOLS FROM XMAWA                                                        XMTFN
579   1 *D2    XMHD                                                                                  XMTFN
580   1 *D2                                                                                          XMTFN
581   1 **********                                                                                   XMTFN
582   1 *D3       OUTPUT                                                                             XMTFN
583   1 *D3    A-REG  - ADDRESS OF TOC ENTRY MATCHING 'KEY' OR                                       XMTFN
584   1 *D3             ADDRESS (WITH INDIRECT BIT SET) OF PREVIOUS TOC ENTRY                        XMTFN
585   1 *D3             WHERE 'KEYS' COULD BE CHAINED IN                                             XMTFN
586   1 *D3                                                                                          XMTFN
587   1 **********                                                                                   XMTFN
588   1 *D5       NOTES                                                                              XMTFN
589   1 *D5    USES .ENTR                                                                            XMTFN
590   1 *D5                                                                                          XMTFN
591   1 **********                                                                                   XMTFN
592   1 *                                                                                            XMTFN
593   1 *                                                                                            XMTFN
594   1 *                                                                                            XMTFN
595   1 *                                                                                            XMTFN
596   1 BEGIN XMTFN                                                                                  XMTFN
597   2    ISOLATE CLASS FROM KEY AND INDEX TO APPROPRIATE CHAIN HEAD                                XMTFN
598   2    START SEARCH WHILE CHAIN POINTER NOT = END-OF-CHAIN (-32768)                              XMTFN
599   3       COMPARE LAST THREE WORDS OF 'KEYS' TO TOC ENTRY                                        XMTFN
600   2    EXIT IF MATCH                                                                             XMTFN
601   3       RETURN ADDRESS OF ENTRY                                                                XMTFN
602   2    EXIT IF 'KEYS' < TOC ENTRY (EXPECTED ENTRY NOT IN CHAIN)                                  XMTFN
603   3       RETURN ADDRESS OF PREVIOUS ENTRY WITH INDIRECT BIT SET                                 XMTFN
604   2    OR ELSE                                                                                   XMTFN
605   3       INDEX TO NEXT TOC ENTRY ON CHAIN                                                       XMTFN
606   2    END LOOP                                                                                  XMTFN
607   3       RETURN ADDRESS OF LAST (PREVIOUS) ENTRY WITH INDIRECT BIT SET                          XMTFN
608   2    END SEARCH                                                                                XMTFN
609   1 END XMTFN                                                                                    XMTFN
```

```
611   1 *D0      CALLING PROCEDURE                                                    XMSST
612   1 *D0                                                                           XMSST
613   1 *D0   JSB  XMSST                                                              XMSST
614   1 *D0                                                                           XMSST
615   1 **********                                                                    XMSST
616   1 *D1                                                                           XMSST
617   1 *D1   ADJUST SEQUENCE LOCATION COUNTER                                        XMSST
618   1 *D1                                                                           XMSST
619   1 **********                                                                    XMSST
620   1 *D2      INPUT                                                                XMSST
621   1 *D2   A-REG  - INTEGER VALUE OF SEQUENCE NUMBER TO BE LOCATED                 XMSST
622   1 *D2                                                                           XMSST
623   1 *D2      EXTERNAL SYMBOLS                                                     XMSST
624   1 *D2   SEQAD - ADDRESS OF EXECUTING SEQUENCE TABLE IN AWA                      XMSST
625   1 *D2   ENDTB - DISPLACEMENT TO LAST ENTRY OF EXECUTING SEQUENCE TABLE          XMSST
626   1 *D2                                                                           XMSST
627   1 **********                                                                    XMSST
628   1 *D3      OUTPUT                                                               XMSST
629   1 *D3   A-REG - DISPLACEMENT TO ENTRY IF SEQUENCE NUMBER IS FOUND, ELSE -32768  XMSST
630   1 *D3                                                                           XMSST
631   1 **********                                                                    XMSST
632   1 *                                                                             XMSST
633   1 *                                                                             XMSST
634   1 *                                                                             XMSST
635   1 *                                                                             XMSST
636   1 BEGIN XMSST                                                                   XMSST
637   2    SAVE RESET SEQUENCE NUMBER IN LOCAL VARIABLE SEQNO                         XMSST
638   2    SET DISPLACEMENT IN A-REG TO ZERO                                          XMSST
639   2    GET ADDRESS OF FIRST ENTRY IN SEQUENCE TABLE INTO X-REG                    XMSST
640   2    START SEARCH UNTIL CURRENT ENTRY DISPLACEMENT > LAST ENTRY DISPLACEMENT    XMSST
641   2    EXIT IF INDICATED NUMBER FOUND                                             XMSST
642   2    OR ELSE                                                                    XMSST
643   3       INCREMENT DISPLACEMENT IN A-REG TO NEXT ENTRY                           XMSST
644   2    END LOOP                                                                   XMSST
645   3       SET A-REG TO -32768 TO INDICATE NUMBER NOT FOUND                        XMSST
646   2    END SEARCH                                                                 XMSST
647   1 END XMSST                                                                     XMSST
```

```
649   1 BEGIN    XMPA%                                                                                    XMPAW
650   2 *DO                                                                                               XMPAW
651   2 *DO   ENTRY:  JSB XMPAW           OR  CALL XMPRM                                                 XMPAW
652   2 *DO           DEF RETURN ADDRESS                                                                  XMPAW
653   2 *D2                                                                                               XMPAW
654   2 *D2   INPUTS: XMPRM HAS REPLY                                                                     XMPAW
655   2 *D2                                                                                               XMPAW
656   2 *D3                                                                                               XMPAW
657   2 *D3   OUTPUTS: XMPRM HAS REQUEST,                                                                 XMPAW
658   2 *D3       AND XVSTB IS UPDATED.                                                                   XMPAW
659   2 *D3                                                                                               XMPAW
660   2    DO UNTIL GOOD PARMS RECEIVED OR TOP AT TERMINATES                                              XMPAW
661   3       IF ABORT CURRENT FLAG CN(- STBLU) THEN                                                      XMPAW
662   4          SET ABORT FLAG IN XMPRM                                                                  XMPAW
663   4          TURN OFF ABORT CURRENT FLAG                                                              XMPAW
664   3       ENDIF                                                                                       XMPAW
665   3       GET CURRENT TOP ASSOCIATED TASK(AT)                                                         XMPAW
666   3       SET IN PARM 1 FIELD OF MANAGER'S ID SEGMENT                                                 XMPAW
667   3       JSB  XVPAW                                                                                  XMPAW
668   3       DEF   *+3       RETURN                                                                      XMPAW
669   3       DEC   0         MANAGER CALL                                                                XMPAW
670   3       DEF   XMPRM     PARM FIELD                                                                  XMPAW
671   3 *   THIS IS AN IMPLIED WAIT                                                                       XMPAW

672   3       :XMSCH   GET CURRENT XVSTB ENTRY (XUSTA)                                                    XMPAW
673   3       IF THERE HAS BEEN A CALL TO PRTN (P1 FIELD IS NOT TOP AT) OR                                XMPAW
674   4          ID SEGMENT IS DORMANT OR                                                                 XMPAW
675   4          ID SEGMENT IS NOT OUR SON THEN (TOP AT HAS TERMINATED)                                   XMPAW
676   4          IF PARM IS NOT 0,3,8,9, OR -32768 THEN                                                   XMPAW
677   5             SET PARM TO -32768                                                                    XMPAW
678   5             PRINT ERROR "INVALID REQUEST"                                                         XMPAW
679   4          ELSE                                                                                     XMPAW
680   5             SET PARM FIELD FROM MANAGER'S ID SEGMENT                                              XMPAW
681   4          ENDIF                                                                                    XMPAW
682   3       ELSE  (TOP AT IS STILL ACTIVE AND RETURNED VIA PAW)                                         XMPAW
683   4          IF PARM NOT 1 OR 2 THEN                                                                  XMPAW
684   5             CALL XMKIL TO ABORT TOP AT                                                            XMPAW
685   5             PRINT ERROR "INVALID REQUEST"                                                         XMPAW
686   4          ELSE                                                                                     XMPAW
687   5             SET PARM FROM CURRENT ID SEGMENT                                                      XMPAW
688   4          ENDIF                                                                                    XMPAW
689   3       ENDIF                                                                                       XMPAW
690   2    ENDDO                                                                                          XMPAW
691   1 END      XMPAW                                                                                    XMPAW
```

```
693   1 *D0          CALLING PROCEDURE                                                    XMDIN
694   1 *D0                                                                               XMDIN
695   1 *D0     JSB XMDIN                                                                 XMDIN
696   1 *D0                                                                               XMDIN
697   1 *D1          FUNCTION                                                             XMDIN
698   1 *D1    INITIALIZE TOC                                                             XMDIN
699   1 *D1    & DWA FUNCTIONS                                                            XMDIN
700   1 *D1                                                                               XMDIN
701   1 *D3          OUTPUT                                                               XMDIN
702   1 *D3    BREG 0=COMPLETE                                                            XMDIN
703   1 *D3         MINUS= ERROR IN INITIALIZATION                                        XMDIN
704   1 *D3                                                                               XMDIN
705   1 *D5          NOTES                                                                XMDIN
706   1 *D5    USES EXEC DISC ALLOCATION,                                                 XMDIN
707   1 *D5         XMTFN, XMAGT                                                          XMDIN
708   1 *D5                                                                               XMDIN
709   1 BEGIN XMDIN                                                                       XMDIN
710   2    GET NUMBER OF DWA TRACKS FROM P5                                               XMDIN
711   2    CALCULATE SIZE OF &DWA(3+6N) N IS # OF TRACKS                                  XMDIN
712   2    CALL XMTFN (&DWA)                                                              XMDIN
713   2    CALL XMAGT (0,8) TOC ENTRY FOR &DWA                                            XMDIN
714   2    CALL XMAGT (1,SIZE) DATA AREA FOR &DWA                                         XMDIN
715   2    INITIALIZE & CHAIN &DWA TOC ENTRY                                              XMDIN
716   2    SET # OF TRACKS FOR &DWA                                                       XMDIN
717   2    CLEAR &DWA                                                                     XMDIN
718   2    CALL EXEC (DISC TRACK ALLOCATION)                                              XMDIN
719   2    IF DISC ADDRESS .EQ. -1  TRACKS NOT AVAILABLE                                  XMDIN
720   2    THEN                                                                           XMDIN
721   3       ISSUE MESSAGE '***XM04 "N" TRACKS NOT AVAILABLE'                            XMDIN
722   3       SET ERROR RETURN                                                            XMDIN
723   2    ELSE                                                                           XMDIN
724   3       SET DISC ADDRESS IN &DWA                                                    XMDIN
725   3       SET ADDRESS OF &DWA FOR DWA MANAGEMENT                                      XMDIN
726   2    ENDIF                                                                          XMDIN
727   1 END XMDIN                                                                         XMDIN
```

```
729   1 BEGIN XMDAL                                                                                    XMDAL
730   2 *       DWA ALLOCATION                                                                         XMDAL
731   2 *D0           CALLING PROCEDURE                                                                XMDAL
732   2 *D0     JSB XMDAL                                                                              XMDAL
733   2 *D0                                                                                            XMDAL
734   2 *D1           FUNCTION                                                                         XMDAL
735   2 *D1     ALLOCATE DWA SPACE                                                                     XMDAL
736   2 *D1                                                                                            XMDAL
737   2 *D2           INPUTS                                                                           XMDAL
738   2 *D2     A(TOC ENTRY OF THE AWA ELEMENT)                                                        XMDAL
739   2 *D2      IN YREG                                                                               XMDAL
740   2 *D2                                                                                            XMDAL
741   2 *D3           OUTPUTS                                                                          XMDAL
742   2 *D3     UPDATE TOC ENTRY FOR DWA                                                               XMDAL
743   2 *D3     RETURNS BREG= ZERO- ALLOCATION COMPLETE                                                XMDAL
744   2 *D3                 = MINUS- ERROR CONDITION                                                   XMDAL
745   2 *D3                                                                                            XMDAL
746   2 *D5           NOTES                                                                            XMDAL
747   2 *D5     CALLS XRNXB, XMBST                                                                     XMDAL
748   2 *D5                                                                                            XMDAL
749   2    IF A(&DWA) .NE. 0,THEN                                                                      XMDAL
750   3       SET START-ADDRESS TO FIRST TRACK WORD IN DWA DIRECTORY                                    XMDAL
751   3       GET SIZE(IN WORDS FROM TOC ENTRY)                                                        XMDAL
752   3       SIZE(IN SECTORS)=(SIZE+63/64)                                                            XMDAL
753   3       DO FOR NUMBER OF TRACKS IN DWA OR DWA SIZE .GE. SIZE                                     XMDAL
754   4          SET BITNUM TO ZERO                                                                    XMDAL
755   4          DO UNTIL DWA SIZE .GE. SIZE OR BITNUM .GE. 96                                         XMDAL
756   5             CALL XRNXB (0,BITNUM,START-ADDRESS)                                                XMDAL
757   5             STARTBIT =BITNUM                                                                   XMDAL
758   5             CALL XRNXB (1,BITNUM,START-ADDRESS)                                                XMDAL
759   5             DWA-SIZE=BITNUM-STARTBIT                                                           XMDAL
760   4          ENDDO                                                                                 XMDAL
761   4          IF BITNUM .GE. 96, THEN                                                               XMDAL
762   5             START-ADDRESS=START ADDRESS+1 TRACK ADDRESS                                        XMDAL
763   4          ENDIF                                                                                 XMDAL
764   3       ENDDO                                                                                    XMDAL
765   3       IF DISC AREA FOUND, THEN                                                                 XMDAL
766   4 * START-ADDRESS & BITNUM DEFINES THE TRACK & SECTOR                                            XMDAL
767   4          SET DISC ADDRESS & SIZE IN TOC ENTRY                                                  XMDAL
768   4          CALL XMBST (TOC-ENTRY)                                                                XMDAL
769   4          SET NORMAL RETURN                                                                     XMDAL
770   3       ELSE                                                                                     XMDAL
                                                                                                       XMDAL
771   4          :XMDAX                                                                                XMDAL
772   4          ISSUE MESSAGE '***XM05 NO DWA SPACE REMAINING'                                        XMDAL
773   4          SET ERROR RETURN                                                                      XMDAL
774   3       ENDIF                                                                                    XMDAL
775   2    ENDIF                                                                                       XMDAL
776   1 END XMDAL
```

```
778  1 BEGIN XMDDA                                                    XMDDA
779  2 *    DVA DEALLOCATION                                          XMDDA
780  2 *05      CALLING PROCEDURE                                     XMDDA
781  2 *00      JSB XMDDA                                             XMDDA
782  2 *00                                                            XMDDA
783  2 *01          FUNCTION                                          XMDDA
784  2 *01      DELETE DVA ELEMENT WHICH CORRESPONDS                  XMDDA
785  2 *01      TO THE AVA ELEMENT                                    XMDDA
786  2 *01                                                            XMDDA
787  2 *02          INPUTS                                            XMDDA
788  2 *02      TOC ENTRY ADDRESS OF AVA ELEMENT                      XMDDA
789  2 *02        IN YREG                                             XMDDA
790  2 *02                                                            XMDDA
791  2 *05          NOTES                                             XMDDA
792  2 *05      USES XMBST                                            XMDDA
793  2 *05                                                            XMDDA
794  2      IF A(BDVA) .NE. 0, THEN                                   XMDDA
795  3          GET DISC ADDRESS                                      XMDDA
796  3          GET DATA SIZE                                         XMDDA
797  3      * FREE THE DISK AREA                                      XMDDA
798  3          CALL XMBST (TOC ENTRY)                                XMDDA
799  2      ENDIF                                                     XMDDA
800  1 END XMDDA                                                      XMDDA
```

```
802    1 BEGIN XMDST                                                                  XMDST
803    2 *      DWA STORE DATA                                                        XMDST
804    2 *D0          CALLING PROCEDURE                                               XMDST
805    2 *D0    JSB XMDST                                                             XMDST
806    2 *D0                                                                          XMDST
807    2 *D1          FUNCTION                                                        XMDST
808    2 *D1    UPDATE DWA ELEMENT WHICH CORRESPONDS                                  XMDST
809    2 *D1    TO THE AWA ELEMENT                                                    XMDST
810    2 *D1                                                                          XMDST
811    2 *D2          INPUTS                                                          XMDST
812    2 *D2    TOC ENTRY ADDRESS OF AWA ELEMENT                                      XMDST
813    2 *D2     IN YREG                                                              XMDST
814    2 *D2                                                                          XMDST
815    2 *D3          OUTPUTS                                                         XMDST
816    2 *D3    UPDATE DWA ELEMENT ON DISC TRACKS                                     XMDST
817    2 *D3                                                                          XMDST
818    2 *D5          NOTES                                                           XMDST
819    2 *D5    USES WRITE                                                            XMDST
820    2 *D5                                                                          XMDST
821    2    IF A(&DWA) .NE. 0, THEN                                                   XMDST
822    3       SET DISC ADDRESS FOR THE WRITE                                         XMDST
823    3       SET DATA ADDRESS FOR THE WRITE                                         XMDST
824    3       WRITE DATA                                                             XMDST
825    2    ENDIF                                                                     XMDST
826    1 END XMDST                                                                    XMDST
```

```
1 BEGIN XMDRT                                                    XMDRT
2 *         DWA RETRIEVE                                         XMDRT
2 *D0             CALLING PROCEDURE                              XMDRT
2 *D0       JSB XMDRT                                            XMDRT
2 *D0                                                            XMDRT
2 *D1             FUNCTION                                       XMDRT
2 *D1       RETRIEVE DWA DATA INTO AWA                           XMDRT
2 *D1                                                            XMDRT
2 *D2             INPUTS                                         XMDRT
2 *D2       A(TOC ENTRY)   IN YREG                               XMDRT
2 *D2                                                            XMDRT
2 *D3             OUTPUTS                                        XMDRT
2 *D3       ADDRESS OF DATA IN THE TOC                           XMDRT
2 *D3       @RES=0, RETRIEVE SUCESSFUL                           XMDRT
2 *D3            MINUS, ERROR NO AWA DATA                        XMDRT
2 *D3                                                            XMDRT
2 *D5             NOTES                                          XMDRT
2 *D5       USES XMAGT,XMDWA,EXEC(READ)                          XMDRT
2 *D5                                                            XMDRT
2    IF NO DWA DIRECTORY , THEN                                  XMDRT
3       SET ERROR CODE -5                                        XMDRT
2    ELSE                                                        XMDRT
3       CALL XMAGT    GET DATA SPACE                             XMDRT
3       IF NO SPACE, THEN                                        XMDRT
4          SET ERROR CODE -1                                     XMDRT
3       ELSE                                                     XMDRT
4          SET DATA ADDRESS IN TOC                               XMDRT
4          GET DISC ADDRESS                                      XMDRT
4          READ DATA INTO AWA                                    XMDRT
4          SET RETURN CODE TO 0                                  XMDRT
3       ENDIF                                                    XMDRT
2    ENDIF                                                       XMDRT
1 END XMDRT                                                      XMDRT
```

```
862    1 BEGIN XMBST                                                                                                    XMBST
863    2 *D0          CALLING PROCEDURE                                                                                 XMBST
864    2 *D0     JSB XMBST                                                                                              XMBST
865    2 *D0                                                                                                            XMBST
866    2 *D1          FUNCTION                                                                                          XMBST
867    2 *D1     SET ALLOCATION & DEALLOCATION IN DWA DIRECTORY                                                         XMBST
868    2 *D1                                                                                                            XMBST
869    2 *D2          INPUTS                                                                                            XMBST
870    2 *D2     AWA TOC ENTRY                                                                                          XMBST
871    2 *D2                                                                                                            XMBST
872    2 *D3          OUTPUTS                                                                                           XMBST
873    2 *D3     UPDATES DWA DIRECTORY                                                                                  XMBST
874    2 *D3                                                                                                            XMBST
875    2    GET TRACK-ADDRESS FROM TOC                                                                                  XMBST
876    2    SUBTRACT START OF TRACKS FROM DWA DIRECTORY FOR RELATIVE TRACK                                              XMBST
877    2    DIVIDE SECTOR ADDRESS BY 16(NUMBER OF BITS/WORD)                                                            XMBST
878    2    QUOTIENT IS NUMBER OF RELATIVE WORDS                                                                        XMBST
879    2    REMAINDER IS BIT-POSITION(BP)                                                                               XMBST
880    2    WORD ADDRESS=DWA ADDRESS+RELATIVE TRACK+RELATIVE WORDS                                                      XMBST
881    2    SUBTRACT BIT-POSITION FROM 16 GIVING NBFW(NUMBER BITS IN FIRST WORD)                                        XMBST
882    2    GET DATA SIZE, ADD 63, DIVIDE BY 64 GIVING NUMBER OF SECTORS(BITS)                                          XMBST
883    2    BITS-NBFW=REMAINING BITS(RB)                                                                                XMBST
884    2    IF RB .LE. 0, THEN                                                                                          XMBST
885    3       SET NUMBER OF WORDS(NW) TO ZERO                                                                          XMBST
886    3       SET NUMBER OF BITS LAST WORD(NBLW) TO ZERO                                                               XMBST
887    3       SET NBFW TO BITS                                                                                         XMBST
888    2    ELSE                                                                                                        XMBST
889    3       DIVIDE RB BY 16                                                                                          XMBST
890    3       SET NW TO QUOTIENT                                                                                       XMBST
891    3       SET NBLW TO REMAINDER                                                                                    XMBST
892    2    ENDIF                                                                                                       XMBST
893    2    LOAD DATA POINTED TO BY WORD ADDRESS                                                                        XMBST
894    2    ROTATE LEFT (BP+NBFW-1 BITS)  SAVE BP BITS  & POSITION NBFW BITS                                            XMBST
895    2    EXCLUSIVE OR SIGN BIT  (ON TO OFF; OFF TO ON)                                                               XMBST
896    2    IF NBFW .GT. 1, THEN                                                                                        XMBST
897    3       SHIFT RIGHT(ARITHMETIC) BY NBFW-1 PROPAGATE BITS                                                         XMBST
898    2    ENDIF                                                                                                       XMBST
899    2    IF DATA IS NEGATIVE, THEN                                                                                   XMBST
900    3       SET FILL WORD TO -1                                                                                      XMBST
901    2    ELSE                                                                                                        XMBST
902    3       SET FILL WORD TO 0                                                                                       XMBST
903    2    ENDIF                                                                                                       XMBST
904    2    ROTATE RIGHT BY BP    RESET SAVED BITS                                                                      XMBST
905    2    STORE WHERE WORD-ADDRESS POINTS                                                                             XMBST
906    2    DO WHILE NW .GT. 0                                                                                          XMBST
907    3       WORD-ADDRESS=WORD-ADDRESS+1                                                                              XMBST
908    3       SET FILL WORD INTO WHERE WORD-ADDRESS POINTS                                                              XMBST
909    2    ENDDO                                                                                                       XMBST
910    2    IF NBLW .GT. 0, THEN                                                                                        XMBST
911    3       WORD-ADDRESS=WORD-ADDRESS+1                                                                              XMBST
912    3       LOAD DATA POINTED TO BY WORD-ADDRESS                                                                     XMBST
913    3       IF NBLW .GT. 1, THEN                                                                                     XMBST
914    4          ROTATE LEFT NBLW-1 BITS                                                                               XMBST
915    3       ENDIF                                                                                                    XMBST
916    3       EXCLUSIVE OR SIGN BIT                                                                                    XMBST
917    3       IF NBLW .GT. 1, THEN                                                                                     XMBST
918    4          SHIFT RIGHT ARITHMETIC BY NBLW-1                                                                      XMBST
919    3       ENDIF                                                                                                    XMBST
920    3       STORE DATA WHERE WORD-ADDRESS POINTS                                                                     XMBST
```

```
XMBST
XMBST
```



```
921     2    ENDIF
922     1 END XMBST
```

```
924   1 BEGIN XMAPK                                                          XMAPK
925   2 *D0                                                                  XMAPK
926   2 *D0            CALLING PROCEDURE                                     XMAPK
927   2 *D0       JSB XMAPK                                                  XMAPK
928   2 *D0       DEF   OPTION                                               XMAPK
929   2 *D0       DEF   PHASE  1 FLAG                                        XMAPK
930   2 *D0                                                                  XMAPK
931   2 *D1           GENERAL COLLAPSE INTERFACE                             XMAPK
932   2 *D1                                                                  XMAPK
933   2 *D3                                                                  XMAPK
934   2 *D3           OUTPUT                                                 XMAPK
935   2 *D3     RETURNS  AREA ADDRESS FOR AREA FOUND                         XMAPK
936   2 *D3     RETURNS -32768 AREA NOT FOUND                                XMAPK
937   2 *D3                                                                  XMAPK
938   2 *D5           NOTES                                                  XMAPK
939   2 *D5     USES XMPK1,XMPK2,XMPK3,XMSRC                                 XMAPK
940   2 *D5                                                                  XMAPK
941   2    IF PHASE1 FLAG NOT SET, THEN                                      XMAPK
942   3       CALL XMPK1   PURGE DWA ELEMENTS FROM AWA                       XMAPK
943   3       PERFORM XMSRC(OPTN,SIZE)                                       XMAPK
944   2    ELSE                                                              XMAPK
945   3       SET RETURN CODE TO NOT FOUND                                   XMAPK
946   2    ENDIF                                                             XMAPK
947   2    IF RETURN CODE IS NOT FOUND, THEN                                 XMAPK
948   3       IF OPTN=1(BACKWARD CHAIN), THEN                                XMAPK
949   4          CALL XMPK2   PACK TOC CHAIN                                 XMAPK
950   4          CALL XMSRC(OPTN,SIZE)                                       XMAPK
951   3       ENDIF                                                          XMAPK
952   3       IF RETURN CODE IS NOT FOUND, THEN                              XMAPK
953   4          CALL XMPK3   PACK AWA DATA AREAS                            XMAPK
954   4          PERFORM XMSRC(OPTN,SIZE)                                    XMAPK
955   3       ENDIF                                                          XMAPK
956   2    ENDIF                                                             XMAPK
957   2    SET RETURN VALUE TO RETURN CODE                                   XMAPK
958   1 END XMAPK                                                            XMAPK
```

```
1 BEGIN XMPK1          CALLING PROCEDURE
2 *DO                     JSB XMPK1
2 *DO
2 *D1
2 *D1                PHASE 1 OF COLLAPSE(PURGE DWA ELEMENTS)
2 *D1
2 *D3                   OUTPUT
2 *D3             UPDATES ADDRESS FIELD IN TOC
2 *D3             FOR EACH DWA DATA ELEMENT
2 *D3
2 *D5                   NOTES
2 *D5             USES XMAFR,XMHD4,XMHD6,XMP1
2 *D5             CURRENTLY ONLY CLASS 4 & 6 ELEMENTS
2 *D5             (INTERFACE TABLES & SEQUENCE TABLES)
2 *D5             ARE DWA ELEMENTS.
2 *D5
2        INCREMENT XMP1   UPDATE PHASE 1 COUNT
2        IF DWA DIRECTORY ADDRESS NOT EQ 0, THEN
3           SAVE X & Y REGS
3           DO FOR ALL CLASS 4 ELEMENTS
4              CALL XMAFR(DATA,SIZE)
3           ENDDO
3           DO FOR ALL CLASS 6 ELEMENTS
4              CALL XMAFR(DATA,SIZE
3           ENDDO
3           RESTORE X & Y REGS
2        ENDIF
1 END XMPK1
```

```
960  XMPK1
961  XMPK1
962  XMPK1
963  XMPK1
964  XMPK1
965  XMPK1
966  XMPK1
967  XMPK1
968  XMPK1
969  XMPK1
970  XMPK1
971  XMPK1
972  XMPK1
973  XMPK1
974  XMPK1
975  XMPK1
976  XMPK1
977  XMPK1
978  XMPK1
979  XMPK1
980  XMPK1
981  XMPK1
982  XMPK1
983  XMPK1
984  XMPK1
985  XMPK1
986  XMPK1
987  XMPK1
988  XMPK1
```

```
990   1 BEGIN  XMPK2                                                                                    XMPK2
991   2 *D0            CALLING PROCEDURE                                                                XMPK2
992   2 *D0      JSB XMPK2                                                                              XMPK2
993   2 *D0                                                                                             XMPK2
994   2 *D1                                                                                             XMPK2
995   2 *D1          PHASE 2 OF COLLAPSE(COLLAPSE TOC ENTRIES)                                          XMPK2
996   2 *D1                                                                                             XMPK2
997   2 *D3          OUTPUT                                                                             XMPK2
998   2 *D3    REORDERS TOC ENTRIES                                                                     XMPK2
999   2 *D3    UPDATES XMFNC( TOC FENCE ADDRESS)                                                        XMPK2
1000  2 *D3                                                                                            XMPK2
1001  2 *D5          NOTES                                                                              XMPK2
1002  2 *D5    USES XMFNC,XMFCP,XMTFN,XMBCP,XMAFR,XM#P2                                                 XMPK2
1003  2 *D5                                                                                            XMPK2
1004  2    INCREMENT XM#P2    UPDATE PHASE 2 COUNT                                                     XMPK2
1005  2    DO WHILE XMFNC .GT. XMFCP AND XMFCP .NE. -32768                                           XMPK2
1006  3 *    ONLY IF THERE ARE FREE ELEMENTS AND THEY ARE IN THE TOC                                   XMPK2
1007  3 *   WILL A TOC COMPRESS BE DONE.                                                               XMPK2
1008  3       GET FIRST-FREE(FRSTF) FROM XMFCP                                                        XMPK2
1009  3       GET OLDSZ FROM THE FREE ELEMENT                                                         XMPK2
1010  3       NWSIZ=OLDSZ-8                                                                           XMPK2
1011  3       IF NWSIZ .EQ. 0,THEN   THE FREE ELEMENT WILL BE DEPLETED                                XMPK2
1012  4          GET NEXT-FREE FROM FIRST-FREE'S FCP                                                  XMPK2
1013  4          SET INTO XMFCP           DELETE FROM THE FORWARD CHAIN                               XMPK2
1014  4          IF XMFCP .NE. -32768, THEN  IF NOT THE LAST FREE ELEMENT                            XMPK2
1015  5             SET NEXT-FREE'S BCP TO -32768                                                    XMPK2
1016  4          ELSE                                                                                XMPK2
1017  5             SET XMBCP TO -32768   DELETE FROM BACKWARD CHAIN                                 XMPK2
1018  4          ENDIF                                                                               XMPK2
1019  3       ELSE                                                                                    XMPK2
1020  4          OLDSZ=NWSIZ          UPDATE LENGTH IN FREE ELEMENT                                   XMPK2
1021  3       ENDIF                                                                                   XMPK2
1022  3       NWTOC=A(FRSTF)+NWSIZ   THE NEW TOC IS LAST 8 WORDS OF FIRST-FREE                        XMPK2
1023  3       OLTOC=XMFNC-8          OLD TOC IS THE ENTRY ABOVE THE FENCE                             XMPK2
1024  3       TMPKEY=OLTOC'S KEY -1  CONSTRUCT A KEY TO FIND PREVIOUS ENTRY                           XMPK2
1025  3       CALL XMTFN(TMPKEY)     FIND PREVIOUS                                                    XMPK2
1026  3       SET PRTOC FROM AREG                                                                     XMPK2
1027  3       COPY TOC ENTRY FROM OLTOC TO NWTOC                                                      XMPK2
1028  3       STORE NWTOC ADDRESS INTO PRTOC'S CHAIN                                                  XMPK2
1029  3       CALL XMAFR(OLTOC,TOC-SIZE)  FREE THE OLD TOC ENTRY                                      XMPK2
1030  2    ENDDO                                                                                      XMPK2
1031  1 END  XMPK2                                                                                    XMPK2
```

```
1033  1 BEGIN XMPK3                                                          XMPK3
1034  2 *DO       CALLING PROCEDURE                                          XMPK3
1035  2 *DO       JSB XMPK3                                                  XMPK3
1036  2 *DO                                                                  XMPK3
1037  2 *D1       FUNCTION                                                   XMPK3
1038  2 *D1     COMPRESS AVA DATA AREA BY MIGRATING DATA AREAS               XMPK3
1039  2 *D1     TO THE HIGHER ADDRESSES AND FREE AREAS TO LOWER ADDRESSES    XMPK3
1040  2 *D1                                                                  XMPK3
1041  2 *D3       OUTPUT                                                     XMPK3
1042  2 *D3     UPDATES XMBCP,XMFCP,XMFNC,XMFRE,XM#P3                        XMPK3
1043  2 *D3                                                                  XMPK3
1044  2 *D5       NOTES                                                      XMPK3
1045  2 *D5     USES XMAFR                                                   XMPK3
1046  2 *D5                                                                  XMPK3
1047  2     INCREMENT XM#P3 (NUMBER OF PHASE 3'S)                            XMPK3
1048  2     IF XMBCP .NE. -32768, THEN  THERE ARE FREE AREAS                 XMPK3
1049  3       CALL XMPK2  INSURE TOC IS COMPRESSED                           XMPK3
1050  3       DO WHILE XMBCP .GT.XMFNC  UNTIL 1 FREE AREA ADJACANT TO THE FENCE XMPK3
1051  4         IF BCP OF LAST-FREE .EQ. -32768  ONLY 1 FREE AREA            XMPK3
1052  5           SET HIGH-WATER TO XMFNC                                    XMPK3
1053  4         ELSE                MULTIPLE AREAS                           XMPK3
1054  5           SET HIGH-WATER TO BCP OF LAST-FREE                         XMPK3
1055  4         ENDIF                                                        XMPK3
1056  4         SET LOW-WATER TO A(LAST-FREE)                                XMPK3
1057  4         PERFORM XMTSC(LOW-WATER,HIGH-WATER)                          XMPK3
1058  3       EXIT IF TOC-ADDRESS .EQ. 0                                     XMPK3
1059  4         PERFOR" XMANV(TOC-ADDRESS)                                   XMPK3
1060  3       ENDDO                                                          XMPK3
1061  2     ENDIF                                                            XMPK3
1062  1 END XMPK3                                                            XMPK3
```

```
1064    1 BEGIN XMTSC    TOC SEARCH                                                    XMTSC
1065    2 *  FIND TOC ENTRY WHICH HAS DATA ADDRESS GREATER THAN                       XMTSC
1066    2 * HIGH-WATER AND LESS THAN LOW-WATER, AND HAS A DATA                       XMTSC
1067    2 * ADDRESS GREATER THAN ANY OTHER FOUND ON THIS SEARCH.                     XMTSC
1068    2 *  RETURN THE TOC-ADDRESS OR 0(NONE FOUND).                                XMTSC
1069    2 *  FIRST TOC ENTRY IS AT SYMBOL XMAWA, THE                                 XMTSC
1070    2 * LAST TOC-ENTRY IS AT XMFNC-8.                                           XMTSC
1071    2    TOC-ENTRY=A(XMAWA)                                                      XMTSC
1072    2    TEST-AD=0; TEST-TOC=0                                                   XMTSC
1073    2    DO UNTIL TOC-ENTRY .GE. XMFNC                                           XMTSC
1074    3       IF DATA ADDRESS IN TOC-ENTRY IS                                      XMTSC
1075    4          NE 0, AND IS                                                      XMTSC
1076    4          LT LOW-WATER, AND IS                                              XMTSC
1077    4          GT HIGH-WATER, AND IS                                             XMTSC
1078    4          GT TEST-AD,                                                       XMTSC
1079    3       THEN                                                                 XMTSC
1080    4          TEST-AD=DATA ADDRESS                                              XMTSC
1081    4          TEST-TOC=TOC-ENTRY                                                XMTSC
1082    3       ENDIF                                                                XMTSC
1083    3       ADD 8 TO TOC-ENT                                                     XMTSC
1084    2    ENDDO                                                                   XMTSC
1085    2    TOC-ADDRESS=TEST-TOC    RETURN 0 OR A TOC ADDRESS                       XMTSC
1086    1 END XMTSC                                                                  XMTSC
```

```
1088  1 BEGIN XMAMV     AVA MOVE                                          XMAMV
1089  2 *  MOVE THE DATA DEFINED BY THE TOC(WHICH IS                      XMAMV
1090  2 *  IMMEDIATELY ABOVE THE LAST FREE AREA) INTO                      XMAMV
1091  2 *  THE BOTTOM OF THE LAST FREE.                                    XMAMV
1092  2 *    UPDATE THE LENGTH OF THE RESULTING FREE AREA.                 XMAMV
1093  2       GET DATA ADDRESS FROM THE TOC                               XMAMV
1094  2       SAVE THE FIRST THREE WORDS OF THE DATA AREA                  XMAMV
1095  2       CALL XMAFR(DATA ADDRESS,SIZE)                               XMAMV
1096  2       GET FREE AREA FROM XMBCP                                    XMAMV
1097  2       DECREASE FREE AREA LENGTH BY MAX(DATA SIZE,3)               XMAMV
1098  2       CALCULATE NEW ADDRESS FROM FREE AREA + FREE LENGTH          XMAMV
1099  2       MOVE DATA FROM DATA ADDRESS TO NEW ADDRESS                  XMAMV
1100  2       MOVE SAVED FIRST THREE WORDS TO NEW ADDRESS                 XMAMV
1101  2       UPDATE DATA ADDRESS IN TCC WITH NEW ADDRESS                 XMAMV
1102  1 END XMAMV
```

```
 SYMBOL DEFINITION TABLE
:DELVER:   478
:END   :   175
:ERR01 :   173
:ERR08 :   174
 MERGE     241
:RENAME:   457
:RETRVE:   513
:STORE :   495
:VERALO:   418
:VERIFY:   413
 XMAFR     213
:XMAGR :   313
 XMAGT     295
 XMAMG     380
 XMAMV    1088
 XMAPK     924
 XMBST     852
 XMDAL     729
:XMDAX :   771
 XMDDA     778
 XMDIN     709
 XMDRT     828
 XMDST     802
:XMEND :    68
 XMGR        27
 XMPAW     649
 XMPK1     960
 XMPK2     990
 XMPK3    1033
:XMREX :   548
:XMSCH :   672
 XMSRC     318
 XMSST     636
 XMTFN     596
 XMTSC    1064
 XMXQT     111

@XQT F.PDLIST
```

```
 2    1 *D0      FORTRAN CALLING PROCEDURE                                    XPATR
 3    1 *D0                                                                   XPATR
 4    1 *D0   CALL XPATR (LU, INTBUF, INTLNG, MRBUFF, INUM, NAME, TYPE, SIZE, XPATR
 5    1 *D0               IDIM, DSPTY)                                        XPATR
 6    1 *D0                                                                   XPATR
 7    1 **********                                                            XPATR
 8    1 *D1                                                                   XPATR
 9    1 *D1      XPATR ALLOWS PROCESSORS TO OBTAIN THE ATTRIBUTES OF THE       XPATR
10    1 *D1      PARAMETERS REFERENCED BY THE INTERFACE TABLE.                 XPATR
11    1 *D1                                                                   XPATR
12    1 **********                                                            XPATR
13    1 *D2      INPUT                                                        XPATR
14    1 *D2   LU     - LOGICAL UNIT NUMBER OF USER TERMINAL                   XPATR
15    1 *D2   INTBUF - INPUT/OUTPUT BUFFER OF 7*(# PARAMETERS + 1) WORDS,     XPATR
16    1 *D2            ALLOCATED WITHIN THE CALLING PROGRAM TO HOLD THE       XPATR
17    1 *D2            INTERFACE TABLE HEADER. FIRST WORD MUST BE ZERO ONLY ON XPATR
18    1 *D2            FIRST USE TO CAUSE INITIALIZATION.                     XPATR
19    1 *D2   INTLNG - LENGTH OF INTBUF                                       XPATR
20    1 *D2   MRBUFF - MANAGER REQUEST BUFFER (64 WORDS) USED TO COMMUNICATE  XPATR
21    1 *D2            WITH THE FDS MANAGER.  MAY BE USED AS A SCRATCH AREA BY XPATR
22    1 *D2            THE PROCESSOR EXCEPT ACCROSS PROCESSOR SERVICE CALLS.  XPATR
23    1 *D2   INUM   - RELATIVE NUMBER OF PARAMETER IN INTERFACE TABLE WHOSE  XPATR
24    1 *D2            ATTRIBUTES ARE REQUESTED.                              XPATR
25    1 *D2                                                                   XPATR
26    1 **********                                                            XPATR
27    1 *D3      OUTPUT                                                       XPATR
28    1 *D3   NAME   - AN ALPHANUMERIC NAME OF UP TO SIX CHARACTERS WHICH     XPATR
29    1 *D3            IDENTIFIES THE DATA ELEMENT OR DRDE WHERE THE DATA IS TO XPATR
30    1 *D3            BE OBTAINED/STORED. A ZERO ENTRY INDICATES INPUT DATA  XPATR
31    1 *D3            WHICH IS LITERAL DATA STORED WITHIN THE INTERFACE TABLE. XPATR
32    1 *D3   TYPE   - DATA TYPE CODE OF THE PARAMETER                        XPATR
33    1 *D3   SIZE   - TOTAL NUMBER OF WORDS OF LITERAL DATA,                 XPATR
34    1 *D3            TOTAL NUMBER OF WORDS OF REFERENCED INPUT DE,          XPATR
35    1 *D3            TOTAL NUMBER OF WORDS OF REFERENCED OUTPUT DE IF       XPATR
36    1 *D3            SUBSCRIPTED OR ZERO IF NOT SUBSCRIPTED,                XPATR
37    1 *D3            TOTAL NUMBER OF BLOCKS OF REFERENCED INPUT DRDE OR     XPATR
38    1 *D3            ZERO IF OUTPUT DRDE                                    XPATR
39    1 *D3   IDIM   - COLUMN LENGTH OF A DATA ELEMENT, MAXIMUM RECORD SIZE OF XPATR
40    1 *D3            AN INPUT DRDE OR THE LENGTH OF A SYMBOLIC STRING.  ZERO XPATR
41    1 *D3            IF AN UNSUBSCRIPTED OUTPUT.                            XPATR
42    1 *D3   DSPTY  - DISPLACEMENT FROM THE BEGINNING OF THE DATA FOR        XPATR
43    1 *D3            SUBSCRIPTED DATA ELEMENTS ELSE ZERO.                   XPATR
44    1 *D3            FOR AN INPUT DRDE THE RTE FILE MANAGER TYPE CODE IS    XPATR
45    1 *D3            RETURNED IN DSPTY.                                     XPATR
46    1 *D3                                                                   XPATR
47    1 *D3      EXTERNAL SYMBOLS                                             XPATR
48    1 *D3   (SEE XPGET)                                                     XPATR
49    1 *D3                                                                   XPATR
50    1 **********                                                            XPATR
51    1 *D4      INTERNAL VARIABLES                                           XPATR
52    1 *D4                                                                   XPATR
53    1 **********                                                            XPATR
54    1 *D5      NOTES                                                        XPATR
55    1 *D5                                                                   XPATR
56    1 *D5   USES .ENTR, XPE13(XPGET), XPINI(XPGET), XPOFN(XPGET), XPREQ,    XPATR
57    1 *D5        XPSBC(XPGET), XPXIT, XVSTB                                 XPATR
58    1 *D5                                                                   XPATR
59    1 *D5   XPATR IS IMPLEMENTED AS A SINGLE MODULE CONTAINING THE ENTRY    XPATR
60    1 *D5   POINTS XPATR, XPGET AND XPPUT                                   XPATR
```

```
61    1 *D5
62    1 *D5   THE SAME INTBUF MUST BE USED BY XPGET, XPPUT, AND XPATR AND NEEDS
63    1 *D5   TO BE INITIALIZED ONLY ONCE BY ANY OF THE THREE ROUTINES.
64    1 *D5
65    1 **********
```

XPATR
XPATR
XPATR
XPATR
XPATR

```
1 BEGIN XPATR                                                          XPATR
  2   PERFORM XPINI(XPGET) TO INITIALIZE GLOBALS AND INTERFACE TABLE   XPATR
  2 EXIT TO :XPE13: (XPGET) IF PARAMETER IS OUT OF RANGE               XPATR
  2     EXTRACT NAME FROM INTERFACE TABLE                              XPATR
  2     SET DSPTY TO ZERO                                              XPATR
  2     IF LITERAL PARAMETER (NAME IS ZERO)                            XPATR
  2     THEN                                                           XPATR
  3       COPY TYPE, SIZE AND IDIM FROM INTERFACE TABLE                XPATR
  3     ELSE                                                           XPATR
  3       IF SUBSCRIPTED (INTERFACE TABLE DISP OR S FIELDS ARE NON-ZERO) XPATR
  3       THEN                                                         XPATR
  4         PERFORM XPSBC(XPGET) TO RETRIEVE TOC ENTRY AND COMPUTE DISPLACEMENT XPATR
  4         STORE TYPE, SIZE, IDIM AND DSPTY                           XPATR
  3       ELSE                                                         XPATR
  4         IF DRDE                                                    XPATR
  4         THEN                                                       XPATR
  5           PERFORM XPQFN(XPGET) TO QUALIFY FILE NAME                XPATR
  4         ENDIF                                                      XPATR
  4         IF INPUT PARAMETER                                         XPATR
  4         THEN                                                       XPATR
  5           CALL XPREQ TO RETRIEVE TOC ENTRY                         XPATR
  5           COPY TYPE, SIZE, IDIM AND DSPTY FROM TOC ENTRY           XPATR
  4         ELSE                                                       XPATR
  5           SET TYPE, SIZE AND IDIM TO ZERO                          XPATR
  4         ENDIF                                                      XPATR
  3       ENDIF                                                        XPATR
  2     ENDIF                                                          XPATR
  1 END XPATR                                                          XPATR
```

```
 96    1 *D0      FORTRAN CALLING PROCEDURE                                                    XPGET
 97    1 *D0                                                                                   XPGET
 98    1 *D0  CALL XPGET (LU, INTBUF, INTLNG, MREUFF, N, INUMS, IN(1), ... ,                  XPGET
 99    1 *D0                IN(N))                                                             XPGET
100    1 *D0                                                                                   XPGET
101    1 **********                                                                            XPGET
102    1 *D1      XPGET ALLOWS PROCESSORS TO OBTAIN DATA FROM THE FDS AWA. IT                  XPGET
103    1 *D1      VERIFIES THE EXISTENCE OF THE DATA ELEMENTS AND DRDE'S                       XPGET
104    1 *D1      REFERENCED BY THE INTERFACE TABLE AND RETRIEVES INPUT DATA                   XPGET
105    1 *D1      BY ONE OF THE FOLLOWING WAYS:                                                XPGET
106    1 *D1           1) REQUESTED DATA FROM LITERAL DATA STORED WITHIN THE                   XPGET
107    1 *D1              INTERFACE TABLE IS RETURNED OR                                       XPGET
108    1 *D1           2) REQUESTED DATA FROM A DATA ELEMENT REFERENCED BY                     XPGET
109    1 *D1              THE INTERFACE TABLE IS RETURNED OR                                   XPGET
110    1 *D1           3) THE NAME OF THE DRDE REFERENCED BY THE INTERFACE                     XPGET
111    1 *D1              TABLE WITH A / SYMBOL AS A PREFIX AND A ONE CHARACTER                XPGET
112    1 *D1              USER CODE AS A SUFFIX IS RETURNED SO THE CALLING                     XPGET
113    1 *D1              PROGRAM MAY RETRIEVE THE DATA.                                       XPGET
114    1 *D1      DATA ELEMENT AND DRDE NAMES ARE SEARCHED FOR IN THE AWA TOC,                 XPGET
115    1 *D1      IF NOT FOUND, A MESSAGE IS ISSUED AND PROCESSING TERMINATED.                 XPGET
116    1 *D1                                                                                   XPGET
117    1 **********                                                                            XPGET
118    1 *D2      INPUT                                                                        XPGET
119    1 *D2  LU      - LOGICAL UNIT NUMBER OF USER TERMINAL                                   XPGET
120    1 *D2  INTBUF  - INPUT/OUTPUT BUFFER OF 7+(# PARAMETERS + 1) WORDS,                     XPGET
121    1 *D2            ALLOCATED WITHIN THE CALLING PROGRAM TO HOLD THE                       XPGET
122    1 *D2            INTERFACE TABLE HEADER. FIRST WORD MUST BE ZERO ONLY ON                XPGET
123    1 *D2            FIRST USE TO CAUSE INITIALIZATION.                                     XPGET
124    1 *D2  INTLNG  - LENGTH OF INTBUF                                                       XPGET
125    1 *D2  MREUFF  - MANAGER REQUEST BUFFER (64 WORDS) USED TO COMMUNICATE                  XPGET
126    1 *D2            WITH THE MANAGER. BUFFER MAY BE USED AS PROGRAM SCRATCH               XPGET
127    1 *D2            AREA BUT NOT ACROSS PROCESSOR SERVICE CALLS.                          XPGET
128    1 *D2  N       - NUMBER OF PARAMETERS TO BE RETRIEVED. IF N=0, ALL                     XPGET
129    1 *D2            PARAMETERS ARE REQUESTED AND INUMS IS NOT USED.                       XPGET
130    1 *D2  INUMS   - ARRAY (N WORDS) OF RELATIVE NUMBERS OF PARAMETERS IN THE              XPGET
131    1 *D2            INTERFACE TABLE TO BE RETRIEVED. THE ORDER OF THE                     XPGET
132    1 *D2            NUMBERS INDICATES THE STORAGE ORDER FOR THE IN(I)                     XPGET
133    1 *D2                                                                                  XPGET
134    1 *D2      SHARED EXTERNAL SYMBOL                                                       XPGET
135    1 *D2  XPGPF   - FLAG INDICATING XPGET/XPPUT NORMAL PROCESSING (-1) OR                 XPGET
136    1 *D2            XPGTI/XPPTI SPECIAL PROCESSING BY-PASSING SUBSCRIPT                   XPGET
137    1 *D2            RESOLUTION (>-1)                                                      XPGET
138    1 *D2                                                                                  XPGET
139    1 **********                                                                           XPGET
140    1 *D3      OUTPUT                                                                      XPGET
141    1 *D3  IN(I)   - LABELS OR VARIABLE NAMES WHERE INPUT DATA IS TO BE                    XPGET
142    1 *D3            STORED.                                                               XPGET
143    1 *D3                                                                                  XPGET
144    1 **********                                                                           XPGET
145    1 *D4      INTERNAL VARIABLES                                                          XPGET
146    1 *D4  AMULT   - ARRAY OF 10 BYTES CONTAINING MULTIPLIERS FOR USE BY THE               XPGET
147    1 *D4            PROCEDURE SUBCMP IN DETERMINING WORD DISPLACEMENTS FOR               XPGET
148    1 *D4            VARIOUS TYPES OF SUBSCRIPTED ARRAYS                                  XPGET
149    1 *D4  CLSFL   - ADDRESS OF THE CLASS FIELD (SECOND WORD) OF THE REQUEST              XPGET
150    1 *D4            ARRAY REQST                                                          XPGET
151    1 *D4  STPT    - FLAG INDICATING GET OR PUT PROCESSING.  VALUES CORRESPOND           XPGET
152    1 *D4            TO BIT SETTINGS FOR INPUT OR OUTPUT FLAGS IN THE FIFTH               XPGET
153    1 *D4            WORD OF INTERFACE TABLE ENTRIES                                      XPGET
154    1 *D4  I       - COUNTER OF THE NUMBER OF PASSES THROUGH THE PARAMETER               XPGET
```

```
155    1 *D4                PROCESSING LOOP                                                    XPGET
156    1 *D4          NP    - VALUE OF -P, WHERE P IS THE NUMBER OF PARAMETERS IN THE          XPGET
157    1 *D4                INTERFACE TABLE                                                    XPGET
158    1 *D4          NAMFL - ADDRESS OF THE NAME FIELD (THIRD WORD) OF THE REQUEST            XPGET
159    1 *D4                ARRAY REQST                                                        XPGET
160    1 *D4          NARY  - NEGATIVE OF THE NUMBER OF IN/OUT ARRAYS IN CALLING             XPGET
161    1 *D4                SEQUENCE                                                           XPGET
162    1 *D4          REQST - EIGHT WORD ARRAY USED FOR CONSTRUCTING AWA MANAGEMENT          XPGET
163    1 *D4                REQUESTS FOR XPREQ                                                 XPGET
164    1 *D4          USRID - USER FILE IDENTIFIER CHARACTER (=IPARM(3))                     XPGET
165    1 *D4                                                                                   XPGET
166    1 *D4             EXTERNAL VARIABLES (SEE XPREQ)                                       XPGET
167    1 *D4          XPCLS                                                                   XPGET
168    1 *D4          XPLU                                                                    XPGET
169    1 *D4          XPRQQ                                                                   XPGET
170    1 *D4                                                                                   XPGET
171    1 **********                                                                           XPGET
172    1 *D5             NOTES                                                                XPGET
173    1 *D5                                                                                   XPGET
174    1 *D5          USES .ENTR, EXEC, XPREQ, XPXIT, XVSTB                                   XPGET
175    1 *D5                                                                                   XPGET
176    1 *D5          XPGET MUST BE INCLUDED IN PROCESSOR AT FDS BUILD TIME.                  XPGET
177    1 *D5          SAME INTBUF MUST BE USED BY XPGET, XPPUT, AND XPATR AND NEEDS           XPGET
178    1 *D5          TO BE INITIALIZED ONLY ONCE BY ANY OF THE THREE ROUTINES                XPGET
179    1 *D5          SINCE REQUESTS FOR INPUT DATA FROM THE AWA MAY BE MADE FOR UP TO        XPGET
180    1 *D5          EIGHT AT A TIME, XPGET RUNS MOST EFFICIENTLY WHEN PARAMETERS ARE        XPGET
181    1 *D5          REQUESTED IN MULTIPLES OF EIGHT.                                        XPGET
182    1 *D5                                                                                   XPGET
183    1 **********                                                                           XPGET
```

```
185    1 *D0      FORTRAN CALLING PROCEDURE                                    XPGET
186    1 *D0                                                                   XPGET
187    1 *D0   CALL XPPUT (LU, INTBUF, INTLNG, MRBUFF, N, INUMS, OUT(1), ... ,XPGET
188    1 *D0               OUT(N))                                             XPGET
189    1 *D0                                                                   XPGET
190    1 **********                                                            XPGET
191    1 *D1      XPPUT ALLOWS PROCESSORS TO STORE DATA INTO DATA ELEMENTS AND  XPGET
192    1 *D1      DRDE'S REFERENCED IN THE INTERFACE TABLE AS FOLLOWS:         XPGET
193    1 *D1            1) DATA ELEMENT NAME IS SEARCHED FOR IN THE TOC,       XPGET
194    1 *D1               IF FOUND, DATA IS STORED                            XPGET
195    1 *D1               IF NOT FOUND, A MESSAGE IS ISSUED AND PROCESSING    XPGET
196    1 *D1               TERMINATES.                                         XPGET
197    1 *D1            2) DRDE NAME IS SEARCHED FOR IN THE TOC. IF THE NAME IS XPGET
198    1 *D1               NOT FOUND, IT IS ADDED TO THE TOC. IF THE NAME IS   XPGET
199    1 *D1               FOUND OR AFTER BEING ADDED TO THE TOC, THE NAME IS  XPGET
200    1 *D1               PREFIXED WITH A / SYMBOL, SUFFIXED WITH A ONE       XPGET
201    1 *D1               CHARACTER USER CODE AND RETURNED SO THE CALLING     XPGET
202    1 *D1               PROGRAM CAN STORE DATA.                             XPGET
203    1 *D1                                                                   XPGET
204    1 **********                                                            XPGET
205    1 *D2      INPUT                                                        XPGET
206    1 *D2   LU      - LOGICAL UNIT NUMBER OF USER TERMINAL                  XPGET
207    1 *D2   INTBUF  - INPUT/OUTPUT BUFFER OF 7*(# PARAMETERS + 1) WORDS,    XPGET
208    1 *D2             ALLOCATED WITHIN THE CALLING PROGRAM TO HOLD THE      XPGET
209    1 *D2             INTERFACE TABLE HEADER. FIRST WORD MUST BE ZERO ONLY ON XPGET
210    1 *D2             FIRST USE TO CAUSE INITIALIZATION.                    XPGET
211    1 *D2   INTLNG  - LENGTH OF INTBUF                                      XPGET
212    1 *D2   MRBUFF  - MANAGER REQUEST BUFFER (64 WORDS) USED TO COMMUNICATE XPGET
213    1 *D2             WITH THE MANAGER. BUFFER MAY BE USED AS PROGRAM SCRATCHXPGET
214    1 *D2             AREA BUT NOT ACROSS PROCESSOR SERVICE CALLS.          XPGET
215    1 *D2   N       - NUMBER OF PARAMETERS TO BE STORED. IF N=0, ALL       XPGET
216    1 *D2             OUTPUT PARAMETERS ARE ASSUMED TO BE SUPPLIED.         XPGET
217    1 *D2   INUMS   - ARRAY (N WORDS) OF RELATIVE NUMBERS OF PARAMETERS IN THEXPGET
218    1 *D2             INTERFACE TABLE REFERENCING DATA ELEMENTS AND DRDE'S   XPGET
219    1 *D2             WHERE THE OUTPUT DATA IS TO BE STORED. THE ORDER OF THEXPGET
220    1 *D2             NUMBERS INDICATES THE STORAGE ORDER FROM OUT(1),..OUT(N)XPGET
221    1 *D2   OUT(N)  - LABELS OR VARIABLE NAMES WHERE OUTPUT DATA IS TO BE   XPGET
222    1 *D2             STORED FROM.                                          XPGET
223    1 *D2                                                                   XPGET
224    1 *D2      SHARED EXTERNAL SYMBOL                                       XPGET
225    1 *D2   XPGPF   - FLAG INDICATING XPGET/XPPUT NORMAL PROCESSING (-1) OR XPGET
226    1 *D2             XPGTI/XPPTI SPECIAL PROCESSING BY-PASSING SUBSCRIPT    XPGET
227    1 *D2             RESOLUTION (>-1)                                      XPGET
228    1 *D2                                                                   XPGET
229    1 **********                                                            XPGET
230    1 *D3      OUTPUT                                                       XPGET
231    1 *D3   NONE                                                            XPGET
232    1 *D3                                                                   XPGET
233    1 **********                                                            XPGET
234    1 *D4      INTERNAL VARIABLES                                           XPGET
235    1 *D4   (SEE XPGET)                                                     XPGET
236    1 *D4                                                                   XPGET
237    1 **********                                                            XPGET
238    1 *D5      NOTES                                                        XPGET
239    1 *D5   (SEE XPGET)                                                     XPGET
240    1 *D5                                                                   XPGET
241    1 **********                                                            XPGET
```

```
243   1 BEGIN XPGET                                                                                                    XPGET
244   2    SET FOR 'GET'                                                                                               XPGET
245   2    PERFORM ACCESS TO RETRIEVE DATA                                                                             XPGET
246   1 END XPGET                                                                                                      XPGET
247   1 BEGIN XPPUT                                                                                                    XPGET
248   2    SET FOR 'PUT'                                                                                               XPGET
249   2    PERFORM ACCESS TO STORE DATA                                                                                XPGET
250   1 END XPPUT                                                                                                      XPGET
251   1 BEGIN ACCESS                                                                                                   XPGET
252   2    PERFORM XPINI TO INITIALIZE GLOBALS AND INTERFACE TABLE                                                      XPGET
253   2    DO FOR EACH PARAMETER REQUESTED                                                                             XPGET
254   3       IF SELECTED PARAMETER IS OUT OF RANGE                                                                    XPGET
255   3       THEN                                                                                                     XPGET
256   4          CALL XPREQ TO PURGE QUEUED REQUESTS                                                                   XPGET
257   3       EXIT TO :XPE13:                                                                                          XPGET
258   3       ENDIF                                                                                                    XPGET
259   3       IF INPUT/OUTPUT TYPE DOES NOT MATCH 'GET'/'PUT' PROCESSING                                               XPGET
260   3       THEN                                                                                                     XPGET
261   4          CALL XPREQ TO PURGE QUEUED REQUESTS                                                                   XPGET
262   3       EXIT TO :ERR12:                                                                                          XPGET
263   3       ENDIF                                                                                                    XPGET
264   3       IF OVER RUNNING CALLING SEQUENCE                                                                         XPGET
265   3       THEN                                                                                                     XPGET
266   4          CALL XPREQ TO PURGE QUEUED REQUESTS                                                                   XPGET
267   3       EXIT TO :XPE13:                                                                                          XPGET
268   3       ENDIF                                                                                                    XPGET
269   3       IF PROCESSING FOR 'GET'                                                                                  XPGET
270   3       THEN                                                                                                     XPGET
271   4          IF PARAMETER IS MEMORY RESIDENT DE (CLASS 2)                                                          XPGET
272   4          THEN                                                                                                  XPGET
273   5             IF INPUT IN LITERAL FORM                                                                           XPGET
274   5             THEN                                                                                               XPGET
275   6                BUILD REQUEST WITH INTERFACE TABLE NAME AND DISPLACEMENT                                         XPGET
276   5             ELSE                                                                                               XPGET
277   6                IF NORMAL XPGET/PUT PROCESSING (XPGPF = -1)                                                     XPGET
278   6                THEN                                                                                           XPGET
279   7                   IF SUBSCRIPTED (DISPLACEMENT > 0 OR DOUBLE SUBSCRIPT FLAG SET)                              XPGET
280   7                   THEN                                                                                        XPGET
281   8                      PERFORM XPSBC TO COMPUTE DISPLACEMENT = F(DIM, SUBS, TYPE)                               XPGET
282   7                   ELSE                                                                                        XPGET
283   8                      DISPLACEMENT IS ZERO                                                                     XPGET
284   7                   ENDIF                                                                                       XPGET
285   6                ELSE                                                                                           XPGET
286   7                   USE DISPLACEMENT FROM INTERFACE TABLE ENTRY                                                 XPGET
287   6                ENDIF                                                                                          XPGET
288   6                BUILD REQUEST WITH DE NAME AND DISPLACEMENT                                                    XPGET
289   5             ENDIF                                                                                             XPGET
290   5             CALL XPREQ TO QUEUE RETRIEVAL AND STORAGE OF INPUT                                                XPGET
291   4          ELSE PARAMETER IS DRDE (CLASS 3)                                                                     XPGET
292   5             BUILD REQUEST FOR TOC ENTRY                                                                       XPGET
293   5             CALL XPREQ TO IMMEDIATELY RETRIEVE TOC ENTRY                                                      XPGET
294   5             PERFORM XPQFN TO CONSTRUCT AND STORE QUALIFIED FILE NAME                                          XPGET
295   5             STORE FILE ATTRIBUTES                                                                             XPGET
296   4          ENDIF                                                                                                XPGET
297   3       ELSE PROCESSING FOR 'PUT'                                                                               XPGET
298   4          IF PARAMETER IS MEMORY RESIDENT DE (CLASS 2)                                                          XPGET
299   4          THEN                                                                                                  XPGET
300   5             IF NORMAL XPGET/PUT PROCESSING (XPGPF = -1)                                                       XPGET
301   5             THEN                                                                                              XPGET
```

```
302  6              IF SUBSCRIPTED                                                                    XPGET
303  6              THEN                                                                              XPGET
304  7                 PERFORM XPSBC TO COMPUTE DISPLACEMENT = F(DIM, SUBS, TYPE)                     XPGET
305  6              ELSE                                                                              XPGET
306  7                 DISPLACEMENT IS ZERO                                                           XPGET
307  7                 BUILD REQUEST TO DELETE ANY EXISTING DE WITH THIS NAME                         XPGET
308  7                 CALL XPREQ TO QUEUE DELETION                                                   XPGET
309  7                 BUILD REQUEST TO REALLOCATE DE                                                 XPGET
310  7                 CALL XPREQ TO QUEUE ALLOCATION                                                 XPGET
311  6              ENDIF                                                                             XPGET
312  5           ELSE                                                                                 XPGET
313  6              USE DISPLACEMENT FROM INTERFACE TABLE ENTRY                                       XPGET
314  5           ENDIF                                                                                XPGET
315  5           BUILD REQUEST TO OUTPUT DATA TO CLASS I/O AND STORE DATA IN AWA                       XPGET
316  5           CALL XPREQ TO QUEUE STORAGE OF DATA                                                  XPGET
317  4        ELSE PARAMETER IS DRDE (CLASS 3)                                                        XPGET
318  5           BUILD REQUEST TO DELETE ANY EXISTING DRDE WITH THIS NAME                           XPGET
319  5           CALL XPREQ TO QUEUE DELETION                                                       XPGET
320  5           BUILD REQUEST TO REALLOCATE DRDE WITH NEW ATTRIBUTES                               XPGET
321  4        EXIT TO :ERR15: IF FILE TYPE NOT 1-15, # BLOCKS < 1 OR MAX REC SIZE NOT 1-1200        XPGET
322  5           CALL XPREQ TO QUEUE REALLOCATION OF DRDE                                           XPGET
323  4        ENDIF                                                                                 XPGET
324  3     ENDIF                                                                                    XPGET
325  2  ENDDO                                                                                       XPGET
326  2  CALL XPREQ TO COMPLETE QUEUED REQUESTS                                                      XPGET
327  1 EXIT ACCESS                                                                                  XPGET

328  2    :ERR12: TERMINATE PROCESSOR FOR INPUT/OUTPUT TYPE INCONSISTENCY                           XPGET

329  2    :XPE13: TERMINATE PROCESSOR FOR INVALID PARAMETER REQUEST                                 XPGET

330  2    :ERR14: CALL XPREQ TO PURGE QUEUED REQUESTS                                               XPGET
331  2    TERMINATE PROCESSOR FOR INVALID DRDE FILE TYPE, BLOCK COUNT OR MAX RECORD SIZE            XPGET
```

```
333   2     BEGIN XPINI                                                                                                    XPGET
334   3        INITIALIZE GLOBAL VALUES FROM LU AND XVSTB                                                                  XPGET
335   3        TERMINATE PROCESSOR WITH XP10 ERROR IF LU NOT IN XVSTB                                                      XPGET
336   3        IF INTERFACE TABLE BUFFER NOT INITIALIZED                                                                   XPGET
337   3        THEN                                                                                                        XPGET
338   4           RETRIEVE INTERFACE TABLE FROM MANAGER CLASS I/O NUMBER                                                   XPGET
339   4           IF RETRIEVAL NOT SUCCESSFUL                                                                              XPGET
340   4           THEN                                                                                                     XPGET
341   5              TERMINATE PROCESSOR WITH 'XP10 PROCESSOR INITILAIZATION ERROR'                                        XPGET
342   4           ENDIF                                                                                                    XPGET
343   3        ENDIF                                                                                                       XPGET
344   2     EXIT TO :XPE13: IF N < 0                                                                                       XPGET
345   2     END XPINI                                                                                                      XPGET
346   2 *                                                                                                                  XPGET
347   2 *                                                                                                                  XPGET
348   2 *                                                                                                                  XPGET
349   2 *                                                                                                                  XPGET
350   2     BEGIN XPQFN                                                                                                    XPGET
351   3        SET / IN FIRST CHARACTER POSITION                                                                           XPGET
352   3        MOVE FOUR WORD NAME INTO MIDDLE POSITIONS                                                                   XPGET
353   3        LOCATE FIRST BLANK CHARACTER                                                                                XPGET
354   3        REPLACE BLANK WITH USER ID                                                                                  XPGET
355   2     END XPQFN                                                                                                      XPGET
356   2 *                                                                                                                  XPGET
357   2 *                                                                                                                  XPGET
358   2 *                                                                                                                  XPGET
359   2 *                                                                                                                  XPGET
360   2     BEGIN XPS9C                                                                                                    XPGET
361   3        IF DOUBLE SUBSCRIPTED                                                                                       XPGET
362   3        THEN                                                                                                        XPGET
363   4           BUILD REQUEST FOR SUBSCRIPTS STORED IN LITERAL BLOCK                                                     XPGET
364   4           CALL XPREQ TO QUEUE RETURN OF SUBSCRIPTS                                                                 XPGET
365   3        ELSE                                                                                                        XPGET
366   4           SINGLE SUBSCRIPT IS CONTAINED IN DISPLACEMENT FIELD & J-SUBSCRIPT IS = 1                                 XPGET
367   3        ENDIF                                                                                                       XPGET
368   3        BUILD REQUEST TO RETURN TOC ENTRY                                                                           XPGET
369   3        CALL XPREQ TO IMMEDIATELY RETRIEVE TOC ENTRY                                                                XPGET
370   3        IF DOUBLE DIMENSIONED (IDIM > 0)                                                                            XPGET
371   3        THEN                                                                                                        XPGET
372   4           DISPLACEMENT = (IDIM*(JSUB-1) + ISUB-1) * # WORDS PER ELEMENT                                            XPGET
373   3        ELSE                                                                                                        XPGET
374   4           DISPLACEMENT = SIZE*(JSUB-1) + (ISUB-1) * # WORDS PER ELEMENT                                            XPGET
375   3        ENDIF                                                                                                       XPGET
376   2     END XPS9C                                                                                                      XPGET
377   1  END ACCESS                                                                                                        XPGET
```

```
379   1 *D0      FORTRAN CALLING PROCEDURE                                                    XPGTI
380   1 *D0                                                                                   XPGTI
381   1 *D0   CALL XPGTI (LU, INTBUF, INTLNG, MRBUFF, INUM, IN, SIZE, DISP)                   XPGTI
382   1 *D0                                                                                   XPGTI
383   1 **********                                                                            XPGTI
384   1 *D1                                                                                   XPGTI
385   1 *D1      XPGTI ALLOWS THE CALLING PROGRAM TO RETRIEVE A SPECIFIED AMOUNT              XPGTI
386   1 *D1      OF DATA FROM AN INPUT PARAMETER BEGINNING AT ANY POINT IN THE                XPGTI
387   1 *D1      PARAMETER ARRAY                                                              XPGTI
388   1 *D1                                                                                   XPGTI
389   1 **********                                                                            XPGTI
390   1 *D2      INPUT                                                                        XPGTI
391   1 *D2   LU      - LOGICAL UNIT NUMBER OF USER TERMINAL                                  XPGTI
392   1 *D2   INTBUF  - INPUT/OUTPUT BUFFER OF 7*(# PARAMETERS + 1) WORDS,                    XPGTI
393   1 *D2             ALLOCATED WITHIN THE CALLING PROGRAM TO HOLD THE                      XPGTI
394   1 *D2             INTERFACE TABLE HEADER.  FIRST WORD MUST BE ZERO ONLY ON              XPGTI
395   1 *D2             FIRST USE TO CAUSE INITIALIZATION                                     XPGTI
396   1 *D2   INTLNG  - LENGTH OF INTBUF                                                      XPGTI
397   1 *D2   MRBUFF  - MANAGER REQUEST BUFFER (64 WORDS) USED TO COMMUNICATE                 XPGTI
398   1 *D2             WITH THE FDS MANAGER.  MAY BE USED AS A SCRATCH AREA BY             XPGTI
399   1 *D2             THE PROCESSOR EXCEPT ACROSS PROCESSOR SERVICE CALLS                   XPGTI
400   1 *D2   INUM    - RELATIVE NUMBER OF PARAMETER IN INTERFACE TABLE FROM                  XPGTI
401   1 *D2             WHICH TO RETURN DATA                                                  XPGTI
402   1 *D2   SIZE    - TOTAL NUMBER OF WORDS TO BE RETURNED                                  XPGTI
403   1 *D2   DISP    - DISPLACEMENT FROM THE BEGINNING OF THE INPUT PARAMETER                XPGTI
404   1 *D2             AT WHICH TO BEGIN DATA RETRIEVAL.  A VALUE OF ZERO                    XPGTI
405   1 *D2             INDICATES THE BEGINNING OF THE AREA SPECIFIED BY THE USER            XPGTI
406   1 *D2             IN THE INTERFACE TABLE.  NEGATIVE VALUES OF DISP MUST NOT            XPGTI
407   1 *D2             BE SPECIFIED, I.E., RETRIEVAL FROM AN AREA PREVIOUS TO               XPGTI
408   1 *D2             THE USERS SUBSCRIPTS IS NOT SUPPORTED                                 XPGTI
409   1 *D2                                                                                   XPGTI
410   1 **********                                                                            XPGTI
411   1 *D3      OUTPUT                                                                       XPGTI
412   1 *D3   IN      - ARRAY OF AT LEAST SIZE WORDS INTO WHICH INPUT DATA IS TO             XPGTI
413   1 *D3             BE STORED                                                             XPGTI
414   1 *D3                                                                                   XPGTI
415   1 **********                                                                            XPGTI
416   1 *D4      INTERNAL                                                                     XPGTI
417   1 *D4   BSDSP   - DISPLACEMENT FROM THE REAL ORIGIN OF THE DATA ELEMENT AS             XPGTI
418   1 *D4             A RESULT OF USER SUBSCRIPTING.  DISP IS ADDED TO THIS               XPGTI
419   1 *D4             VALUE IN THE REQUEST TO THE MANAGER                                   XPGTI
420   1 *D4   GTPT    - DISPLACEMENT INTO JGTPT DEPENDING ON WHETHER GET (0) OR              XPGTI
421   1 *D4             PUT (1) PROCESSING IS INVOLVED                                        XPGTI
422   1 *D4   JGTPT   - TWO WORD VECTOR CONTAING 'JSB XPGET' OR 'JSB XPPUT'                  XPGTI
423   1 *D4             INSTRUCTIONS.  USED TO DYNAMICALLY PRODUCE XPGET AND                XPGTI
424   1 *D4             XPPUT CALLS                                                           XPGTI
425   1 *D4                                                                                   XPGTI
426   1 **********                                                                            XPGTI
427   1 *D5      NOTES                                                                        XPGTI
428   1 *D5   USES EXEC, .ENTR, XPATR, XPE13(XPATR), XPGET(XPATR), XPPUT(XPATR)              XPGTI
429   1 *D5   USES .ENTR, EXEC, XPATR, XPERR(XPATR), XPE13(XPATR), XPGET(XPATR),             XPGTI
430   1 *D5       XPPUT(XPATR)                                                              XPGTI
431   1 **********                                                                            XPGTI
```

```
433   1 *D0      FORTRAN CALLING PROCEDURE                                                    XPGTI
434   1 *D0                                                                                   XPGTI
435   1 *D0   CALL XPPTI (LU, INTBUF, INTLNG, MRBUFF, INUM, OUT, SIZE, DISP)                  XPGTI
436   1 *D0                                                                                   XPGTI
437   1 **********                                                                            XPGTI
438   1 *D1                                                                                   XPGTI
439   1 *D1      XPPTI ALLOWS THE CALLING PROGRAM TO STORE A SPECIFIED AMOUNT OF              XPGTI
440   1 *D1      DATA INTO AN OUTPUT PARAMETER BEGINNING AT ANY POINT IN THE                  XPGTI
441   1 *D1      PARAMETER ARRAY                                                              XPGTI
442   1 *D1                                                                                   XPGTI
443   1 **********                                                                            XPGTI
444   1 *D2      INPUT                                                                        XPGTI
445   1 *D2   LU     - LOGICAL UNIT NUMBER OF USER TERMINAL                                   XPGTI
446   1 *D2   INTBUF - INPUT/OUTPUT BUFFER OF 7*(# PARAMETERS + 1) WORDS,                     XPGTI
447   1 *D2            ALLOCATED WITHIN THE CALLING PROGRAM TO HOLD THE                       XPGTI
448   1 *D2            INTERFACE TABLE HEADER.  FIRST WORD MUST BE ZERO ONLY ON               XPGTI
449   1 *D2            FIRST USE TO CAUSE INITIALIZATION                                      XPGTI
450   1 *D2   INTLNG - LENGTH OF INTBUF                                                       XPGTI
451   1 *D2   MRBUFF - MANAGER REQUEST BUFFER (64 WORDS) USED TO COMMUNICATE                  XPGTI
452   1 *D2            WITH THE FDS MANAGER.  MAY BE USED AS A SCRATCH AREA BY               XPGTI
453   1 *D2            THE PROCESSOR EXCEPT ACROSS PROCESSOR SERVICE CALLS                   XPGTI
454   1 *D2   INUM   - RELATIVE NUMBER OF PARAMETER IN INTERFACE TABLE INTO                   XPGTI
455   1 *D2            WHICH TO STORE DATA                                                   XPGTI
456   1 *D2   OUT    - ARRAY OF AT LEAST SIZE WORDS FROM WHICH OUTPUT DATA IS TO            XPGTI
457   1 *D2            BE TAKEN                                                              XPGTI
458   1 *D2   SIZE   - TOTAL NUMBER OF WORDS TO BE STORED                                    XPGTI
459   1 *D2   DISP   - DISPLACEMENT FROM THE BEGINNING OF THE OUTPUT PARAMETER               XPGTI
460   1 *D2            AT WHICH TO BEGIN DATA STORAGE.  A VALUE OF ZERO                      XPGTI
461   1 *D2            INDICATES THE BEGINNING OF THE AREA SPECIFIED BY THE USER             XPGTI
462   1 *D2            IN THE INTERFACE TABLE.  NEGATIVE VALUES OF DISP MUST NOT             XPGTI
463   1 *D2            BE SPECIFIED, I.E., STORAGE INTO AN AREA PREVIOUS TO THE              XPGTI
464   1 *D2            USERS SUESCRIPTS IS NOT SUPPORTED.                                    XPGTI
465   1 *D2                                                                                  XPGTI
466   1 **********                                                                           XPGTI
467   1 *D3      OUTPUT                                                                      XPGTI
468   1 *D3   NONE                                                                           XPGTI
469   1 *D3                                                                                  XPGTI
470   1 **********                                                                           XPGTI
471   1 *D4      INTERNAL                                                                    XPGTI
472   1 *D4   SEE XPGTI                                                                      XPGTI
473   1 *D4                                                                                  XPGTI
474   1 **********                                                                           XPGTI
475   1 *D5      NOTES                                                                       XPGTI
476   1 *D5   XPPTI IS AN ENTRY POINT INTO XPGTI                                             XPGTI
477   1 *D5                                                                                  XPGTI
478   1 **********                                                                           XPGTI
```

```
480   1 BEGIN XPGTI                                                                                                    XPGTI
481   2    SET FOR GET PROCESSING                                                                                      XPGTI
482   2    PERFORM XPXXI TO RETRIEVE DATA                                                                              XPGTI
483   1 END XPGTI                                                                                                      XPGTI
484   1 BEGIN XPPTI                                                                                                    XPGTI
485   2    SET FOR PUT PROCESSING                                                                                      XPGTI
486   2    PERFORM XPXXI TO STORE DATA                                                                                 XPGTI
487   1 END XPPTI                                                                                                      XPGTI
488   1 BEGIN XPXXI                                                                                                    XPGTI
489   2    SET XPGPF(XPGET) TO BY-PASS NORMAL XPGET/XPPUT SUBSCRIPT PROCESSING                                          XPGTI
490   2    CALL XPATR TO ASSURE INITIALIZATION OF INTBUF AND RETURN BASE DISPLACEMENT                                   XPGTI
491   1 EXIT TO :XPE13: (XPATR) IF INDICATED PARAMETER IS A DRDE FILE                                                  XPGTI
492   2    SAVE INTERFACE TABLE ENTRY CLASS/TYPE WORD, FLAG/DISP WORD AND SIZE WORD                                    XPGTI
493   2    IF LITERAL (NAME = 0)                                                                                       XPGTI
494   2    THEN                                                                                                        XPGTI
495   2    EXIT TO :ERR15: IF DISP + SIZE > SAVED SIZE                                                                 XPGTI
496   3       RESET BASE DISPLACEMENT TO ACTUAL DISP FIELD IN INTERFACE TABLE ENTRY                                     XPGTI
497   2    ELSE                                                                                                        XPGTI
498   3       INCREMENT BASE DISPLACEMENT TO CONVERT TO SUBSCRIPT                                                      XPGTI
499   3       SET TYPE FIELD TO FREE                                                                                   XPGTI
500   2    ENDIF                                                                                                       XPGTI
501   2    CLEAR ENTRY SUBSCRIPT BIT                                                                                   XPGTI
502   2    SET DISP FIELD TO SUM OF BASE DISPLACEMENT AND DISP                                                         XPGTI
503   2    SET SIZE FIELD TO SIZE                                                                                      XPGTI
504   2    CALL XPGET/XPPUT TO TRANSFER DATA                                                                           XPGTI
505   2    RESTORE ORIGINAL INTERFACE TABLE ENTRY                                                                      XPGTI
506   2    RESTORE XPGPF(XPGET) TO NOMINAL VALUE                                                                       XPGTI
507   1 EXIT XPXXI                                                                                                     XPGTI

508   2    :ERR15: TERMINATE PROCESSOR FOR ATTEMPT TO RETRIEVE TOO MUCH DATA                                           XPGTI
509   1 END XPXXI                                                                                                      XPGTI
```

```
511    1 *D0      CALLING PROCEDURE                                                                    XPREQ
512    1 *D0                                                                                           XPREQ
513    1 *D0   JSB  XPREQ                                                                              XPREQ
514    1 *D0   DEF  *+N+1    WHERE N IS THE NUMBER OF ACTUAL ARGUMENTS                                 XPREQ
515    1 *D0   DEF  OPTN     REQUIRED OPTION                                                           XPREQ
516    1 *D0   DEF  REQST    REQUIRED REQUEST                                                          XPREQ
517    1 *D0   DEF  ADRES    OPTIONAL ADDRESS (SEE OPTN)                                               XPREQ
518    1 *D0                                                                                           XPREQ
519    1 **********                                                                                    XPREQ
520    1 *D1                                                                                           XPREQ
521    1 *D1      XPGET/XPPUT BUFFERED AWA MANAGEMENT SERVICE                                          XPREQ
522    1 *D1                                                                                           XPREQ
523    1 **********                                                                                    XPREQ
524    1 *D2      INPUT                                                                                XPREQ
525    1 *D2   OPTN   - OPTION WORD                                                                    XPREQ
526    1 *D2            SIGN BIT   - 0 QUEUE REQUEST AND RETURN                                        XPREQ
527    1 *D2                         1 QUEUE REQUEST AND CLOSE BUFFER                                  XPREQ
528    1 *D2            RIGHT BIT  - 0 QUEUE REQUEST AND TRANSFER DATA TO/FROM                         XPREQ
529    1 *D2                         ADRES                                                             XPREQ
530    1 *D2                         1 QUEUE REQUEST ONLY                                              XPREQ
531    1 *D2   REQST  - EIGHT WORD REQUEST (SEE SDD 6.2.6.2)                                           XPREQ
532    1 *D2   ADRES  - STORAGE AREA TO RECEIVE RETURNED VALUES OR SUPPLY OUTPUT                       XPREQ
533    1 *D2            VALUES (USED ONLY WHEN OPTN(RIGHT BIT) = 0)                                    XPREQ
534    1 *D2                                                                                           XPREQ
535    1 *D2      EXTERNAL SYMBOLS                                                                     XPREQ
536    1 *D2   XPLU   - TERMINAL LOGICAL UNIT NUMBER                                                   XPREQ
537    1 *D2   XPRQB  - ADDRESS OF 64 WORD MANAGER REQUEST BUFFER                                      XPREQ
538    1 *D2   XPCLS  - CLASS I/O NUMBER FOR MANAGER COMMUNICATIONS                                    XPREQ
539    1 **********                                                                                    XPREQ
540    1 *D3      OUTPUT                                                                               XPREQ
541    1 *D3   IF A REQUEST FAILS, A MESSAGE IS ISSUED AND PROCESSING IS                               XPREQ
542    1 *D3   TERMINATED, OTHERWISE INCOMMING DATA WILL BE STORED IN THE                              XPREQ
543    1 *D3   ADDRESSES SUPPLIED.                                                                     XPREQ
544    1 *D3                                                                                           XPREQ
545    1 **********                                                                                    XPREQ
546    1 *D4      LOCAL VARIABLES                                                                      XPREQ
547    1 *D4   AADDS  - ADDRESS OF STORAGE AREA ADDRESS TABLE (ADDS)                                   XPREQ
548    1 *D4   ADDS   - TABLE OF ADDRESSES FOR STORAGE OF FETCHED DATA                                 XPREQ
549    1 *D4   APRMS  - ADDRESS OF PARMS ARRAY                                                         XPREQ
550    1 *D4   CLASS  - CLASS NUMBER USED TO TRANSMIT DATA TO MANAGER                                  XPREQ
551    1 *D4   PARMS  - PARAMETER RETURN AND SCRATCH AREA                                              XPREQ
552    1 *D4   PTR    - SAVE AREA FOR ADDS AND XPRQB POINTER                                           XPREQ
553    1 *D4                                                                                           XPREQ
554    1 **********                                                                                    XPREQ
555    1 *D5      NOTES                                                                                XPREQ
556    1 *D5   USES .ENTR, EXEC, XPXIT, XVPAW                                                          XPREQ
557    1 *D5                                                                                           XPREQ
558    1 *D5   IF DATA IS TO BE TRANSFERRED TO/FROM 'ADRES', THEN 'REQST' SIZE                         XPREQ
559    1 *D5   FIELD (WORD 6) MUST BE SET.                                                             XPREQ
560    1 *D5                                                                                           XPREQ
561    1 **********                                                                                    XPREQ
```

```
563   1 BEGIN XPREQ                                                                                                    XPREQ
564   2    MOVE REQUEST INTO BUFFER                                                                                    XPREQ
565   2    IF OPTION IS TO TRANSFER DATA                                                                               XPREQ
566   2    THEN                                                                                                        XPREQ
567   3       IF REQUEST IS TO RETRIEVE DATA (8)                                                                       XPREQ
568   3       THEN                                                                                                     XPREQ
569   4          STORE ADDRESS IN TABLE                                                                                XPREQ
570   3       ELSE SHOULD BE A REQUEST TO STORE DATA (7)                                                               XPREQ
571   4          OUTPUT DATA TO CLASS I/O                                                                              XPREQ
572   4          STORE CLASS NUMBER IN REQUEST WORD 8                                                                  XPREQ
573   3       ENDIF                                                                                                    XPREQ
574   2    ENDIF                                                                                                       XPREQ
575   2    INCREMENT POINTER                                                                                           XPREQ
576   2    IF BUFFER FULL OR OPTION IS TO CLOSE NON-EMPTY BUFFER                                                       XPREQ
577   2    THEN                                                                                                        XPREQ
578   3       CLOSE BUFFER                                                                                             XPREQ
579   3       TRANSMIT BUFFER TO MANAGER                                                                               XPREQ
580   3       PAW MANAGER WITH REQUEST FOR AWA MANAGEMENT                                                              XPREQ
581   3       RETRIEVE RETURN PARAMETERS                                                                               XPREQ
582   3       IF REWRITE FLAG SET (PARM5)                                                                              XPREQ
583   3       THEN                                                                                                     XPREQ
584   4          RETRIEVE REQUEST BUFFER                                                                               XPREQ
585   3       ENDIF                                                                                                    XPREQ
586   3       IF REQUESTS WERE SUCCESSFUL                                                                              XPREQ
587   3       THEN                                                                                                     XPREQ
588   4          DO FOR EACH REQUEST IN BUFFER                                                                         XPREQ
589   5             IF REQUEST TO RETURN DATA (8, 9 OR 16)                                                             XPREQ
590   5             THEN                                                                                               XPREQ
591   6                RETRIEVE AND STORE DATA IN ADDRESS CONTAINED IN TABLE                                           XPREQ
592   5             ENDIF                                                                                              XPREQ
593   4          ENDDO                                                                                                 XPREQ
594   4          CLEAR POINTER AND LOCAL CLASS NUMBER                                                                  XPREQ
595   3       ELSE                                                                                                     XPREQ
596   4          OUTPUT FAILURE MESSAGE (XP11)                                                                         XPREQ
597   4          DO FOR EACH REQUEST IN BUFFER                                                                         XPREQ
598   5             IF REQUEST SUCCESSFUL FOR DATA RETRIEVAL OR UNSUCCESSFUL STORE                                     XPREQ
599   5             THEN                                                                                               XPREQ
600   6                FREE CLASS I/O NUMBER AND SAM BUFFER                                                            XPREQ
601   5             ENDIF                                                                                              XPREQ
602   4          ENDDO                                                                                                 XPREQ
603   3       EXIT PROCESSOR WITH REQUEST FOR SEQUENCE TERMINATION                                                     XPREQ
604   3       ENDIF                                                                                                    XPREQ
605   2    ENDIF                                                                                                       XPREQ
606   1 END XPREQ                                                                                                      XPREQ
```

```
608   1 CD**************                                                    XPRDS
609   1 CD0                                                                XPRDS
610   1 CD0       FORTRAN CALLING PROCEDURE FOR PROCESSOR TC SPECIFIC TYPE  XPRDS
611   1 CD0                                                                XPRDS
612   1 CD0             CALL XPRDS (LU,PRMLEN,PROMPT,TYPE,DATLEN,IDIM,DATA,RETCS)  XPRDS
613   1 CD0                                                                XPRDS
614   1 CD**************                                                    XPRDS
615   1 CD1                                                                XPRDS
616   1 CD1       CALLS XPRDM,DECODES COMMUNICATIONS BUFFER LOOKING FOR A SPECIFIC  XPRDS
617   1 CD1       "TYPE" AND STORES IT IN "DATA" FOR "DATLEN" LOGICAL ELEMENTS  XPRDS
618   1 CD1                                                                XPRDS
619   1 CD**************                                                    XPRDS
620   1 CD2                                                                XPRDS
621   1 CD2       INPUTS FROM CALLING SEQUENCE:                            XPRDS
622   1 CD2                                                                XPRDS
623   1 CD2              LU - (INTEGER, 1 WORD) LOGICAL UNIT OF USER'S TERMINAL  XPRDS
624   1 CD2          PRMLEN - (INTEGER, 1 WORD) LENGTH IN WORDS OF THE CHARACTER  XPRDS
625   1 CD2                   STRING USED FOR THE USER PROMPT              XPRDS
626   1 CD2          PROMPT - (INTEGER, PRMLEN WORDS) IS THE CHARACTER STRING  XPRDS
627   1 CD2                   USED AS THE USER PROMPT                      XPRDS
628   1 CD2           TYPE  - (INTEGER, 1 WORD) CODE FOR THE DATA TYPE     XPRDS
629   1 CD2                   EXPECTED AS THE USER'S RESPONSE AS FOLLOWS:  XPRDS
630   1 CD2                             0 - FREE                           XPRDS
631   1 CD2                             1 - INTEGER                        XPRDS
632   1 CD2                             2 - REAL                           XPRDS
633   1 CD2                             3 - DOUBLE PRECISION               XPRDS
634   1 CD2                             4 - CHARACTER STRING LENGTH 2      XPRDS
635   1 CD2                             5 - CHARACTER STRING LENGTH 6      XPRDS
636   1 CD2                             6 - CHARACTER STRING LENGTH 18     XPRDS
637   1 CD2                             7 - CHARACTER STRING LENGTH 36     XPRDS
638   1 CD2                             8 - CHARACTER STRING LENGTH 72     XPRDS
639   1 CD2          DATLEN - (INTEGER, 1 WORD) NUMBER OF LOGICAL ELEMENTS IN  XPRDS
640   1 CD2                   THE DATA AREA. IF TYPE IS FREE, DATLEN IS THE  XPRDS
641   1 CD2                   NUMBER OF WORDS.                             XPRDS
642   1 CD2           IDIM  - (INTEGER, 1 WORD) THE COLUMN LENGTH OF THE DATA  XPRDS
643   1 CD2                   AREA IF IT IS A 2 DIMENSION ARRAY, ELSE 1 OR 0  XPRDS
644   1 CD2                   IF IT IS A VECTOR.                           XPRDS
645   1 CD2                                                                XPRDS
646   1 CD**************                                                    XPRDS
647   1 CD3                                                                XPRDS
648   1 CD3       OUTPUTS FROM CALLING SEQUENCE:                           XPRDS
649   1 CD3                                                                XPRDS
650   1 CD3             DATA  - (INTEGER, DIM DEPENDS ON DATLEN AND TYPE)  XPRDS
651   1 CD3                     AREA TO CONTAIN THE USERS RESPONSE         XPRDS
652   1 CD3             RETCS - (INTEGER, 1 WORD) RETURN CODE PASSED BACK TO  XPRDS
653   1 CD3                     CALLER:                                    XPRDS
654   1 CD3                 0 - NORMAL RETURN,DATA AREA CONTAINS USER'S RESPONSE  XPRDS
655   1 CD3                 1 - USER ENTERED Z. CONTENTS OF BUFFER UNPREDICTABLE.  XPRDS
656   1 CD3                 2 - USER ENTERED A CR. THERE IS NO RESPONSE.   XPRDS
657   1 CD3                 3 - USER PROMPT WAS TOO LONG. MAXIMUM LENGTH IS  XPRDS
658   1 CD3                     34 CHARACTERS OR 17 WORDS.                 XPRDS
659   1 CD3                 4 - PARAMETER LIST IS INVALID. EITHER:         XPRDS
660   1 CD3                       A.IDIM LESS THAN ZERO                    XPRDS
661   1 CD3                       B.DATLEN LESS THAN ONE                   XPRDS
662   1 CD3                       C.INVALID TYPE SPECIFIED.                XPRDS
663   1 CD3                                                                XPRDS
664   1 CD**************                                                    XPRDS
665   1 CD4                                                                XPRDS
666   1 CD4       INTERNAL VARIABLES                                       XPRDS
```

```
667   1 CD4                                                                  XPRDS
668   1 CD4          DATPTR - INDEX IN WORDS INTO DATA ARRAY WHERE NEXT       XPRDS
669   1 CD4                   ELEMENT IS TO BE STORED                        XPRDS
670   1 CD4          EFFTYP - THE EFFECTIVE INDEX (TYPE+1) OF THE CHARACTER   XPRDS
671   1 CD4                   STRING RESPONDED. IF TYPE IS FREE, IT IS THE   XPRDS
672   1 CD4                   CLOSEST SUPPORTED LENGTH, ELSE IT IS THE LENGTH XPRDS
673   1 CD4                   SPECIFIED BY TYPE.                             XPRDS
674   1 CD4          IND    - CONTAINS THE INDEX INTO MSGS ARAY OF THE CURRENT XPRDS
675   1 CD4                   ERROR MESSAGE.                                 XPRDS
676   1 CD4          LEN    - LENGTH OF CURRENT ERROR MESSAGE.               XPRDS
677   1 CD4          LENGTH - 9 WORD ARRAY REPRESENTING THE LENGTH IN WORDS   XPRDS
678   1 CD4                   OF THE 9 DATA TYPES 0-8 RESPECTIVELY           XPRDS
679   1 CD4          STKIND - COUNT OF NUMBER OF NESTED REPEATS AND INDEX     XPRDS
680   1 CD4                   TO THE CURRENT TOP OF THE STACK                XPRDS
681   1 CD4          STKPRN - 4 WORD ARRAY FOR STACK OF PARENTHESIS FLAGS     XPRDS
682   1 CD4          STKREP - 4 WORD ARRAY FOR STACK OF REPEAT COUNTS         XPRDS
683   1 CD4          STKTOK - 4 WORD ARRAY FOR STACK OF TOKEN POINTERS        XPRDS
684   1 CD4                   WHERE EACH REPEAT GROUP BEGINS                 XPRDS
685   1 CD4          TOKPTR - POINTER TO CURRENT TOKEN BEING PROCESSED        XPRDS
686   1 CD4          TYPE1  - INDEX 1-9 INTO LENGTH ARRAY BASED ON 0-8 TYPE   XPRDS
687   1 CD4                   TYPE1 = TYPE +1                                XPRDS
688   1 CD4                                                                  XPRDS
689   1 CD************                                                       XPRDS
690   1 CD5                                                                  XPRDS
691   1 CD5      SUBROUTINES AND FUNCTIONS CALLED:                           XPRDS
692   1 CD5                                                                  XPRDS
693   1 CD5           EXEC, XPRDM, XRMOV                                     XPRDS
694   1 CD5                                                                  XPRDS
695   1 CD5      PDL ROUTINES INCLUDED:                                      XPRDS
696   1 CD5                                                                  XPRDS
697   1 CD5           XPRDS, STRING, SUBSCR, REPET, ERRMSG                   XPRDS
698   1 CD5                                                                  XPRDS
699   1 CD************                                                       XPRDS
```

```
701    1 BEGIN XPRDS                                                                                    XPRDS
702    2    IF TYPE, DATA LENGTH, AND I DIMENSION ARE VALID THEN                                        XPRDS
703    3       CALL XPRDM TO PROMPT USER AND DECODE RESPONSE                                            XPRDS
704    3       IF XPRDM RETURN CODE IS ZERO THEN                                                        XPRDS
705    4          DO UNTIL EOS TOKEN IS FOUND                                                           XPRDS
706    5             SET MESSAGE NUMBER TO XP01                                                         XPRDS
707    5             CASE TOKEN (:NUMBER:,:NUMBER:,:NUMBER:,:STRING:,:NULL:,                            XPRDS
708    6                ,:SUBSCR:,:REPET:)                                                             XPRDS

709    6                :NUMBER:                                                                       XPRDS
710    6                ERREXIT IF TYPE DOES NOT MATCH OR IS NOT FREE PERFORM ERRMSG                    XPRDS
711    6                SET MESSAGE NUMBER TO XP02                                                     XPRDS
712    5                ERREXIT IF THERE IS NO ROOM IN DATA AREA PERFORM ERRMSG                         XPRDS
713    6                SET PREVIOUS TOKEN TO DATA                                                     XPRDS
714    6                CALL XRMOV TO MOVE DATA INTO DATA AREA                                         XPRDS
715    6                INCREMENT POINTER IN DATA AREA                                                 XPRDS
716    6                INCREMENT POINTER TO NEXT TOKEN                                                XPRDS

717    6                :STRING:                                                                       XPRDS
718    6                PERFORM STRING                                                                 XPRDS

719    6                :NULL:                                                                         XPRDS
720    6                SET MESSAGE NUMBER TO XP02                                                     XPRDS
721    6                ERREXIT IF THERE IS NO ROOM FOR THIS TOKEN PERFORM ERRMSG                       XPRDS
722    6                SET PREVIOUS TOKEN TO DATA                                                     XPRDS
723    6                INCREMENT POINTER IN DATA AREA                                                 XPRDS
724    6                INCREMENT POINTER TO NEXT TOKEN                                                XPRDS

725    6                :SUBSCR:                                                                       XPRDS
726    6                PERFORM SUBSCR                                                                 XPRDS

727    6                :REPET:                                                                        XPRDS
728    6                PERFORM REPET                                                                  XPRDS
729    5             ENDCASE                                                                           XPRDS
730    5             DO UNTIL TOKEN IS NOT A CLOSED PARENTHESIS                                        XPRDS
731    6                IF STACK IS NOT EMPTY AND                                                      XPRDS
732    7                   ((PREVIOUS TOKEN IS DATA AND TOP OF STACK PAREN FLAG =0) OR                 XPRDS
733    7                   (TOKEN IS CLOSE PAREN AND TOP OF STACK PAREN FLAG =1))THEN                  XPRDS
734    7                   DECREMENT TOP OF STACK REPEAT COUNT BY 1                                    XPRDS
735    7                   IF TOP OF STACK REPEAT COUNT > 0 THEN                                       XPRDS
736    8                      RESET TOKEN POINTER TO TOP OF STACK INDEX                                XPRDS
737    7                   ELSE                                                                        XPRDS
738    8                      POP TOP ENTRY ON STACK                                                   XPRDS
739    8                      IF TOKEN IS A CLOSED PAREN THEN                                          XPRDS
740    9                         INCREMENT TO NEXT TOKEN                                               XPRDS
741    8                      ENDIF                                                                    XPRDS
742    7                   ENDIF                                                                       XPRDS
743    6                ENDIF                                                                          XPRDS
744    5             ENDDO                                                                             XPRDS
745    4          ENDDO                                                                                XPRDS
746    3       ENDIF                                                                                   XPRDS
747    3       SET XPRDS RETURN CODE = XPRDM RETURN CODE                                               XPRDS
748    2    ELSE                                                                                       XPRDS
749    3       SET XPRDS RETURN CODE TO SAY INVALID PARAMETER LIST                                     XPRDS
750    2    ENDIF                                                                                      XPRDS
751    1 END XPRDS                                                                                     XPRDS
```

```
753   1 BEGIN STRING                                                                                    XPRDS
754   2    ERREXIT IF TYPE IS NOT CHARACTER OR FREE PERFORM ERRMSG                                       XPRDS
755   2    DETERMINE EFFECTIVE LENGTH OF RESPONSE AS NEXT LARGER SUPPORTED LENGTH                        XPRDS
756   2    ERREXIT IF TYPE OF RESPONSE > TYPE REQUESTED AND                                              XPRDS
757   2    ERREXIT IF TYPE IS NOT FREE PERFORM ERRMSG                                                    XPRDS
758   2    IF TYPE IS NOT FREE THEN                                                                      XPRDS
759   3       SET EFFECTIVE LENGTH = LENGTH REQUESTED                                                    XPRDS
760   2    ENDIF                                                                                         XPRDS
761   2    SET MESSAGE NUMBER TO XP02                                                                    XPRDS
762   2    ERREXIT IF THERE IS NO ROOM IN DATA AREA FOR THIS ELEMENT PERFORM ERRMSG                       XPRDS
763   2    CALL XRMOV TO MOVE BLANKS INTO DATA AREA FOR EFFECTIVE LENGTH                                 XPRDS
764   2    CALL XRMOV TO MOVE CHARACTER STRING INTO DATA AREA FOR REAL LENGTH                            XPRDS
765   2    SET PREVIOUS TOKEN IS DATA                                                                    XPRDS
766   2    INCREMENT POINTER IN DATA AREA                                                                XPRDS
767   2    INCREMENT TO NEXT TOKEN                                                                       XPRDS
768   1 END STRING                                                                                       XPRDS
769   1 *                                                                                                XPRDS
770   1 *                                                                                                XPRDS
771   1 *                                                                                                XPRDS
772   1 BEGIN SUBSCR                                                                                     XPRDS
773   2    INCREMENT POINTER TO NEXT TOKEN                                                               XPRDS
774   2    SET MESSAGE NUMBER TO XP07                                                                    XPRDS
775   2    ERREXIT IF TOKEN IS NOT AN INTEGER TO PERFORM ERRMSG                                          XPRDS
776   2    IF I-DIMENSION > 1 THEN                                                                       XPRDS
777   3       SET I TO INTESER VALUE                                                                     XPRDS
778   3       INCREMENT POINTER TO NEXT TOKEN                                                            XPRDS
779   3       ERREXIT IF TOKEN IS NOT AN INTEGER OR                                                      XPRDS
780   3       ERREXIT IF NEXT TOKEN IS NOT A CLOSE PAREN TO PERFORM ERRMSG                               XPRDS
781   3       SET MESSAGE NUMBER TO XP03                                                                 XPRDS
782   3       ERREXIT IF INTEGER > I-DIMENSION TO PERFORM ERRMSG                                         XPRDS
783   3       ERREXIT IF SUBSCRIPT IS OUT OF RANGE TO PERFORM ERRMSG                                     XPRDS
784   2    ELSE                                                                                          XPRDS
785   3       ERREXIT IF NEXT TOKEN IS NOT A CLOSE PAREN TO PERFORM ERRMSG                               XPRDS
786   3       SET MESSAGE NUMBER TO XP03                                                                 XPRDS
787   3       ERREXIT IF SUBSCRIPT IS OUT OF RANGE                                                       XPRDS
788   2    ENDIF                                                                                         XPRDS
789   2    ADJUST INDEX INTO DATA AREA ACCORDING TO SUBSCRIPT                                            XPRDS
790   2    INCREMENT POINTER BY 3 TOKENS                                                                 XPRDS
791   2    SET PREVIOUS TOKEN = SUBSCRIPT                                                                XPRDS
792   2    SET MESSAGE NUMBER TO XP04                                                                    XPRDS
793   2    ERREXIT IF TOKEN IS AN EOS OR                                                                 XPRDS
794   2    ERREXIT IF TOKEN IS A REPEAT OR                                                               XPRDS
795   2    ERREXIT IF TOKEN IS A CLOSE PAREN OR                                                          XPRDS
796   2    ERREXIT IF TOKEN IS A SUBSCRIPT TO PERFORM ERRMSG                                             XPRDS
797   1 END SUBSCR                                                                                       XPRDS
```

```
799   1 BEGIN REPET                                                    XPROS
800     2   SET MESSAGE NUMBER TO XPOS                                 XPROS
801     2   ERREXIT IF TOKEN IS AN EOS TO PERFORM ERRMSG               XPROS
802     2   INCREMENT STACK POINTER                                    XPROS
803     2   SET MESSAGE NUMBER TO XPO6                                 XPROS
804     2   ERREXIT IF THERE ARE TOO MANY NESTED REPEATS PERFORM ERRMSG XPROS
805     2   PUSH REPEAT COUNT ON STACK                                 XPROS
806     2   SET PARENTHESIS FLAG TO 0                                  XPROS
807     2   IF TOKEN IS AN OPEN PARENTHESIS THEN                       XPROS
808     3       INCREMENT POINTER TO NEXT TOKEN                        XPROS
809     3       SET PARENTHESIS FLAG TO 1                              XPROS
810     2   ENDIF                                                      XPROS
811     2   PUSH TOKEN INDEX AND PAREN FLAG ON STACK                   XPROS
812     2   SET PREVIOUS TOKEN IS A REPEAT                             XPROS
813   1 END REPET                                                      XPROS
814   1 *                                                              XPROS
815   1 *                                                              XPROS
816   1 *                                                              XPROS
817   1 BEGIN ERRMSG                                                   XPROS
818     2   CALL EXEC TO WRITE ERROR MESSAGE                           XPROS
819     2   PERFORM XPRDS - NO RETURN                                  XPROS
820   1 END ERRMSG                                                     XPROS
```

```
822   1 CD*************                                                                  XPRDM
823   1 CD0                                                                            XPRDM
824   1 CD0       FORTRAN CALLING PROCEDURE FOR PROCESSOR TC MIXED TYPE                XPRDM
825   1 CD0                                                                            XPRDM
826   1 CD0             CALL XPRDM (LU,PRMLEN,PROMPT,COMLEN,COMBUF,RETC)               XPRDM
827   1 CD0                                                                            XPRDM
828   1 CD*************                                                                XPRDM
829   1 CD1                                                                            XPRDM
830   1 CD1       WRITES "PRMLEN" WORDS OF "PROMPT" TO USER READS THE USER'S           XPRDM
831   1 CD1       RESPONSE, CONVERTS RESPONSE TO TOKENS IN "COMBUF" AND PASSES         XPRDM
832   1 CD1       BACK A RETURN CODE "RETC" INDICATING THE SUCCESS OF XPRDM            XPRDM
833   1 CD1       AND THE USEFULNESS OF "COMBUF".                                      XPRDM
834   1 CD1                                                                            XPRDM
835   1 CD*************                                                                XPRDM
836   1 CD2                                                                            XPRDM
837   1 CD2       INPUTS FROM CALLING SEQUENCE:                                        XPRDM
838   1 CD2                                                                            XPRDM
839   1 CD2             LU - (INTEGER, 1 WORD) LOGICAL UNIT OF USER'S TERMINAL         XPRDM
840   1 CD2         PRMLEN - (INTEGER , 1 WORD) LENGTH IN WORDS OF THE CHARACTER       XPRDM
841   1 CD2                  STRING USED FOR THE USER PROMPT                           XPRDM
842   1 CD2         PROMPT - (INTEGER, PRMLEN WORDS) IS THE CHARACTER STRING          XPRDM
843   1 CD2                  USED AS THE USER PROMPT.                                  XPRDM
844   1 CD2         COMLEN - (INTEGER, 1 WORD) LENGTH IN WORDS OF THE                 XPRDM
845   1 CD2                  COMMUNICATIONS BUFFER (COMBUF)                            XPRDM
846   1 CD2                                                                            XPRDM
847   1 CD*************                                                                XPRDM
848   1 CD3                                                                            XPRDM
849   1 CD3       OUTPUTS FROM CALLING SEQUENCE:                                       XPRDM
850   1 CD3                                                                            XPRDM
851   1 CD3             COMBUF - (INTEGER, COMLEN WORDS) ARRAY TO CONTAIN THE          XPRDM
852   1 CD3                      ENCODED USER RESPONSE                                 XPRDM
853   1 CD3              RETC - (INTEGER, 1 WORD) RETURN CODE PASSED BACK TO THE       XPRDM
854   1 CD3                      CALLING PROGRAM AS FOLLOWS:                           XPRDM
855   1 CD3                  0 - NORMAL RETURN  BUFFER CONTAINS USERS RESPONSE         XPRDM
856   1 CD3                  1 - USER ENTERED %.CONTENTS OF BUFFER UNPREDICTABLE.      XPRDM
857   1 CD3                  2 - USER ENTERED CR. BUFFER CONTAINS NO INFORMATION       XPRDM
858   1 CD3                  3 - USER PROMPT WAS TOO LONG. MAXIMUM LENGTH IS           XPRDM
859   1 CD3                      34 CHARACTERS OR 17 WORDS.                            XPRDM
860   1 CD3                                                                            XPRDM
861   1 CD*************                                                                XPRDM
862   1 CD4                                                                            XPRDM
863   1 CD4       INTERNAL VARIABLES                                                   XPRDM
864   1 CD4                                                                            XPRDM
865   1 CD4              COUNT - COUNTER USED FOR COUNTING NUMBER CHARACTERS IN        XPRDM
866   1 CD4                      A CHARACTER STRING AND NUMER DIGITS IN A NUMBER.      XPRDM
867   1 CD4           DBLINT - DOUBLE WORD USED TO ACCUMULATE AN INTEGER VALUE         XPRDM
868   1 CD4           DBLWD  - DOUBLE WORD USED TO ACCUMULATE A REAL OR DOUBLE         XPRDM
869   1 CD4                    VALUE.                                                  XPRDM
870   1 CD4           FLGCOM - COMMA FLAG                                              XPRDM
871   1 CD4                          0 - LAST CHARACTER NOT A COMMA                    XPRDM
872   1 CD4                          1 - LAST CHARACTER WAS A COMMA                    XPRDM
873   1 CD4           FLGCON - CONTINUE FLAG                                           XPRDM
874   1 CD4                          0 - THIS IS NOT A RESPONSE TO A CONTINUE          XPRDM
875   1 CD4                          1 - THIS IS A CONTINUED RESPONSE                  XPRDM
876   1 CD4           FLGNUM - NUMBER FLAG                                             XPRDM
877   1 CD4                          0 - POSITIVE NUMBER                               XPRDM
878   1 CD4                          1 - NEGATIVE NUMBER                               XPRDM
879   1 CD4           FLGPOW - POWER FLAG                                              XPRDM
880   1 CD4                          0 - POSITIVE POWER                                XPRDM
```

```
881    1 CD4                          1 - NEGATIVE POWER                                                    XPRDM
882    1 CD4            FLGTYP - TYPE OF REAL NUMBER                                                        XPRDM
883    1 CD4                          0 - SINGLE PRECISION                                                  XPRDM
884    1 CD4                          1 - DOUBLE PRECISION                                                  XPRDM
885    1 CD4                                                                                                XPRDM
886    1 CD*************                                                                                    XPRDM
887    1 CD5                                                                                                XPRDM
888    1 CD5         SUBROUTINES AND FUNCTIONS USED:                                                        XPRDM
889    1 CD5               EXEC, KCVT, XRMOV, XRPCK, XRUPK, OVF                                             XPRDM
890    1 CD5                                                                                                XPRDM
891    1 CD5         PDL ROUTINES USED:                                                                     XPRDM
892    1 CD5                                                                                                XPRDM
893    1 CD5               XPRDM, TOKENS, QUOTE, DIGIT, DCOL, DECPT, EORD,                                  XPRDM
894    1 CD5               INTEGR, REAL, DBL, INVAL, COMPUL                                                 XPRDM
895    1 CD5                                                                                                XPRDM
896    1 CD*************                                                                                    XPRDM
```

```
898   1 BEGIN XPRDM                                                                                                    XPRDM
899   2    IF PROMPT IS NOT TOO LONG THEN                                                                              XPRDM
900   3       CALL XRMOV TO MOVE PROMPT INTO OUTPUT AREA                                                               XPRDM
901   3       CALL EXEC TO WRITE PROMPT                                                                                XPRDM
902   3       SET XPRDM RETURN CODE TO NORMAL RETURN                                                                   XPRDM
903   3       SET CONTINUE FLAG OFF                                                                                    XPRDM
904   3       INITIALIZE COMMUNICATIONS BUFFER                                                                         XPRDM

905   3       :CONLOP:                                                                                                 XPRDM
906   3       CALL XRMOV TO INITIALIZE INPUT BUFFER TO BLANKS                                                          XPRDM
907   3       CALL EXEC TO READ RESPONSE                                                                               XPRDM
908   3       CALL XRUPK TO CONVERT A2 RESPONSE TO R1 FORMAT                                                           XPRDM
909   3       IF NUMBER OF WORDS READ IS NOT ZERO THEN                                                                 XPRDM
910   4          SET COMMA FLAG ON                                                                                     XPRDM
911   4          DO WHILE CONTINUE FLAG IS OFF                                                                         XPRDM
912   5             IF INPUT BUFFER IS COMPLETELY SCANNED THEN                                                         XPRDM
913   6                SET CONTINUE FLAG ON                                                                            XPRDM
914   6                IF COMMA FLAG IS ON THEN                                                                        XPRDM
915   7                   CALL EXEC TO WRITE CONTINUE                                                                  XPRDM
916   7                   GO TO :CONLOP:                                                                               XPRDM
917   6                ENDIF                                                                                           XPRDM
918   5             ELSE                                                                                               XPRDM
919   6                ERREXIT IF COMBUF IS FULL PERFORM COMFUL                                                        XPRDM
920   6                IF INPUT CHARACTER IS A COMMA THEN                                                              XPRDM
921   7                   IF COMMA FLAG IS ON THEN                                                                     XPRDM
922   8                      ERREXIT IF COMBUF CANNOT HOLD TOKEN PERFORM COMFUL                                        XPRDM
923   8                      STORE NULL FIELD TOKEN IN COMBUF                                                          XPRDM
924   8                      INCREMENT #WORDS IN COMBUF BY 1                                                           XPRDM
925   8                      INCREMENT #TOKENS IN COMBUF BY 1                                                          XPRDM
926   7                   ENDIF                                                                                        XPRDM
927   7                   SET COMMA FLAG ON                                                                            XPRDM
928   7                   GET NEXT INPUT CHARACTER                                                                     XPRDM
929   6                ELSE                                                                                            XPRDM
930   7                   SET COMMA FLAG OFF                                                                           XPRDM
931   7                   PERFORM TOKENS                                                                               XPRDM
932   6                ENDIF                                                                                           XPRDM
933   5             ENDIF                                                                                              XPRDM
934   4          ENDDO                                                                                                 XPRDM
935   3       ELSE                                                                                                     XPRDM
936   4          IF CONTINUE FLAG IS OFF THEN                                                                          XPRDM
937   5             SET XPRDM RETURN CODE TO SAY USER ENTERED CR                                                       XPRDM
938   4          ENDIF                                                                                                 XPRDM
939   3       ENDIF                                                                                                    XPRDM
940   2    ELSE                                                                                                        XPRDM
941   3       SET XPRDM RETURN CODE TO SAY PROMPT IS TOO LONG                                                          XPRDM
942   2    ENDIF                                                                                                       XPRDM
943   1 END XPRDM                                                                                                      XPRDM
```

```
945    1 BEGIN TOKENS                                                                                                  XPRDM
946    2    SET NEGATIVE NUMBER FLAG OFF                                                                               XPRDM
947    2    SET NEGATIVE POWER FLAG OFF                                                                                XPRDM
948    2    SET POWER = 0                                                                                              XPRDM
949    2    IF INPUT CHARACTER IS A DIGIT THEN                                                                         XPRDM
950    3       PERFORM DIGIT                                                                                           XPRDM
951    2    ELSE                                                                                                       XPRDM
952    3       *     X     "     (     )     .     +     -                                                            XPRDM
953    3       CASE J (:A:,:B:,:C:,:D:,:F:,:G:,:G:)                                                                   XPRDM

954    4          :A:                                                                                                 XPRDM
955    4          SET XTPRM RETURN CODE TO SAY X ENTERED                                                              XPRDM
956    4          SET CONTINUE FLAG ON                                                                                XPRDM

957    4          :B:                                                                                                 XPRDM
958    4          PERFORM QUOTE                                                                                       XPRDM

959    4          :C:                                                                                                 XPRDM
960    4          SET PAREN INDICATOR = 0                                                                             XPRDM
961    4          GO TO :E:                                                                                           XPRDM

962    4          :D:                                                                                                 XPRDM
963    4          SET PAREN INDICATOR = 1                                                                             XPRDM
964    4          GO TO :E:                                                                                           XPRDM

965    4          :E:                                                                                                 XPRDM
966    4          ERREXIT IF THERE IS NO ROOM FOR THIS TOKEN TO PERFORM COMFUL                                        XPRDM
967    4          STORE TOKEN (OPAR + PAREN INDICATOR) IN COMBUF                                                      XPRDM
968    4          INCREMENT #WORDS IN COMBUF BY 1                                                                     XPRDM
969    4          INCREMENT #TOKENS IN COMBUF BY 1                                                                    XPRDM
970    4          GET NEXT CHARACTER                                                                                  XPRDM

971    4          :F:                                                                                                 XPRDM
972    4          ERREXIT IF NEXT CHARACTER IS NOT A DIGIT PERFORM INVAL                                              XPRDM
973    4          SET INTEGER = 0                                                                                     XPRDM
974    4          PERFORM DECPT                                                                                       XPRDM

975    4          :G:                                                                                                 XPRDM
976    4          IF INPUT CHARACTER IS A - THEN                                                                      XPRDM
977    5             SET NEGATIVE NUMBER FLAG ON                                                                      XPRDM
978    4          ENDIF                                                                                               XPRDM
979    4          GET NEXT CHARACTER                                                                                  XPRDM
980    4          IF INPUT CHARACTER IS A DIGIT THEN                                                                  XPRDM
981    5             PERFORM DIGIT                                                                                    XPRDM
982    4          ELSE                                                                                                XPRDM
983    5             IF INPUT CHARACTER IS A . THEN                                                                   XPRDM
984    6                GO TO :F:                                                                                     XPRDM
985    5             ELSE                                                                                             XPRDM
986    6                PERFORM INVAL - NO RETURN                                                                     XPRDM
987    5             ENDIF                                                                                            XPRDM
988    4          ENDIF                                                                                               XPRDM
989    3       ENDCASE                                                                                                XPRDM
990    2    ENDIF                                                                                                      XPRDM
991    1 END TOKENS                                                                                                    XPRDM
```

```
 993    1 BEGIN QUOTE                                                                                                                XPRDM
 994    2    GET NEXT CHARACTER                                                                                                      XPRDM
 995    2    SET #CHARACTERS = 0                                                                                                     XPRDM
 996    2    DO WHILE (INPUT CHARACTER IS NOT A QUOTE AND                                                                            XPRDM
 997    3       INPUT BUFFER HAS NOT BEEN COMPLETELY SCANNED) OR                                                                     XPRDM
 998    3       (INPUT CHARACTER IS A QUOTE AND                                                                                      XPRDM
 999    3       NEXT CHARACTER IS A QUOTE AND                                                                                        XPRDM
1000    3       INPUT BUFFER HAS NOT BEEN COMPLETELY SCANNED)                                                                        XPRDM
1001    3       INCREMENT #CHARACTERS BY 1                                                                                           XPRDM
1002    3       MOVE CHARACTER INTO TEMPORARY BUFFER (#CHARACTERS)                                                                   XPRDM
1003    3       IF INPUT CHARACTER IS A QUOTE THEN                                                                                   XPRDM
1004    4          GET NEXT CHARACTER                                                                                                XPRDM
1005    3       ENDIF                                                                                                                XPRDM
1006    3       GET NEXT CHARACTER                                                                                                   XPRDM
1007    2    ENDDO                                                                                                                   XPRDM
1008    2    ERREXIT IF LENGTH OF CHARACTER STRING IS 0 OR                                                                           XPRDM
1009    2    ERREXIT IF INPUT CHARACTER IS NOT A QUOTE PERFORM INVAL                                                                 XPRDM
1010    2    ERREXIT IF THERE IS NO ROOM IN COMBUF FOR THIS TOKEN PERFORM COMFUL                                                     XPRDM
1011    2    STORE CHARACTER STRING TOKEN IN COMBUF                                                                                  XPRDM
1012    2    STORE NUMBER OF CHARACTERS IN COMBUF                                                                                    XPRDM
1013    2    CALL XRPCK TO CONVERT CHARACTERS FROM R1 TO A2 FORMAT                                                                   XPRDM
1014    2    INCREMENT #WORDS IN COMBUF BY 2+((#CHARACTERS + 1) /2)                                                                  XPRDM
1015    2    INCREMENT #TOKENS IN COMBUF BY 1                                                                                        XPRDM
1016    2    GET NEXT CHARACTER                                                                                                      XPRDM
1017    1 END QUOTE                                                                                                                  XPRDM
```

```
1019    1 BEGIN DIGIT                                                                                                                  XPRDM
1020    2    PERFORM DCOL                                                                                                              XPRDM
1021    2    IF INPUT BUFFER IS NOT EXHAUSTED THEN                                                                                     XPRDM
1022    3       IF INPUT CHARACTER IS A . THEN                                                                                         XPRDM
1023    4          PERFORM DECPT                                                                                                       XPRDM
1024    3       ELSE                                                                                                                   XPRDM
1025    4          IF INPUT CHARACTER IS AN "E" OR A "D" THEN                                                                          XPRDM
1026    5             PERFORM EORD                                                                                                     XPRDM
1027    4          ELSE                                                                                                                XPRDM
1028    5             IF INPUT CHARACTER IS AN "R" THEN                                                                                XPRDM
1029    6                ERREXIT IF THERE IS NO ROOM IN COMBUF FOR THIS TOKEN PERFORM COMFUL                                           XPRDM
1030    6                ERREXIT IF INTEGER IS ZERO PERFORM INVAL                                                                      XPRDM
1031    6                STORE REPEAT TOKEN IN COMBUF                                                                                  XPRDM
1032    6                INCREMENT #WORDS IN COMBUF BY 2                                                                               XPRDM
1033    6                INCREMENT #TOKENS BY 1                                                                                        XPRDM
1034    6                GET NEXT CHARACTER                                                                                            XPRDM
1035    5             ELSE                                                                                                             XPRDM
1036    6                PERFORM INTEGR                                                                                                XPRDM
1037    5             ENDIF                                                                                                            XPRDM
1038    4          ENDIF                                                                                                               XPRDM
1039    3       ENDIF                                                                                                                  XPRDM
1040    2    ELSE                                                                                                                      XPRDM
1041    3       PERFORM INTEGR                                                                                                         XPRDM
1042    2    ENDIF                                                                                                                     XPRDM
1043    1 END DIGIT                                                                                                                    XPRDM
1044    1 *                                                                                                                            XPRDM
1045    1 *                                                                                                                            XPRDM
1046    1 *                                                                                                                            XPRDM
1047    1 BEGIN DCOL                                                                                                                   XPRDM
1048    2    SET INTEGER=0                                                                                                             XPRDM
1049    2    SET COUNTER=0                                                                                                             XPRDM
1050    2    DO WHILE CHARACTER IS A DIGIT AND                                                                                         XPRDM
1051    3       WHILE INPUT BUFFER IS NOT EXHAUSTED                                                                                    XPRDM
1052    3       SET INTEGER= (INTEGER * 10) + INPUT CHARACTER - 48                                                                     XPRDM
1053    3       ERREXIT IF INTEGER OVERFLOW HAS OCCURRED TO PERFORM OVFLOW                                                             XPRDM
1054    3       INCREMENT COUNTER BY 1                                                                                                 XPRDM
1055    3       GET NEXT CHARACTER                                                                                                     XPRDM
1056    2    ENDDO                                                                                                                     XPRDM
1057    1 END DCOL                                                                                                                     XPRDM
```

```
1059    1 BEGIN DECPT                                                                                    XPRDM
1060    2    CONVERT INTEGER VALUE TO DOUBLE PRECISION VALUE                                             XPRDM
1061    2    GET NEXT CHARACTER                                                                          XPRDM
1062    2    IF INPUT BUFFER IS NOT EXHAUSTED THEN                                                       XPRDM
1063    3       IF INPUT CHARACTER IS A DIGIT THEN                                                       XPRDM
1064    4          PERFORM DCOL                                                                          XPRDM
1065    4          ADD FRACTIONAL PART TO DOUBLE PRECISION VALUE                                         XPRDM
1066    4          ERREXIT IF INTEGER OVERFLOW HAS OCCURRED TO PERFORM OVFLOW                            XPRDM
1067    3       ENDIF                                                                                    XPRDM
1068    3       IF INPUT CHARACTER IS AN "E" OR A "D" THEN                                               XPRDM
1069    4          PERFORM EORD                                                                          XPRDM
1070    3       ELSE                                                                                     XPRDM
1071    4          PERFORM REAL                                                                          XPRDM
1072    3       ENDIF                                                                                    XPRDM
1073    2    ELSE                                                                                        XPRDM
1074    3       PERFORM REAL                                                                             XPRDM
1075    2    ENDIF                                                                                       XPRDM
1076    1 END DECPT                                                                                      XPRDM
```

```
1078    1 BEGIN EORD                                                                                                    XPRDM
1079    2    IF INPUT CHARACTER IS AN "E" THEN                                                                          XPRDM
1080    3       SET TYPE FLAG TO "E"                                                                                    XPRDM
1081    2    ELSE                                                                                                       XPRDM
1082    3       SET TYPE FLAG TO "D"                                                                                    XPRDM
1083    2    ENDIF                                                                                                      XPRDM
1084    2    GET NEXT CHARACTER                                                                                         XPRDM
1085    2    ERREXIT IF INPUT BUFFER IS EXHAUSTED PERFORM INVAL                                                         XPRDM
1086    2    IF INPUT CHARACTER IS A - THEN                                                                             XPRDM
1087    3       SET NEGATIVE POWER FLAG ON                                                                              XPRDM
1088    3       GET NEXT CHARACTER                                                                                      XPRDM
1089    2    ELSE                                                                                                       XPRDM
1090    3       IF INPUT CHARACTER IS A + THEN                                                                          XPRDM
1091    4          SET NEGATIVE POWER FLAG OFF                                                                          XPRDM
1092    4          GET NEXT CHARACTER                                                                                   XPRDM
1093    3       ENDIF                                                                                                   XPRDM
1094    2    ENDIF                                                                                                      XPRDM
1095    2    ERREXIT IF INPUT BUFFER IS EXHAUSTED OR                                                                    XPRDM
1096    2    ERREXIT IF INPUT CHARACTER IS NOT A DIGIT PERFORM INVAL                                                    XPRDM
1097    2    PERFORM DCOL                                                                                               XPRDM
1098    2    IF NEGATIVE POWER FLAG IS ON THEN                                                                          XPRDM
1099    3       SET POWER = -POWER                                                                                      XPRDM
1100    2    ENDIF                                                                                                      XPRDM
1101    2    IF TYPE FLAG IS "E" THEN                                                                                   XPRDM
1102    3       PERFORM REAL                                                                                            XPRDM
1103    2    ELSE                                                                                                       XPRDM
1104    3       BEGIN DBL                                                                                               XPRDM
1105    4          ERREXIT IF NEXT TOKEN IS NOT A COMMA AND                                                             XPRDM
1106    4          ERREXIT IF NEXT TOKEN IS NOT A CLOSED PAREN AND                                                      XPRDM
1107    4          ERREXIT IF INPUT BUFFER IS NOT EXHAUSTED PERFORM INVAL                                               XPRDM
1108    4          ERREXIT IF THERE IS NO ROOM IN COMBUF FOR THIS TOKEN PERFORM COMFUL                                  XPRDM
1109    4          SET DOUBLE = DOUBLE * 10 ** POWER                                                                    XPRDM
1110    4          ERREXIT IF INTEGER OVERFLOW HAS OCCURRED TO PERFORM OVFLOW                                           XPRDM
1111    4          IF NEGATIVE NUMBER FLAG IS ON THEN                                                                   XPRDM
1112    5             SET DOUBLE = -DOUBLE                                                                              XPRDM
1113    4          ENDIF                                                                                                XPRDM
1114    4          STORE DOUBLE TOKEN IN COMBUF                                                                         XPRDM
1115    4          INCREMENT #WORDS IN COMBUF BY 4                                                                      XPRDM
1116    4          INCREMENT #TOKENS IN COMBUF BY 1                                                                     XPRDM
1117    3       END DBL                                                                                                 XPRDM
1118    2    ENDIF                                                                                                      XPRDM
1119    1 END EORD                                                                                                      XPRDM
```

```
1121    1 BEGIN INTEGR                                                                                  XPRDM
1122    2    ERREXIT IF NEXT TOKEN IS NOT A COMMA AND                                                   XPRDM
1123    2    ERREXIT IF NEXT TOKEN IS NOT A CLOSED PAREN AND                                            XPRDM
1124    2    ERREXIT IF INPUT BUFFER IS NOT EXHAUSTED PERFORM INVAL                                     XPRDM
1125    2    ERREXIT IF THERE IS NO ROOM IN COMBUF FOR THIS TOKEN PERFORM COMFUL                        XPRDM
1126    2    CONVERT DOUBLE TO INTEGER                                                                  XPRDM
1127    2    ERREXIT IF INTEGER OVERFLOW HAS OCCURRED TO PERFORM OVFLOW                                 XPRDM
1128    2    IF NEGATIVE NUMBER FLAG IS ON THEN                                                         XPRDM
1129    3       SET INTEGER = -INTEGER                                                                  XPRDM
1130    2    ENDIF                                                                                      XPRDM
1131    2    STORE INTEGER TOKEN IN COMBUF                                                              XPRDM
1132    2    INCREMENT #WORDS IN COMBUF BY 2                                                            XPRDM
1133    2    INCREMENT #TOKENS IN COMBUF BY 1                                                           XPRDM
1134    1 END INTEGR                                                                                    XPRDM
1135    1 *                                                                                             XPRDM
1136    1 *                                                                                             XPRDM
1137    1 *                                                                                             XPRDM
1138    1 BEGIN REAL                                                                                    XPRDM
1139    2    ERREXIT IF NEXT TOKEN IS NOT A COMMA AND                                                   XPRDM
1140    2    ERREXIT IF NEXT TOKEN IS NOT A CLOSED PAREN AND                                            XPRDM
1141    2    ERREXIT IF INPUT BUFFER IS NOT EXHAUSTED PERFORM INVAL                                     XPRDM
1142    2    ERREXIT IF THERE IS NO ROOM IN COMBUF FOR THIS TOKEN PERFORM COMFUL                        XPRDM
1143    2    SET REAL = DOUBLE * 10 ** POWER                                                            XPRDM
1144    2    ERREXIT IF INTEGER OVERFLOW HAS OCCURRED TO PERFORM OVFLOW                                 XPRDM
1145    2    IF NEGATIVE NUMBER FLAG IS ON THEN                                                         XPRDM
1146    3       SET REAL = -REAL                                                                        XPRDM
1147    2    ENDIF                                                                                      XPRDM
1148    2    STORE TOKEN IN COMBUF                                                                      XPRDM
1149    2    INCREMENT #WORDS IN COMBUF BY 3                                                            XPRDM
1150    2    INCREMENT #TOKENS IN COMBUF BY 1                                                           XPRDM
1151    1 END REAL                                                                                      XPRDM
```

```
1153  1 BEGIN INVAL                                                    XPRDM
1154  2     CALL KCVT TO CONVERT OCTAL CHARACTER NUMBER TO ASKII        XPRDM
1155  2     CALL EXEC TO WRITE ERROR MESSAGE                           XPRDM
1156  2     PERFORM XPRDM TO DISPLAY ORIGINAL PROMPT - NO RETURN       XPRDM
1157  1 END INVAL                                                      XPRDM
1158  1 *                                                              XPRDM
1159  1 *                                                              XPRDM
1160  1 *                                                              XPRDM
1161  1 BEGIN COMFUL                                                   XPRDM
1162  2     CALL EXEC TO WRITE ERROR MESSAGE                           XPRDM
1163  2     PERFORM XPRDM TO DISPLAY ORIGINAL PROMPT - NO RETURN       XPRDM
1164  1 END COMFUL                                                     XPRDM
1165  1 *                                                              XPRDM
1166  1 *                                                              XPRDM
1167  1 *                                                              XPRDM
1168  1 BEGIN OVFLOW                                                   XPRDM
1169  2     CALL KCVT TO CONVERT OCTAL TO ASKII                        XPRDM
1170  2     CALL EXEC TO WRITE ERROR MESSAGE                           XPRDM
1171  2     PERFORM XPRDM TO DISPLAY ORIGINAL PROMPT - NO RETURN       XPRDM
1172  1 END OVFLOW                                                     XPRDM
```

```
1174    1 CD0       FORTRAN CALLING PROCEDURE                                                    XPXIT
1175    1 CD0                                                                                    XPXIT
1176    1 CD0    CALL XPXIT (LU, RPARMS)                                                         XPXIT
1177    1 CD0                                                                                    XPXIT
1178    1 C*********                                                                             XPXIT
1179    1 CD1                                                                                    XPXIT
1180    1 CD1       TERMINATE A PROGRAM AFTER WAITING ON ANY BUFFERED I/O TO LU TO               XPXIT
1181    1 CD1       COMPLETE THEN PASS PARAMETERS BACK TO THE FATHER TASK.                       XPXIT
1182    1 CD1                                                                                    XPXIT
1183    1 C*********                                                                             XPXIT
1184    1 CD2       INPUT                                                                        XPXIT
1185    1 CD2    LU      - LOGICAL UNIT OF TERMINAL OR OTHER OUTPUT DEVICE FOR WHICH             XPXIT
1186    1 CD2              PRINT MAY NOT BE COMPLETED.  A VALUE OF ZERO WILL                     XPXIT
1187    1 CD2              INDICATE NO I/O WAIT                                                  XPXIT
1188    1 CD2    RPARMS - ARRAY OF FIVE PARAMETERS TO BE RETURNED TO FATHER TASK               XPXIT
1189    1 CD2                                                                                    XPXIT
1190    1 C*********                                                                             XPXIT
1191    1 CD3       OUTPUT                                                                       XPXIT
1192    1 CD3    RPARMS IS RETURNED TO THE FATHER TASK                                           XPXIT
1193    1 CD3                                                                                    XPXIT
1194    1 C*********                                                                             XPXIT
1195    1 CD5       NOTES                                                                        XPXIT
1196    1 CD5    USES EXEC, PRTN                                                                 XPXIT
1197    1 CD5                                                                                    XPXIT
1198    1 CD5    XPXIT DOES NOT RETURN TO THE CALLER.                                            XPXIT
1199    1 CD5                                                                                    XPXIT
1200    1 C*********                                                                             XPXIT
1201    1 *                                                                                      XPXIT
1202    1 *                                                                                      XPXIT
1203    1 *                                                                                      XPXIT
1204    1 *                                                                                      XPXIT
1205    1 BEGIN XPXIT                                                                            XPXIT
1206    2    IF LU IS NON-ZERO                                                                   XPXIT
1207    2    THEN                                                                                XPXIT
1208    3       WAIT ON ANY CLASS I/O TO COMPLETE (CLASS I/O CONTROL THEN GET)                    XPXIT
1209    2    ENDIF                                                                               XPXIT
1210    2    RETURN PARAMETERS TO FATHER                                                         XPXIT
1211    2    TERMINATE PROGRAM                                                                   XPXIT
1212    1 END XPXIT                                                                              XPXIT
```

```
 SYMBOL DEFINITION TABLE
:A      :    954
 ACCESS      251
:B      :    957
:C      :    959
 COMFUL     1161
:COMLOP:     905
:D      :    962
 DBL        1104
 DCOL       1047
 DECPT      1059
 DIGIT      1019
:E      :    965
 EORD       1078
 ERRMSG      817
:ERR12  :    328
:ERR14  :    330
:ERR15  :    508
:F      :    971
:G      :    975
 INTEGR     1121
 INVAL      1153
:NULL   :    719
:NUMBER:     709
 OVFLOW     1168
 QUOTE       993
 REAL       1138
 REPET       799
:REPET  :    727
:STRING:     717
 STRING      753
 SUBSCR      772
:SUBSCR:     725
 TOKENS      945
 XPATR        67
:XPE13  :    329
 XPGET       245
 XPGTI       480
 XPINI       333
 XPPTI       484
 XPPUT       247
 XPQFN       350
 XPRDM       898
 XPRDS       701
 XPREQ       563
 XPSBC       340
 XPXIT      1205
 XPXXI       488

@XGT F.PULIST
```

```
 2    1 *D0      FORTRAN CALLING PROCEDURE                                                    XRBIT
 3    1 *D0                                                                                   XRBIT
 4    1 *D0   CALL XRBIT (BIT, BITNUM, STRING)                                                XRBIT
 5    1 *D0                                                                                   XRBIT
 6    1 **********                                                                            XRBIT
 7    1 *D1                                                                                   XRBIT
 8    1 *D1      BIT SET/CLEAR ROUTINE FOR MULTIPLE WORD BIT STRINGS.  SET THE                XRBIT
 9    1 *D1   VALUE OF BIT NUMBER 'BITNUM' IN STRING 'STRING' TO 'BIT'.                       XRBIT
10    1 *D1                                                                                   XRBIT
11    1 **********                                                                            XRBIT
12    1 *D2      INPUT                                                                        XRBIT
13    1 *D2   BIT    - INTEGER VALUE OF ZERO OR ONE TO BE SET IN BIT NUMBER                   XRBIT
14    1 *D2            'BITNUM' OF 'STRING                                                    XRBIT
15    1 *D2   BITNUM - INTEGER BIT NUMBER OF BIT TO BE SET/CLEARED (FIRST BIT OF             XRBIT
16    1 *D2            STRING IS BIT NUMBER ZERO)                                           XRBIT
17    1 *D2   STRING - BIT STRING OF AT LEAST 'BITNUM' BITS IN LENGTH INTO WHICH            XRBIT
18    1 *D2            'BIT' IS TO BE SET                                                   XRBIT
19    1 *D2                                                                                   XRBIT
20    1 **********                                                                            XRBIT
21    1 *D3      OUTPUT                                                                       XRBIT
22    1 *D3   STRING - BIT NUMBER 'BITNUM' HAS THE VALUE OF 'BIT'                             XRBIT
23    1 *D3                                                                                   XRBIT
24    1 **********                                                                            XRBIT
25    1 *D5      NOTES                                                                        XRBIT
26    1 *D5   USES .ENTR, XRSET                                                               XRBIT
27    1 *D5                                                                                   XRBIT
28    1 **********                                                                            XRBIT
29    1 *                                                                                     XRBIT
30    1 *                                                                                     XRBIT
31    1 *                                                                                     XRBIT
32    1 *                                                                                     XRBIT
33    1 BEGIN XRBIT                                                                           XRBIT
34    2    TRANSFER CALLING SEQUENCE                                                        XRBIT
35    2    COMPUTE ADDRESS OF WORD CONTAINING BITNUM                                        XRBIT
36    2    COMPUTE BIT NUMBER WITHIN WORD                                                   XRBIT
37    2    CALL XRSET TO SET/CLEAR BIT                                                      XRBIT
38    1 END XRBIT                                                                           XRBIT
```

```
40    1 *DO      INTEGER FUNCTION                                                                    XRCPR
41    1 *DO                                                                                          XRCPR
42    1 *DO     XRCPR(LENGTH, ARRAY1, ARRAY2)                                                        XRCPR
43    1 *DO                                                                                          XRCPR
44    1 **********                                                                                   XRCPR
45    1 *D1                                                                                          XRCPR
46    1 *D1      COMPARE 'LENGTH' WORDS OF 'ARRAY1' TO 'ARRAY2'.  IF THE ARRAYS                      XRCPR
47    1 *D1      ARE IDENTICAL A VALUE OF 0 IS RETURNED.  IF THE COMPARE FAILS                       XRCPR
48    1 *D1      A VALUE IS RETURNED INDICATING WHETHER THE FIRST UNEQUAL                            XRCPR
49    1 *D1      ELEMENT OF ARRAY1 IS LESS THAN ARRAY2 (-1) OR GREATER (+1)                          XRCPR
50    1 *D1                                                                                          XRCPR
51    1 **********                                                                                   XRCPR
52    1 *D2      INPUT                                                                               XRCPR
53    1 *D2   LENGTH - POSITIVE INTEGER NUMBER OF WORDS TO BE COMPARED                               XRCPR
54    1 *D2   ARRAY1 - ARRAY OF INTEGERS OR CHARACTERS TO BE COMPARED TO ARRAY2                      XRCPR
55    1 *D2   ARRAY2 - ARRAY OF INTEGERS OR CHARACTERS TO BE COMPARED TO ARRAY1                      XRCPR
56    1 *D2                                                                                          XRCPR
57    1 **********                                                                                   XRCPR
58    1 *D3      OUTPUT                                                                              XRCPR
59    1 *D3   FUNCTION VALUE - 0, ARRAYS EQUAL                                                      XRCPR
60    1 *D3                    -1, ARRAYS NOT EQUAL, ARRAY1 < ARRAY2                                 XRCPR
61    1 *D3                    +1, ARRAYS NOT EQUAL, ARRAY1 > ARRAY2                                 XRCPR
62    1 *D3                                                                                          XRCPR
63    1 **********                                                                                   XRCPR
64    1 *D5      NOTES                                                                               XRCPR
65    1 *D5   USES .ENTR                                                                             XRCPR
66    1 *D5                                                                                          XRCPR
67    1 **********                                                                                   XRCPR
68    1 *                                                                                            XRCPR
69    1 *                                                                                            XRCPR
70    1 *                                                                                            XRCPR
71    1 *                                                                                            XRCPR
72    1 *                                                                                            XRCPR
73    1 BEGIN XRCPR                                                                                  XRCPR
74    2    TRANSFER CALLING SEQUENCE                                                                 XRCPR
75    2    SETUP COMPARE                                                                             XRCPR
76    2    COMPARE ARRAY1 AND ARRAY2                                                                 XRCPR
77    2    RETURN RESULT FLAG                                                                        XRCPR
78    1 END XRCPR                                                                                    XRCPR
```

```
 80    1 CD0      FORTRAN CALLING PROCEDURE.                                                    XRD18
 81    1 CD0                                                                                    XRD18
 82    1 CD0   CALL XRD18 (DOUBLE, ASCII)                                                       XRD18
 83    1 CD0                                                                                    XRD18
 84    1 C*********                                                                             XRD18
 85    1 CD1                                                                                    XRD18
 86    1 CD1      CONVERT A DOUBLE PRECISION REAL NUMBER TO AN ASCII STRING IN                   XRD18
 87    1 CD1      1PD18.11 FORMAT                                                               XRD18
 88    1 CD1                                                                                    XRD18
 89    1 C*********                                                                             XRD18
 90    1 CD2      INPUT                                                                         XRD18
 91    1 CD2   DOUBLE - THREE WORD DOUBLE PRECISION REAL NUMBER TO BE CONVERTED                 XRD18
 92    1 CD2                                                                                    XRD18
 93    1 C*********                                                                             XRD18
 94    1 CD3      OUTPUT                                                                        XRD18
 95    1 CD3   ASCII  - NINE WORD ASCII CHARACTER STRING REPRESENTATION OF                      XRD18
 96    1 CD3              'DOUBLE'                                                              XRD18
 97    1 CD3                                                                                    XRD18
 98    1 C*********                                                                             XRD18
 99    1 CD4      LOCAL                                                                         XRD18
100    1 CD4   D      - WORKING LOCATION FOR ABSOLUTE VALUE OR 'DOUBLE'                         XRD18
101    1 CD4              REPEATEDLY MODIFIED TO EXTRACT REMAINING DECIMAL DIGITS               XRD18
102    1 CD4                                                                                    XRD18
103    1 C*********                                                                             XRD18
104    1 CD5      NOTES                                                                         XRD18
105    1 CD5   USES DBLE, FLOAT, IAND, IDINT, IOR, KCVT, XREXT                                  XRD18
106    1 CD5                                                                                    XRD18
107    1 C*********                                                                             XRD18
108    1 *                                                                                      XRD18
109    1 *                                                                                      XRD18
110    1 *                                                                                      XRD18
111    1 *                                                                                      XRD18
112    1 BEGIN XRD18                                                                            XRD18
113    2    SET SIGN FIELD                                                                      XRD18
114    2    MOVE ABSOLUTE VALUE OF 'DOUBLE' INTERNAL                                            XRD18
115    2    COMPUTE EXPONENT                                                                    XRD18
116    2    SET SIGN AND VALUE OF EXPONENT FIELD                                                XRD18
117    2    REDUCE VALUE TO RANGE OF 1 < VALUE <10                                              XRD18
118    2    EXTRACT FIRST DIGIT, MERGE WITH SIGN AND STORE FIELD                                XRD18
119    2    EXTRACT SECOND DIGIT, MERGE WITH DECIMAL AND STORE FIELD                            XRD18
120    2    DO FOR NEXT FIVE PAIRS OF DIGITS                                                    XRD18
121    3       MULTIPLY BY 100 TO EXTRACT PAIR                                                  XRD18
122    3       EXTRACT DIGITS AND STORE FIELD                                                   XRD18
123    2    ENDDO                                                                               XRD18
124    1 END XRD18                                                                              XRD18
```

```
126   1 CD0      FORTRAN CALLING PROCEDURE                                                    XREQ
127   1 CD0                                                                                   XREQ
128   1 CD0   CALL XREQ                                                                       XREQ
129   1 CD0                                                                                   XREQ
130   1 C*********                                                                            XREQ
131   1 CD1                                                                                   XREQ
132   1 CD1   MAKE A WORK AREA MANAGEMENT REQUEST AND WAIT FOR RESPONSE                       XREQ
133   1 CD1                                                                                   XREQ
134   1 C*********                                                                            XREQ
135   1 CD2      INPUT                                                                        XREQ
136   1 CD2   COMMON XE - CLASNO, FLAGS, LU, REQBUF, REQPTR                                   XREQ
137   1 CD2   ID SEGMENT PARAMETERS RETURNED FROM THE MANAGER                                 XREQ
138   1 CD2                                                                                   XREQ
139   1 C*********                                                                            XREQ
140   1 CD3      OUTPUT                                                                       XREQ
141   1 CD3   COMMON XE - REQPTR                                                              XREQ
142   1 CD3   CLASS I/O WRITE/READ TO CLASS 'CLASNO'                                          XREQ
143   1 CD5   REQBUF AND RESPONSE IS PRINTED IF REQUESTED                                     XREQ
144   1 CD3                                                                                   XREQ
145   1 C*********                                                                            XREQ
146   1 CD5      NOTES                                                                        XREQ
147   1 CD5   COMMUNICATES WITH FDS MANAGER FATHER TASK                                       XREQ
148   1 CD5   USES EXEC, IAND, XRMOV, XRMSG, XR06, XRSFL, XUDPL, XVPAW                        XREQ
149   1 CD5                                                                                   XREQ
150   1 C*********                                                                            XREQ
151   1 *                                                                                     XREQ
152   1 *                                                                                     XREQ
153   1 *                                                                                     XREQ
154   1 *                                                                                     XREQ
155   1 BEGIN XREQ                                                                            XREQ
156   2    PERFORM TRACE                                                                      XREQ
157   2    OUTPUT REQUESTS TO MANAGER                                                         XREQ
158   2    REQUEST AWA MANAGEMENT AND WAIT FOR RESPONSE                                       XREQ
159   2    RETRIEVE MANAGER RESPONSE                                                          XREQ
160   2    PERFORM TRACE                                                                      XREQ
161   2    RETURN RESPONSE IN REQPTR                                                          XREQ
162   1 END XREQ                                                                              XREQ
163   1 BEGIN TRACE                                                                           XREQ
164   2    IF TRACE REQUESTED                                                                 XREQ
165   2    THEN                                                                               XREQ
166   3       DO FOR EACH REQUEST                                                             XREQ
167   4          CALL XUDPL TO FORMAT LINE                                                    XREQ
168   4          OUTPUT LINE                                                                  XREQ
169   3       ENDDO                                                                           XREQ
170   3       OUTPUT PARM1 AND PARM2                                                          XREQ
171   2    ENDIF                                                                              XREQ
172   1 END TRACE                                                                             XREQ
```

```
174   1 *D0      INTEGER FUNCTION                                                                                    XREXT
175   1 *D0                                                                                                          XREXT
176   1 *D0   XREXT(START, LENGTH, SOURCE)                                                                           XREXT
177   1 *D0                                                                                                          XREXT
178   1 **********                                                                                                   XREXT
179   1 *D1                                                                                                          XREXT
180   1 *D1      EXTRACT 'LENGTH' BITS OF 'SOURCE' BEGINNING WITH BIT 'START'                                        XREXT
181   1 *D1      AND RIGHT ADJUST                                                                                    XREXT
182   1 *D1                                                                                                          XREXT
183   1 **********                                                                                                   XREXT
184   1 *D2      INPUT                                                                                               XREXT
185   1 *D2   START  - INTEGER INDICATING LEFT MOST BIT OF FIELD TO BE EXTRACTED                                     XREXT
186   1 *D2            (SIGN BIT = 0)                                                                              XREXT
187   1 *D2   LENGTH - POSITIVE INTEGER SIZE OF FIELD TO BE EXTRACTED                                               XREXT
188   1 *D2   SOURCE - WORD FROM WHICH FIELD IS TO BE EXTRACTED                                                     XREXT
189   1 *D2                                                                                                          XREXT
190   1 **********                                                                                                   XREXT
191   1 *D5      NOTES                                                                                               XREXT
192   1 *D5   USES .ENTR                                                                                             XREXT
193   1 *D5                                                                                                          XREXT
194   1 **********                                                                                                   XREXT
195   1 *                                                                                                            XREXT
196   1 *                                                                                                            XREXT
197   1 *                                                                                                            XREXT
198   1 *                                                                                                            XREXT
199   1 BEGIN XREXT                                                                                                  XREXT
200   2    TRANSFER CALLING SEQUENCE                                                                                 XREXT
201   2    IF START NOT = 0                                                                                          XREXT
202   2    THEN                                                                                                      XREXT
203   3       CONSTRUCT SHIFT                                                                                        XREXT
204   3       LOAD A WITH SOURCE                                                                                     XREXT
205   3       SHIFT BA LEFT START BITS                                                                               XREXT
206   2    ELSE                                                                                                      XREXT
207   3       LOAD A WITH SOURCE                                                                                     XREXT
208   2    ENDIF                                                                                                     XREXT
209   2    SAVE A                                                                                                    XREXT
210   2    CLEAR B                                                                                                   XREXT
211   2    CONSTRUCT SHIFT                                                                                           XREXT
212   2    RESTORE A                                                                                                 XREXT
213   2    SHIFT BA LEFT LENGTH BITS                                                                                 XREXT
214   2    MOVE RESULT FROM B TO A                                                                                   XREXT
215   1 END XREXT                                                                                                    XREXT
```

```
217   1 CD0      FORTRAN CALLING PROCEDURE.                                                    XRE14
218   1 CD0                                                                                    XRE14
219   1 CD0   CALL XRE14 (REAL, ASCII)                                                        XRE14
220   1 CD0                                                                                    XRE14
221   1 C*********                                                                             XRE14
222   1 CD1                                                                                    XRE14
223   1 CD1      CONVERT A SINGLE PRECISION REAL NUMBER TO AN ASCII STRING IN                  XRE14
224   1 CD1      1PE14.6 FORMAT                                                                XRE14
225   1 CD1                                                                                    XRE14
226   1 C*********                                                                             XRE14
227   1 CD2      INPUT                                                                         XRE14
228   1 CD2   REAL   - TWO WORD SINGLE PRECISION REAL NUMBER TO BE CONVERTED                   XRE14
229   1 CD2                                                                                    XRE14
230   1 C*********                                                                             XRE14
231   1 CD3      OUTPUT                                                                        XRE14
232   1 CD3   ASCII  - SEVEN WORD ASCII CHARACTER STRING REPRESENTATION OF                     XRE14
233   1 CD3              'REAL'                                                                XRE14
234   1 CD3                                                                                    XRE14
235   1 C*********                                                                             XRE14
236   1 CD4      LOCAL                                                                         XRE14
237   1 CD4   R      - WORKING LOCATION FOR ABSOLUTE VALUE OR 'REAL' REPEATEDLY              XRE14
238   1 CD4              MODIFIED TO EXTRACT REMAINING DECIMAL DIGITS                          XRE14
239   1 CD4                                                                                    XRE14
240   1 C*********                                                                             XRE14
241   1 CD5      NOTES                                                                         XRE14
242   1 CD5   USES FLOAT, IAND, IFIX, IOR, KCVT, XREXT, XRSFL                                 XRE14
243   1 CD5                                                                                    XRE14
244   1 C*********                                                                             XRE14
245   1 *                                                                                      XRE14
246   1 *                                                                                      XRE14
247   1 *                                                                                      XRE14
248   1 *                                                                                      XRE14
249   1 BEGIN XRE14                                                                            XRE14
250   2    SET SIGN FIELD                                                                      XRE14
251   2    MOVE ABSOLUTE VALUE OF 'REAL' INTERNAL                                              XRE14
252   2    COMPUTE EXPONENT                                                                    XRE14
253   2    SET SIGN AND VALUE OF EXPONENT FIELD                                                XRE14
254   2    REDUCE VALUE TO RANGE OF 1 <= VALUE < 10                                            XRE14
255   2    EXTRACT FIRST DIGIT AND STORE FIELD                                                 XRE14
256   2    SET DECIMAL FIELD                                                                   XRE14
257   2    DO FOR NEXT THREE PAIRS OF DIGITS                                                   XRE14
258   3       MULTIPLY BY 100 TO EXTRACT PAIR                                                  XRE14
259   3       EXTRACT DIGITS AND STORE FIELD                                                   XRE14
260   2    ENDDO                                                                               XRE14
261   1 END XRE14                                                                              XRE14
```

```
263    1 CD0      FORTRAN CALLING PROCEDURE                                                                    XRI6
264    1 CD0                                                                                                   XRI6
265    1 CD0   CALL XRI6 (INTEGR, ASCII)                                                                       XRI6
266    1 CD0                                                                                                   XRI6
267    1 C*********                                                                                            XRI6
268    1 CD1                                                                                                   XRI6
269    1 CD1      CONVERT A SIXTEEN BIT SIGNED BINARY INTEGER TO A SIX CHARACTER                               XRI6
270    1 CD1      ASCII STRING                                                                                 XRI6
271    1 CD1                                                                                                   XRI6
272    1 C*********                                                                                            XRI6
273    1 CD2      INPUT                                                                                        XRI6
274    1 CD2   INTEGR - SIXTEEN BIT INTEGER TO BE CONVERTED                                                    XRI6
275    1 CD2                                                                                                   XRI6
276    1 C*********                                                                                            XRI6
277    1 CD3      OUTPUT                                                                                       XRI6
278    1 CD3   ASCII  - THREE WORD CHARACTER STRING REPRESENTATION OF 'INTEGR'                                  XRI6
279    1 CD3                                                                                                   XRI6
280    1 C*********                                                                                            XRI6
281    1 CD4      LOCAL                                                                                        XRI6
282    1 CD4   I      - INTERNAL LOCATION FOR 'INTEGR' REPEATEDLY MODIFIED TO                                  XRI6
283    1 CD4            PRODUCE 'ASCII'                                                                       XRI6
284    1 CD4   WRK    - SEVEN WORD WORKING BUFFER FOR CONSTRUCTION OF 'ASCII'                                  XRI6
285    1 CD4                                                                                                   XRI6
286    1 C*********                                                                                            XRI6
287    1 CD5      NOTES                                                                                        XRI6
288    1 CD5   USES XRMOV AND XRPCK                                                                            XRI6
289    1 CD5                                                                                                   XRI6
290    1 C*********                                                                                            XRI6
291    1 *                                                                                                     XRI6
292    1 *                                                                                                     XRI6
293    1 *                                                                                                     XRI6
294    1 *                                                                                                     XRI6
295    1 BEGIN XRI6                                                                                            XRI6
296    2    BLANK WORKING SPACE                                                                                XRI6
297    2    CONSTRUCT 'ASCII' LEAST SIGNIFICANT DIGITS FIRST USING REMAINDERING                                XRI6
298    2    SET SIGN OF 'INTEGR' IN 'ASCII'                                                                    XRI6
299    2    CALL XRPCK TO CONVERT FROM R1 TO A2 FORMAT                                                         XRI6
300    1 END XRI6                                                                                              XRI6
```

```
302   1 *D0      FORTRAN CALLING PROCEDURES                                                    XRLCK
303   1 *D0                                                                                    XRLCK
304   1 *D0   CALL XRLCK (RCODE)                                                               XRLCK
305   1 *D0   CALL XRULK (RCODE)                                                               XRLCK
306   1 *D0                                                                                    XRLCK
307   1 **********                                                                             XRLCK
308   1 *D1                                                                                    XRLCK
309   1 *D1      XRLCK AND XRULK PROVIDE A MECHANISM FOR SERIALIZING THE UPDATE                 XRLCK
310   1 *D1      OF FDS GLOBAL SYSTEM TABLES AND FILES.  THE RESOURCE NUMBER                   XRLCK
311   1 *D1      STORED IN THE XVSTB RESIDENT STATUS TABLE IS USED AS THE                      XRLCK
312   1 *D1      LOCKING MECHANISM                                                             XRLCK
313   1 *D1                                                                                    XRLCK
314   1 *********                                                                              XRLCK
315   1 *D2      INPUT                                                                         XRLCK
316   1 *D2   XVSTB RESOURCE NUMBER                                                            XRLCK
317   1 *D2                                                                                    XRLCK
318   1 **********                                                                             XRLCK
319   1 *D3      OUTPUT                                                                        XRLCK
320   1 *D3   RCODE  - INTEGER RETURN CODE (0 - SUCCESSFUL, 1 - FAILURE)                       XRLCK
321   1 *D3                                                                                    XRLCK
322   1 **********                                                                             XRLCK
323   1 *D4      LOCAL                                                                         XRLCK
324   1 *D4   STAT   - STATUS OF THIS COPY OF XEXEC USE OF XVSTB RN                            XRLCK
325   1 *D4            1 - RN LOCKED                                                           XRLCK
326   1 *D4            4 - RN UNLOCKED                                                         XRLCK
327   1 *D4                                                                                    XRLCK
328   1 **********                                                                             XRLCK
329   1 *D5      NOTES                                                                         XRLCK
330   1 *D5   USES .ENTR, RNRQ.                                                                XRLCK
331   1 *D5                                                                                    XRLCK
332   1 *D5                                                                                    XRLCK
333   1 *D5   THIS ROUTINE MAY NOT BE OVERLAYED                                                XRLCK
334   1 *D5                                                                                    XRLCK
335   1 **********                                                                             XRLCK
```

```
337   1 BEGIN XRLCK                                                                  XRLCK
338   2    SET REQUEST FOR LOCK FUNCTION                                             XRLCK
339   2    PERFORM RLOCK                                                             XRLCK
340   1 END XRLCK                                                                    XRLCK
341   1 BEGIN XRULK                                                                  XRLCK
342   2    SET REQUEST FOR UNLOCK FUNCTION                                           XRLCK
343   2    PERFORM RLOCK                                                             XRLCK
344   1 END XRULK                                                                    XRLCK
345   1 BEGIN RLOCK                                                                  XRLCK
346   2    IF REQUEST IS CONSISTENT WITH STATUS                                      XRLCK
347   2    THEN                                                                      XRLCK
348   3       SET NEW STATUS                                                         XRLCK
349   3       IF RN IN STB IS DEFINED, I.E., FDS HAS INITIALIZED SINCE IBL           XRLCK
350   3       THEN                                                                   XRLCK
351   4          CALL RNRQ TO ACCOMPLISH FUNCTION (WAIT IF NECESSARY ON LOCK)        XRLCK
352   3       ENDIF                                                                  XRLCK
353   3       CLEAR RETURN CODE                                                      XRLCK
354   2    ELSE                                                                      XRLCK
355   3       SET RETURN CODE                                                        XRLCK
356   2    ENDIF                                                                     XRLCK
357   1 END RLOCK                                                                    XRLCK
```

```
359  1 *DO      INTEGER FUNCTION                                                XRLOC
360  1 *DO                                                                      XRLOC
361  1 *DO      XRLOC(A)                                                        XRLOC
362  1 *DO                                                                      XRLOC
363  1 **********                                                              XRLOC
364  1 *D1                                                                      XRLOC
365  1 *D1      RETURN THE 16-BIT MAPPED ADDRESS OF A                           XRLOC
366  1 *D1                                                                      XRLOC
367  1 **********                                                              XRLOC
368  1 *D2        INPUT                                                         XRLOC
369  1 *D2    A   - VARIABLE, ROUTINE, ETC. FOR WHICH THE ADDRESS IS DESIRED    XRLOC
370  1 *D2                                                                      XRLOC
371  1 **********                                                              XRLOC
372  1 *D3        OUTPUT                                                        XRLOC
373  1 *D3    XRLOC  - 16-BIT ADDRESS OF A                                      XRLOC
374  1 *D3                                                                      XRLOC
375  1 **********                                                              XRLOC
376  1 *D5        NOTES                                                         XRLOC
377  1 *D5    NO EXTERNAL REFERENCES                                            XRLOC
378  1 *D5                                                                      XRLOC
379  1 **********                                                              XRLOC
380  1 *                                                                        XRLOC
381  1 *                                                                        XRLOC
382  1 *                                                                        XRLOC
383  1 *                                                                        XRLOC
384  1 BEGIN XRLOC                                                              XRLOC
385  2    TRANSFER CALLING SEQUENCE                                             XRLOC
386  2    LOAD THE ADDRESS OF THE CALLING PARAMETER                             XRLOC
387  1 END XRLOC                                                                XRLOC
```

```
389   1 *D0      FORTRAN CALLING PROCEDURE                                                    XRMOV
390   1 *D0                                                                                   XRMOV
391   1 *D0   CALL XRMOV (LENGTH, SOURCE, OBJECT)                                             XRMOV
392   1 *D0                                                                                   XRMOV
393   1 **********                                                                            XRMOV
394   1 *D1                                                                                   XRMOV
395   1 *D1      MOVE 'LENGTH' WORDS FROM 'SOURCE' TO 'OBJECT'                                XRMOV
396   1 *D1                                                                                   XRMOV
397   1 **********                                                                            XRMOV
398   1 *D2      INPUT                                                                        XRMOV
399   1 *D2   LENGTH - POSITIVE INTEGER INDICATING NUMBER OF WORDS TO MOVE                    XRMOV
400   1 *D2   SOURCE - ARRAY OF WORDS TO BE MOVED                                             XRMOV
401   1 *D2                                                                                   XRMOV
402   1 **********                                                                            XRMOV
403   1 *D3      OUTPUT                                                                       XRMOV
404   1 *D3   OBJECT - ARRAY RECEIVING MOVED WORDS                                           XRMOV
405   1 *D3                                                                                   XRMCV
406   1 **********                                                                            XRMOV
407   1 *D5      NOTES                                                                        XRMOV
408   1 *D5   USES .ENTR                                                                      XRMOV
409   1 *D5                                                                                   XRMOV
410   1 **********                                                                            XRMOV
411   1 *                                                                                     XRMOV
412   1 *                                                                                     XRMOV
413   1 *                                                                                     XRMOV
414   1 *                                                                                     XRMOV
415   1 BEGIN XRMOV                                                                           XRMOV
416   2    TRANSFER CALLING SEQUENCE                                                          XRMOV
417   2    INITIALIZE MOVE                                                                    XRMOV
418   2    MOVE LENGTH WORDS FROM SOURCE TO OBJECT                                            XRMOV
419   1 END XRMOV                                                                             XRMOV
```

```
421   1 CD0      FORTRAN CALLING PROCEDURE                                          XRMSG
422   1 CD0                                                                         XRMSG
423   1 CD0   CALL XRMSG (NUMBER, LOCATE, LENGTH, SOURCE)                          XRMSG
424   1 CD0                                                                         XRMSG
425   1 C*********                                                                  XRMSG
426   1 CD1                                                                         XRMSG
427   1 CD1      INSERT 'LENGTH' WORDS OF TEXT FROM 'SOURCE' INTO MESSAGE           XRMSG
428   1 CD1      'NUMBER' BEGINNING AFTER 'LOCATE' WORDS, CONCATENATE TO PREFIX,    XRMSG
429   1 CD1      TRUNCATE TO EIGHTY CHARACTERS AND OUTPUT TO USER TERMINAL          XRMSG
430   1 CD1                                                                         XRMSG
431   1 C*********                                                                  XRMSG
432   1 CD2      INPUT                                                              XRMSG
433   1 CD2   NUMBER - INTEGER MESSAGE NUMBER OF THE FORM 'ANN' WHERE               XRMSG
434   1 CD2            A  - AREA INDICATOR AS FOLLOWS                               XRMSG
435   1 CD2                 1 - AS                                                  XRMSG
436   1 CD2                 2 - XD                                                  XRMSG
437   1 CD2                 3 - XE                                                  XRMSG
438   1 CD2                 4 - XI                                                  XRMSG
439   1 CD2                 5 - XS                                                  XRMSG
440   1 CD2                 6 - XT                                                  XRMSG
441   1 CD2                 7 - XX                                                  XRMSG
442   1 CD2                 8 - XL                                                  XRMSG
443   1 CD2                 9 - DF                                                  XRMSG
444   1 CD2                10 - SC                                                  XRMSG
445   1 CD2            NN - MESSAGE NUMBER OR ZERO WHICH INDICATES ONLY 'LENGTH'    XRMSG
446   1 CD2                 WORDS OF 'SOURCE' TO BE OUTPUT                          XRMSG
447   1 CD2   LOCATE - INTEGER NUMBER OF WORDS OF MESSAGE TO PRECEED                XRMSG
448   1 CD2            'SOURCE' (NOT USED IF 'AN' OF 'NUMBER' IS ZERO)              XRMSG
449   1 CD2   LENGTH - INTEGER NUMBER OF WORDS OF 'SOURCE' TO BE INSERTED INTO      XRMSG
450   1 CD2            MESSAGE.  ZERO INDICATES NO INSERTION                        XRMSG
451   1 CD2   SOURCE - ARRAY OF CHARACTERS TO BE INSERTED INTO MESSAGE (NOT USED    XRMSG
452   1 CD2            IF 'LENGTH' IS ZERO)                                         XRMSG
453   1 CD2                                                                         XRMSG
454   1 CD2      COMMON                                                             XRMSG
455   1 CD2   LU     - USERS LOGICAL UNIT NUMBER                                    XRMSG
456   1 CD2                                                                         XRMSG
457   1 C*********                                                                  XRMSG
458   1 CD3      OUTPUT                                                             XRMSG
459   1 CD3   UP TO EIGHTY CHARACTER LINE OF TEXT TO UNIT 'LU' OF THE FORM          XRMSG
460   1 CD3                                                                         XRMSG
461   1 CD5   *** AANN  MESSAGE(1-LOCATE) SOURCE(1-LENGTH) REMAINDER OF MESSAGE    XRMSG
462   1 CD5                                                                         XRMSG
463   1 C*********                                                                  XRMSG
464   1 CD5      NOTES                                                              XRMSG
465   1 CD5   USES FDS SYSTEM MESSAGE FILE ]XRMSG                                   XRMSG
466   1 CD5   USES CLOSE, EXEC, IAND, KCVT, OPEN, READF, XRMOV, XUDBG               XRMSG
467   1 CD5                                                                         XRMSG
468   1 C*********                                                                  XRMSG
```

```
470   1 BEGIN XRMSG                                                                                                    XRMSG
471   2    SEPARATE NUMBER INTO AREA AND MESSAGE NUMBER                                                                XRMSG
472   2    SET NUMBER IN PREFIX                                                                                        XRMSG
473   2    READ MESSAGE DIRECTORY RECORD                                                                               XRMSG
474   2    IF AREA VALID                                                                                               XRMSG
475   2    THEN                                                                                                        XRMSG
476   3       SET AREA CODE IN PREFIX                                                                                  XRMSG
477   3       IF MESSAGE NUMBER > 0                                                                                    XRMSG
478   3       THEN                                                                                                     XRMSG
479   4          IF VALID MESSAGE NUMBER                                                                               XRMSG
480   4          THEN                                                                                                  XRMSG
481   5             COMPUTE MESSAGE RECORD NUMBER                                                                      XRMSG
482   5             READ RECORD                                                                                        XRMSG
483   5             CALL XRMOV TO MOVE LOCATE WORDS FROM RECORD INTO BUFFER                                            XRMSG
484   4          ELSE                                                                                                  XRMSG
485   4          EXIT TO :ERROR:                                                                                       XRMSG
486   4          ENDIF                                                                                                 XRMSG
487   3       ENDIF                                                                                                    XRMSG
488   3       CALL XRMOV TO MOVE LENGTH WORDS FROM SOURCE INTO BUFFER (MAX OF 40 TOTAL WORDS)                          XRMSG
489   3       IF MESSAGE NUMBER > 0                                                                                    XRMSG
490   3       THEN                                                                                                     XRMSG
491   4          CALL XRMOV TO MOVE REMAINING RECORD INTO BUFFER (MAX OF 40 TOTAL WORDS)                               XRMSG
492   3       ENDIF                                                                                                    XRMSG
493   2    ELSE                                                                                                        XRMSG
494   3       SET AREA IN PREFIX                                                                                       XRMSG

495   3       :ERROR:  CALL XRMOV TO MOVE 'XRMSG ERROR' INTO BUFFER                                                    XRMSG
496   3       CALL XRMOV TO MOVE LENGTH WORDS OF SOURCE INTO BUFFER (MAX OF 40 TOTAL WORDS)                            XRMSG
497   2    ENDIF                                                                                                       XRMSG
498   2    OUTPUT BUFFER TO USER'S TERMINAL                                                                            XRMSG
499   2    IF DEBUG IS REQUESTED                                                                                       XRMSG
500   2    THEN                                                                                                        XRMSG
501   3       CALL XUDBG                                                                                               XRMSG
502   2    ENDIF                                                                                                       XRMSG
503   1 END XRMSG                                                                                                      XRMSG
```

```
505   1 *D0       INTEGER FUNCTION                                                                XRNX8
506   1 *D0                                                                                       X2NX5
507   1 *D0    XRNX8(BIT, BITNUM, STRING)                                                         XRNX8
508   1 *D0                                                                                       XRNX8
509   1 **********                                                                               XRNX8
510   1 *D1                                                                                       XRNX8
511   1 *D1    SEARCH BIT STRING 'STRING' BEGINNING AT BIT NUMBER 'BITNUM' FOR                    XRNX8
512   1 *D1    THE NEXT OCCURANCE OF BIT SETTING 'BIT'                                            XRNX8
513   1 *D1                                                                                       XRNX8
514   1 **********                                                                               XRNX8
515   1 *D2       INPUT                                                                           XRNX8
516   1 *D2    BIT    - INTEGER VALUE THE LAST BIT OF WHICH IS TO BE COMPARED TO                  XRNX8
517   1 *D2             BITS OF 'STRING' FOR A MATCH                                           XRNX8
518   1 *D2    BITNUM - UNSIGNED SIXTEEN BIT INTEGER INDICATING THE BIT NUMBER IN               XRNX8
519   1 *D2             'STRING' WITH WHICH TO BEGIN THE SEARCH (FIRST BIT OF                  XRNX8
520   1 *D2             'STRING' IS BIT NUMBER ZERO)                                           XRNX8
521   1 *D2    STRING - BIT STRING TO BE SEARCHED.  SEARCH WILL CONTINUE THROUGH               XRNX8
522   1 *D2             MEMORY UNTIL A VALUE OF 'BIT' IS DETECTED                              XRNX8
523   1 *D2                                                                                    XRNX8
524   1 **********                                                                             XRNX8
525   1 *D3       OUTPUT                                                                       XRNX8
526   1 *D3    FUNCTION VALUE - BIT NUMBER OF NEXT OCCURANCE OF 'BIT' >= 'BITNUM'              XRNX8
527   1 *D3                                                                                    XRNX8
528   1 **********                                                                             XRNX8
529   1 *D5       NOTES                                                                        XRNX8
530   1 *D5    USES .ENTR                                                                      XRNX8
531   1 *D5                                                                                    XRNX8
532   1 CD5    SEARCH WILL NOT TERMINATE UNTIL A VALUE OF 'BIT' IS DETECTED OR                 XRNX8
533   1 CD5    ALL OF MEMORY HAS BEEN EXAMINED.  THUS, APPROPRIATE STEPS SHOULD                XRNX8
534   1 CD5    BE TAKEN TO FORCE A MATCH AT THE END OF THE STRING.                             XRNX8
535   1 *D5                                                                                    XRNX8
536   1 *D5    A MAXIMUM BIT STRING LENGTH OF 65535 BITS (4096 WORDS) CAN BE                   XRNX8
537   1 *D5    MEANINGFULLY ACCOMMODATED.                                                      XRNX8
538   1 *D5                                                                                    XRNX8
539   1 **********                                                                             XRNX8
```

```
541    1 BEGIN XRNXB                                                                    XRNXB
542    2    COMPUTE STARTING WORD NUMBER                                                XRNXB
543    2    COMPUTE STARTING BIT NUMBER                                                 XRNXB
544    2    LOAD STARTING WORD                                                          XRNXB
545    2    SHIFT WORD UNTIL STARTING BIT NUMBER IS IN SIGN BIT                         XRNXB
546    2    INITIALIZE BIT COUNT                                                        XRNXB
547    2    DO WHILE SIGN BIT IS NOT EQUAL TO BIT                                       XRNXB
548    3       IF WORD NOT FINISHED                                                     XRNXB
549    3       THEN                                                                     XRNXB
550    4          SHIFT WORD LEFT                                                       XRNXB
551    4          INCREMENT BIT COUNT                                                   XRNXB
552    3       ELSE                                                                     XRNXB
553    4          DO UNTIL WORD WITH SOME 'BIT' VALUE FOUND                             XRNXB
554    5             LOAD NEXT WORD                                                     XRNXB
555    4          ENDDO                                                                 XRNXB
556    4          CLEAR BIT COUNT                                                       XRNXB
557    3       ENDIF                                                                    XRNXB
558    2    ENDDO                                                                       XRNXB
559    2    RETURN VALUE OF MATCHING BIT NUMBER                                         XRNXB
560    1 END XRNXB                                                                      XRNXB
```

```
562   1 *00      FORTRAN CALLING PROCEDURE                                                    XR06
563   1 *00                                                                                   XR06
564   1 *00   CALL XR06 (BINARY, OCTAL)                                                      XR06
565   1 *00                                                                                   XR06
566   1 **********                                                                            XR06
567   1 *01                                                                                   XR06
568   1 *01   CONVERT A WORD FROM BINARY TO SIX CHARACTER OCTAL REPRESENTATION                XR06
569   1 *01                                                                                   XR06
570   1 **********                                                                            XR06
571   1 *02      INPUT                                                                        XR06
572   1 *02   BINARY - BINARY WORD TO BE CONVERTED                                           XR06
573   1 *02                                                                                   XR06
574   1 **********                                                                            XR06
575   1 *03      OUTPUT                                                                       XR06
576   1 *03   OCTAL  - THREE WORD ARRAY CONTAINING OCTAL REPRESENTATION OF                   XR06
577   1 *03            'BINARY' IN ASCII FORMAT (06)                                        XR06
578   1 *03                                                                                   XR06
579   1 **********                                                                            XR05
580   1 *05      NOTES                                                                        XR06
581   1 *05   USES .ENTR                                                                     XR06
582   1 *05                                                                                   XR06
583   1 **********                                                                            XR06
584   1 *                                                                                     XR06
585   1 *                                                                                     XR06
586   1 *                                                                                     XR06
587   1 *                                                                                     XR06
588   1 BEGIN XR06                                                                            XR06
589   2    TRANSFER CALLING SEQUENCE                                                          XR06
590   2    FORM 18-BIT WORD USING 8 AND 2 MOST SIGNIFICANT BITS OF A                          XR06
591   2    SET BYTE FLAG HIGH                                                                 XR06
592   2    DO FOR EACH PAIR OF OCTAL DIGITS (3)                                               XR06
593   3       SET PREFIX BITS (000000B)                                                       XR06
594   3       SHIFT IN DIGIT                                                                  XR06
595   3       IF BYTE FLAG SET HIGH                                                           XR06
596   3       THEN                                                                            XR06
597   4          SHIFT FOR ACCOMODATION OF LOW BYTE                                           XR06
598   3       ELSE                                                                            XR06
599   4          STORE PAIR OF DIGITS IN OCTAL(I)                                             XR06
600   4          CLEAR FOR NEXT PAIR OF DIGITS                                                XR06
601   3       ENDIF                                                                           XR06
602   3       FLIP BYTE FLAG                                                                  XR06
603   2    ENDDO                                                                              XR06
604   1 END XR06                                                                              XR06
```

```
606   1 *D0       FORTRAN CALLING PROCEDURE                                                    XRPCK
607   1 *D0                                                                                    XRPCK
608   1 *D0    CALL XRPCK (LENGTH, UNPKED, PACKED)                                             XRPCK
609   1 *D0                                                                                    XRPCK
610   1 **********                                                                             XRPCK
611   1 *D1                                                                                    XRPCK
612   1 *D1       CONVERT 'LENGTH' CHARACTERS OF 'UNPKED' FROM R1 FORMAT TO A2                 XRPCK
613   1 *D1       FORMAT AND STORE IN 'PACKED'                                                 XRPCK
614   1 *D1                                                                                    XRPCK
615   1 **********                                                                             XRPCK
616   1 *D2       INPUT                                                                        X PCK
617   1 *D2    LENGTH - POSITIVE INTEGER NUMBER OF CHARACTERS IN UNPKED                        XRPCK
618   1 *D2    UNPKED - ARRAY OF CHARACTERS IN R1 FORMAT                                       XRPCK
619   1 *D2                                                                                    XRPCK
620   1 **********                                                                             XRPCK
621   1 *D3       OUTPUT                                                                       XRPCK
622   1 *D3    PACKED - ARRAY OF (LENGTH+1)/2 WORDS IN A2 FORMAT.  IF LENGTH IS                XRPCK
623   1 *D3             ODD, THE LAST WORD WILL BE BLANK FILLED.                               XRPCK
624   1 *D3                                                                                    XRPCK
625   1 **********                                                                             XRPCK
626   1 *D5       NOTES                                                                        XRPCK
627   1 *D5    USES .ENTR                                                                      XRPCK
628   1 *D5                                                                                    XRPCK
629   1 **********                                                                             XRPCK
630   1 *                                                                                      XRPCK
631   1 *                                                                                      XRPCK
632   1 *                                                                                      XRPCK
633   1 *                                                                                      XRPCK
634   1 BEGIN XRPCK                                                                            XRPCK
635   2    TRANSFER CALLING SEQUENCE                                                           XRPCK
636   2    SET BYTE FLAG FOR HIGH BYTE                                                         XRPCK
637   2    INITIALIZE PACKED POINTER                                                           XRPCK
638   2    DO FOR EACH CHARACTER IN UNPKED                                                     XRPCK
639   3       IF BYTE FLAG SET HIGH                                                            XRPCK
640   3       THEN                                                                             XRPCK
641   4          LOAD A WITH NEXT WORD OF UNPKED                                               XRPCK
642   4          SHIFT CHARACTER INTO HIGH BYTE                                                XRPCK
643   3       ELSE                                                                             XRPCK
644   4          INCLUSIVE OR NEXT WORD OF UNPKE  INTO A                                       XRPCK
645   4          STORE A IN PACKED                                                             XRPCK
646   4          INCREMENT POINTER                                                             XRPCK
647   3       ENDIF                                                                            XRPCK
648   3       FLIP BYTE FLAG                                                                   XRPCK
649   2    ENDDO                                                                               XRPCK
650   2    IF BYTE FLAG SET LOW                                                                XRPCK
651   2    THEN                                                                                XRPCK
652   3       INCLUSIVE OR BLANK INTO LOW BYTE                                                 XRPCK
653   2    ENDIF                                                                               XRPCK
654   2    STORE A IN PACKED                                                                   XRPCK
655   1 END XRPCK                                                                              XRPCK
```

```
657   1 *D0      FORTRAN CALLING PROCEDURE                                                                XRQFN
658   1 *D0                                                                                               XRQFN
659   1 *D0   CALL XRQFN (PREFIX, NAME4, NAME6)                                                           XRQFN
660   1 *D0                                                                                               XRQFN
661   1 **********                                                                                        XRQFN
662   1 *D1                                                                                               XRQFN
663   1 *D1      XRQFN BUILDS A QUALIFIED FILE NAME OF UPTO SIX CHARACTERS IN                             XRQFN
664   1 *D1      LENGTH BY PREFIXING THE INPUT ONE TO FOUR CHARACTER NAME WITH                            XRQFN
665   1 *D1      THE PREFIX CHARACTER AND APPENDING A USER QUALIFIER CODE TO THE                          XRQFN
666   1 *D1      END                                                                                      XRQFN
667   1 *D1                                                                                               XRQFN
668   1 **********                                                                                        XRQFN
669   1 *D2      INPUT                                                                                    XRQFN
670   1 *D2   PREFIX - FILE TYPE PREFIX STORED IN R1 FORMAT                                               XRQFN
671   1 *D2   NAME4  - ONE TO FOUR CHARACTER PACKED NAME TO BE QUALIFIED                                  XRQFN
672   1 *D2   COMMON XE - QUAL                                                                            XRQFN
673   1 *D2                                                                                               XRQFN
674   1 **********                                                                                        XRQFN
675   1 *D3      OUTPUT                                                                                   XRQFN
676   1 *D3   NAME6  - THREE TO SIX CHARACTER PACKED QUALIFIED NAME                                       XRQFN
677   1 *D3                                                                                               XRQFN
678   1 **********                                                                                        XRQFN
679   1 *D5      NOTES                                                                                    XRQFN
680   1 *D5   USES .ENTR                                                                                  XRQFN
681   1 *D5                                                                                               XRQFN
682   1 **********                                                                                        XRQFN
683   1 *                                                                                                 XRQFN
684   1 *                                                                                                 XRQFN
685   1 *                                                                                                 XRQFN
686   1 *                                                                                                 XRQFN
687   1 BEGIN XRQFN                                                                                       XRQFN
688   2    STORE PREFIX IN FIRST POSITION OF INTERNAL CHARACTER STRING                                     XRQFN
689   2    MOVE NAME4 INTO NEXT FOUR POSITIONS                                                            XRQFN
690   2    STORE BLANK IN SIXTH POSITION                                                                  XRQFN
691   2    LOCATE FIRST BLANK CHARACTER                                                                   XRQFN
692   2    REPLACE BLANK WITH USER ID CHARACTER (QUAL)                                                    XRQFN
693   2    MOVE QUALIFIED NAME TO NAME6                                                                   XRQFN
694   1 END XRQFN                                                                                         XRQFN
```

```
696   1 *D0      FORTRAN CALLING PROCEDURE                                                    XRSET
697   1 *D0                                                                                   XRSET
698   1 *D0   CALL XRSET (START, LENGTH, SOURCE, OBJECT)                                      XRSET
699   1 *D0                                                                                   XRSET
700   1 **********                                                                            XRSET
701   1 *D1                                                                                   XRSET
702   1 *D1      REPLACE 'LENGTH' BITS OF 'OBJECT', BEGINNING WITH BIT 'START',               XRSET
703   1 *D1      WITH THE RIGHT MOST 'LENGTH' BITS OF 'SOURCE'                                XRSET
704   1 *D1                                                                                   XRSET
705   1 **********                                                                            XRSET
706   1 *D2      INPUT                                                                        XRSET
707   1 *D2   START  - INTEGER INDICATING LEFT MOST BIT OF FIELD TO BE REPLACED              XRSET
708   1 *D2            (SIGN BIT = 0)                                                       XRSET
709   1 *D2   LENGTH - POSITIVE INTEGER SIZE OF FIELD BEING REPLACED                        XRSET
710   1 *D2   SOURCE - WORD CONTAINING REPLACEMENT FIELD RIGHT ADJUSTED                      XRSET
711   1 *D2                                                                                   XRSET
712   1 **********                                                                            XRSET
713   1 *D3      OUTPUT                                                                       XRSET
714   1 *D3   OBJECT - WORD INTO WHICH FIELD IS TO BE INSERTED                               XRSET
715   1 *D3                                                                                   XRSET
716   1 **********                                                                            XRSET
717   1 *D5      NOTES                                                                        XRSET
718   1 *D5   USES .ENTR                                                                      XRSET
719   1 *D5                                                                                   XRSET
720   1 **********                                                                            XRSET
721   1 *                                                                                     XRSET
722   1 *                                                                                     XRSET
723   1 *                                                                                     XRSET
724   1 *                                                                                     XRSET
725   1 BEGIN XRSET                                                                           XRSET
726   2    TRANSFER CALLING SEQUENCE                                                        XRSET
727   2    CONSTRUCT SHIFT INSTRUCTIONS                                                     XRSET
728   2    SHIFT LENGTH BITS OF SOURCE INTO HIGH END OF CLEARED REGISTER                    XRSET
729   2    SHIFT REGISTER RIGHT START BITS TO PROPERLY POSITION FIELD                       XRSET
730   2    CONSTRUCT MASK AND CLEAR FIELD OF OBJECT                                         XRSET
731   2    INCLUSIVE OR POSITIONED SOURCE FIELD INTO OBJECT                                 XRSET
732   1 END XRSET                                                                           XRSET
```

```
734   1 *D0       INTEGER FUNCTIONS                                                                                    XRSFL
735   1 *D0                                                                                                            XRSFL
736   1 *D0    XRSFL(COUNT, SOURCE)                                                                                    XRSFL
737   1 *D0    XRSFR(COUNT, SOURCE)                                                                                    XRSFL
738   1 *D0                                                                                                            XRSFL
739   1 **********                                                                                                     XRSFL
740   1 *D1                                                                                                            XRSFL
741   1 *D1       SHIFT 'SOURCE' LEFT/RIGHT LOGICALLY 'COUNT' BITS                                                     XRSFL
742   1 *D1                                                                                                            XRSFL
743   1 **********                                                                                                     XRSFL
744   1 *D2       INPUT                                                                                                XRSFL
745   1 *D2    COUNT  - POSITIVE INTEGER SPECIFYING NUMBER OF BITS TO SHIFT                                            XRSFL
746   1 *D2    SOURCE - WORD TO BE SHIFTED                                                                             XRSFL
747   1 *D2                                                                                                            XRSFL
748   1 **********                                                                                                     XRSFL
749   1 *D5       NOTES                                                                                                XRSFL
750   1 *D5    USES .ENTR                                                                                              XRSFL
751   1 *D5                                                                                                            XRSFL
752   1 **********                                                                                                     XRSFL
753   1 *                                                                                                              XRSFL
754   1 *                                                                                                              XRSFL
755   1 *                                                                                                              XRSFL
756   1 *                                                                                                              XRSFL
757   1 BEGIN XRSFL                                                                                                    XRSFL
758   2    SET FOR LEFT SHIFT                                                                                          XRSFL
759   2    PERFORM SHIFT (FLAG, COUNT, SOURCE)                                                                         XRSFL
760   1 END XRSFL                                                                                                      XRSFL
761   1 BEGIN XRSFR                                                                                                    XRSFL
762   2    SET FOR RIGHT SHIFT                                                                                         XRSFL
763   2    PERFORM SHIFT (FLAG, COUNT, SOURCE)                                                                         XRSFL
764   1 END XRSFR                                                                                                      XRSFL
765   1 BEGIN SHIFT                                                                                                    XRSFL
766   2    TRANSFER CALLING SEQUENCE                                                                                   XRSFL
767   2    CONSTRUCT SHIFT INSTRUCTION                                                                                 XRSFL
768   2    LOAD A WITH SOURCE                                                                                          XRSFL
769   2    CLEAR B                                                                                                     XRSFL
770   2    SHIFT BA AS SPECIFIED                                                                                       XRSFL
771   1 END SHIFT                                                                                                      XRSFL
```

```
773   1 *DO      FORTRAN CALLING PROCEDURE                                                    XRUNQ
774   1 *DO                                                                                   XRUNQ
775   1 *DO   CALL XRUNQ (NAME6, NAME4)                                                       XRUNQ
776   1 *DO                                                                                   XRUNQ
777   1 **********                                                                            XRUNQ
778   1 *D1                                                                                   XRUNQ
779   1 *D1      XRUNQ REMOVES THE PREFIX AND SUFFIX QUALIFING CHARACTERS FROM                XRUNQ
780   1 *D1      A SIX CHARACTER FILE NAME                                                    XRUNQ
781   1 *D1                                                                                   XRUNQ
782   1 **********                                                                            XRUNQ
783   1 *D2      INPUT                                                                        XRUNQ
784   1 *D2   NAME6  - THREE TO SIX CHARACTER PACKED QUALIFIED NAME                           XRUNQ
785   1 *D2                                                                                   XRUNQ
786   1 **********                                                                            XRUNQ
787   1 *D3      OUTPUT                                                                       XRUNQ
788   1 *D3   NAME4  - ONE TO FOUR CHARACTER PACKED NAME WITH PRIFIX AND SUFFIX               XRUNQ
789   1 *D3            REMOVED                                                              XRUNQ
790   1 *D3                                                                                   XRUNQ
791   1 **********                                                                            XRUNQ
792   1 *D5      NOTES                                                                        XRUNQ
793   1 *D5   USES .ENTR                                                                      XRUNQ
794   1 *D5                                                                                   XRUNQ
795   1 **********                                                                            XRUNQ
796   1 *                                                                                     XRUNQ
797   1 *                                                                                     XRUNQ
798   1 *                                                                                     XRUNQ
799   1 *                                                                                     XRUNQ
800   1 BEGIN XRUNQ                                                                           XRUNQ
801   2    MOVE CHARACTERS 2-5 OF NAME6 INTO NAME4                                          XRUNQ
802   2    IF SIXTH CHARACTER IS BLANK, THEN                                                XRUNQ
803   3       LOCATE LAST NON-BLANK CHARACTER OF NAME4                                      XRUNQ
804   3       BLANK THAT CHARACTER                                                          XRUNQ
805   2    ENDIF                                                                            XRUNQ
806   1 END XRUNQ                                                                           XRUNQ
```

```
808  1 *DO      FORTRAN CALLING PROCEDURE                                    XRUPK
809  1 *DO                                                                   XRUPK
810  1 *DO    CALL XRUPK (LENGTH, PACKED, UNPKED, COUNT)                     XRUPK
811  1 *DO                                                                   XRUPK
812  1 **********                                                            XRUPK
813  1 *D1                                                                   XRUPK
814  1 *D1      CONVERT 'LENGTH' WORDS OF 'PACKED' FROM A2 FORMAT TO R1 FORMAT,  XRUPK
815  1 *D1      REMOVING BLANKS NOT DELIMITED BY QUOTE MARKS, AND RETURN IN  XRUPK
816  1 *D1      'UNPKED' WITH THE NON-BLANK CHARACTER COUNT IS 'COUNT'       XRUPK
817  1 *D1                                                                   XRUPK
818  1 **********                                                            XRUPK
819  1 *D2      INPUT                                                        XRUPK
820  1 *D2    LENGTH - POSITIVE INTEGER NUMBER OF WORDS IN PACKED            XRUPK
821  1 *D2    PACKED - ARRAY OF CHARACTER DATA I' A2 FORMAT                  XRUPK
822  1 *D2                                                                   XRUPK
823  1 **********                                                            XRUPK
824  1 *D3      OUTPUT                                                       XRUPK
825  1 *D3    UNPKED - ARRAY OF NON-BLANK CHARACTERS IN R1 FORMAT            XRUPK
826  1 *D3    COUNT  - NUMBER OF CHARACTERS IN UNPKED                        XRUPK
827  1 *D3                                                                   XRUPK
828  1 **********                                                            XRUPK
829  1 *D5      NOTES                                                        XRUPK
830  1 *D5    USES .ENTR                                                     XRUPK
831  1 *D5    IF PACKED AND UNPKED ARE THE SAME ADDRESS SPACE UNPKED MAY OVERLAY  XRUPK
832  1 *D5    PACKED                                                         XRUPK
833  1 *D5                                                                   XRUPK
834  1 *D5    ** CAUTION: XRUPK CANNOT HANDLE QUOTE MARKS WITHIN CHARACTER   XRUPK
835  1 *D5    STRINGS.                                                       XRUPK
836  1 *D5                                                                   XRUPK
837  1 **********                                                            XRUPK
```

```
839   1 BEGIN XRUPK                                                    XRUPK
840   2    TRANSFER CALLING SEQUENCE                                   XRUPK
841   2    TURN ON BLANK REMOVAL                                       XRUPK
842   2    INITIALIZE COUNT                                            XRUPK
843   2    DO FOR EACH WORD OF PACKED                                  XRUPK
844   3       LOAD A WITH NEXT WORD                                    XRUPK
845   3       ROTATE A 8 BITS                                          XRUPK
846   3       DO FOR EACH BYTE OF WORD                                 XRUPK
847   4          AND OFF HIGH BYTE                                     XRUPK
848   4          IF A = QUOTE MARK                                     XRUPK
849   4          THEN                                                  XRUPK
850   5             CHANGE BLANK REMOVAL OPTION                        XRUPK
851   4          ENDIF                                                 XRUPK
852   4          IF BLANK REMOVAL IS ON                                XRUPK
853   4          THEN                                                  XRUPK
854   5             IF A NOT = BLANK                                   XRUPK
855   5             THEN                                               XRUPK
856   6                INCREMENT COUNT                                 XRUPK
857   6                STORE A IN UNPKED                               XRUPK
858   5             ENDIF                                              XRUPK
859   4          ELSE                                                  XRUPK
860   5             INCREMENT COUNT                                    XRUPK
861   5             STORE A IN UNPACKED                                XRUPK
862   4          ENDIF                                                 XRUPK
863   4          RELOAD A WITH WORD                                    XRUPK
864   3       ENDDO                                                    XRUPK
865   2    ENDDO                                                       XRUPK
866   2    RETURN VALUE OF COUNT                                       XRUPK
867   1 END XRUPK                                                      XRUPK
```

```
869   1 *D0      FORTRAN CALLING PROCEDURE                                                    XR1SP
870   1 *D0                                                                                   XR1SP
871   1 *D0    CALL XR1SP (CHSTR, LNGTH)                                                      XR1SP
872   1 *D0                                                                                   XR1SP
873   1 **********                                                                           XR1SP
874   1 *D0                                                                                   XR1SP
875   1 *D1      XR1SP COMPACTS A CHARACTER STRING STORED IN A2 FORMAT SUCH THAT               XR1SP
876   1 *D1      NO MORE THAN ONE CONSECUTIVE BLANK APPEARS EXCEPT BETWEEN                    XR1SP
877   1 *D1      QUOTATION MARKS.                                                             XR1SP
878   1 *D1                                                                                   XR1SP
879   1 **********                                                                           XR1SP
880   1 *D2      INPUT                                                                        XR1SP
881   1 *D2    CHSTR  - CHARACTER STRING OF LNGTH WORDS IN A2 FORMAT                          XR1SP
882   1 *D2    LNGTH  - LENGTH OF CHSTR IN WORDS                                              XR1SP
883   1 *D2                                                                                   XR1SP
884   1 **********                                                                           XR1SP
885   1 *D3      OUTPUT                                                                       XR1SP
886   1 *D3    CHSTR  - CHARACTER STRING OF LNGTH WORDS IN A2 FORMAT WITH                     XR1SP
887   1 *D3             DUPLICATE BLANKS REMOVED                                            XR1SP
888   1 *D3    LNGTH  - NEW LENGTH OF CHSTR IN WORDS                                          XR1SP
889   1 *D3                                                                                   XR1SP
890   1 **********                                                                           XR1SP
891   1 *D5      EXTERNAL REFERENCES                                                          XR1SP
892   1 *D5    .ENTR                                                                          XR1SP
893   1 *D5                                                                                   XR1SP
894   1 **********                                                                           XR1SP
895   1 *D6      SPECIAL REMARKS                                                              XR1SP
896   1 *D5    THIS ROUTINE ASSUMES                                                           XR1SP
897   1 *D6      1.  THE INPUT CHARACTER STRING HAS A POSITIVE LENGTH >0                      XR1SP
898   1 *D6      2.  SPECIAL HANDLING OF STRINGS WITHIN QUOTATION MARKS DOES NOT              XR1SP
899   1 *D6          BEGIN UNTIL THE FIRST OCCURANCE OF QUOTATION MARKS ON EACH               XR1SP
900   1 *D6          ENTRY                                                                    XR1SP
901   1 *D6      3.  THE FIRST/LAST CHARACTER IN A CHARACTER STRING HAS THE                   XR1SP
902   1 *D6          HIGH ORDER BIT OF THE QUOTE CHARACTER 'ON' TO INDICATE                   XR1SP
903   1 *D6          BEGIN/END OF A CHARACTER STRING.THIS BIT IS TURNED 'OFF'                 XR1SP
904   1 *D6          BY THIS PROGRAM PRIOR TO EXIT.THIS CHANGE MADE TO                        XR1SP
905   1 *D6          INCORPORATE QUOTE MARKS W/I CHARACTER STRINGS.                           XR1SP
906   1 *D6          OTHER PROCESSORS CHANGED FOR THIS MODIFICATION WERE:                     XR1SP
907   1 *D6            A. XTLAN,XPRDM                                                         XR1SP
908   1 *D6            B. XILSS/XZLSS                                                         XR1SP
909   1 *D6                                                                                   XR1SP
910   1 **********                                                                           XR1SP
```

```
912    1 BEGIN XR1SP                                                                                                    XR1SP
913    2    CONVERT WORD COUNT INTO CHARACTER COUNT                                                                     XR1SP
914    2    SET STORE INDEX TO FIRST POSITION IN CHARACTER STRING                                                       XR1SP
915    2    CLEAR BLANK FLAG                                                                                            XR1SP
916    2    CLEAR CHARACTER STRING FLAG                                                                                 XR1SP
917    2    DO FOR EACH CHARACTER IN STRING                                                                             XR1SP
918    3       IF CHARACTER IS SPECIAL CHARACTER INDICATING CHARACTER STRING , THEN                                      XR1SP
919    4          REPLACE CHARACTER STRING INDICATOR WITH QUOTE MARK                                                    XR1SP
920    4          FLIP CHARACTER STRING FLAG                                                                           XR1SP
921    4          CLEAR BLANK FLAG                                                                                     XR1SP
922    3       ELSE                                                                                                     XR1SP
923    4          IF CHARACTER STRING FLAG IS CLEAR, THEN                                                           XR1SP
924    5             IF CHARACTER IS A BLANK, THEN                                                                  XR1SP
925    6                IF BLANK FLAG IS SET (AT LEAST ONE PREDEDING BLANK), THEN                                   XR1SP
926    7                   SKIP THIS CHARACTER (EXIT TO ENDDO)                                                      XR1SP
927    6                ELSE                                                                                        XR1SP
928    7                   SET BLANK FLAG                                                                           XR1SP
929    6                ENDIF                                                                                       XR1SP
930    5             ELSE                                                                                           XR1SP
931    6                CLEAR BLANK FLAG                                                                            XR1SP
932    5             ENDIF                                                                                          XR1SP
933    4          ENDIF                                                                                             XR1SP
934    3       ENDIF                                                                                                XR1SP
935    3       STORE CHARACTER AT INDEXED POSITION                                                                  XR1SP
936    3       INCREMENT STORE INDEX                                                                                XR1SP
937    2    ENDDO                                                                                                   XR1SP
938    2    IF NUMBER OF CHARACTERS STORED IS ODD                                                                   XR1SP
939    3       STORE ONE MORE BLANK                                                                                 XR1SP
940    2    ENDIF                                                                                                   XR1SP
941    2    CONVERT STORE INDEX TO WORD COUNT AND RETURN                                                            XR1SP
942    1 END XR1SP                                                                                                  XR1SP
```

```
 SYMBOL DEFINITION TABLE
:ERROR :    495
 RLOCK      345
 SHIFT      765
 TRACE      963
 XRBIT       33
 XRCPR       73
 XRD18      112
 XREQ       155
 XREXT      199
 XRE14      249
 XRI6       295
 XRLCK      337
 XRLOC      384
 XRMOV      415
 XRMSG      470
 XRNXB      541
 XRO6       588
 XRPCK      634
 XRQFN      687
 XRSET      725
 XRSFL      757
 XRSFR      761
 XRULK      341
 XRUNQ      800
 XRUPK      839
 XR1SP      912

@XQT F.PDLIST
```

```
      1 CD0
      1 CD0       FORTRAN CALLING PROCEEDURE
      1 CD0
      1 CD0     CALL XRLDS(XSEQE)
      1 CD0
      1 C*********
      1 CD1
      1 CD1     XSEQE IS THE MAIN ROUTINE OF THE SEQUENCE TABLE EDITOR
      1 CD1
      1 CD2
      1 CD2       INPUT
      1 CD2
      1 CD2     COMMON XE - LU
      1 CD2
      1 CD2     COMMON XB - DEBUG,  IRETC,  NEWTAB, NUMENT, PRMLEN
      1 CD2                 PROMPT, PRMTMD, WKBUF
      1 CD2
      1 C*********
      1 CD3
      1 CD3       OUTPUT
      1 CD3
      1 CD3     COMMON XE - REQBUF, REQPTR
      1 CD3
      1 CD3     COMMON XB - NUMENT, PRMLEN, PRMTMD, PROMPT, WKBUF
      1 CD3
      1 C*********
      1 C*********
      1 CD5
      1 CD5       NOTES
      1 CD5
      1 CD5     USES ROUTINES
      1 CD5
      1 CD5       EXEC
      1 CD5       XERTN
      1 CD5       XREQ
      1 CD5       XRMOV
      1 CD5       XRMSG
      1 CD5       XSNPT
      1 CD5       XSPCK
      1 CD5       XSPRM
      1 CD5       XTCOM
      1 CD5       XUDBG
      1 CD5
      1 C*********
```

```
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
XSEQE
```

```
47    1 BEGIN XSEQE                                                                  XSEQE
48    2    DO UNTIL A X OR 'EXIT' IS ENTERED                                         XSEQE
49    3       CALL XSPRM TO BUILD A PROMPT BASED ON PROMPT MODE FOR THE               XSEQE
50    3       NEXT TABLE ENTRY (INDICATED BY TABNDX)                                 XSEQE
51    3       CALL XTCOM TO ISSUE THE PROMPT AND RETURN RESPONSE                     XSEQE
52    3       ERNEXIT IF ERROR IN XTCOM :ERR10:                                      XSEQE
53    2    EXIT XSEDT IF X WAS ENTERED                                               XSEQE
54    3       IF NOTHING (ONLY CR) ENTERED, THEN                                     XSEQE
55    4          IF PROMPT MODE IS NOT 'ALL', THEN                                   XSEQE
56    5             CALL XRMSG - 'INVALID INPUT'                                     XSEQE
57    4          ENDIF                                                               XSEQE
58    3       ELSE                                                                   XSEQE
59    4          CALL XSNPT TO PROCESS INPUT BASED ON PROMPT MODE,                   XSEQE
60    4          CURRENT TABLE ENTRY (TABNDX), AND PROMPTED SEQUENCE                 XSEQE
61    4          NUMBER (PRMNUM)                                                     XSEQE
62    3       ENDIF                                                                  XSEQE
63    2    ENDDO                                                                     XSEQE
64    2    BUILD AWA REQUEST TO DELETE/VERIFY ABSCENCE OF NEWTAB                     XSEQE
65    2    CALL XSPCK TO PACK THE TABLE BUFFER (REMOVE DELETED ENTRIES)               XSEQE
66    2    BUILD AWA REQUEST TO ALLOCATE NEWTAB                                      XSEQE
67    2    IF NUMBER OF TABLE ENTRIES (NUMENT) > 0, THEN                             XSEQE
68    3       BUILD AWA REQUEST TO STORE NEWTAB                                      XSEQE
69    3       CALL XREQ TO PROCESS THE REQUESTS                                      XSEQE
70    3       IF THE ALLOCATE REQUEST FAILED, THEN                                   XSEQE
71    4          CALL XRMSG - 'AWA/DWA FULL, SEQUENCE TABLE NOT STORED'              XSEQE
72    4          CALL EXEC TO FREE CLASS NO. AND SAM BUFFER                          XSEQE
73    3       ELSE                                                                   XSEQE
74    4          CALL XRMSG - 'VACUOUS TABLE -- NOT STORED'                          XSEQE
75    3       ENDIF                                                                  XSEQE
76    2    ENDIF                                                                     XSEQE
77    1 EXIT XSEQE                                                                   XSEQE

78    2    :ERR10: CALL XRMSG - 'SYSTEM ERROR'                                       XSEQE
79    1 END XSEQE                                                                    XSEQE
```

```
1 CD0                                                          XSPRM
1 CD0                                                          XSPRM
1 CD0                                                          XSPRM
1 CD0        FORTRAN CALLING PROCEEDURE                        XSPRM
1 CD0                                                          XSPRM
1 CD0   CALL XSPRM                                             XSPRM
1 CD0                                                          XSPRM
1 C*********                                                   XSPRM
1 CD1                                                          XSPRM
1 CD1   XSPRM BUILDS SEQUENCE TABLE EDITOR PROMPTS.            XSPRM
1 CD1                                                          XSPRM
1 C*********                                                   XSPRM
1 CD2                                                          XSPRM
1 CD2        INPUT                                             XSPRM
1 CD2                                                          XSPRM
1 CD2   COMMON XE - LU                                         XSPRM
1 CD2                                                          XSPRM
1 CD2   COMMON XB - DEBUG, NUMENT, PRNTMD, TABNDX, WKBUF       XSPRM
1 CD2                                                          XSPRM
1 C*********                                                   XSPRM
1 CD3                                                          XSPRM
1 CD3        OUTPUT                                            XSPRM
1 CD3                                                          XSPRM
1 CD3   COMMON XB - PRMLEN, PROMPT, SEQNO, TABNDX              XSPRM
1 CD3                                                          XSPRM
1 C*********                                                   XSPRM
1 CD5                                                          XSPRM
1 CD5        NOTES                                             XSPRM
1 CD5                                                          XSPRM
1 CD5   USES ROUTINES                                          XSPRM
1 CD5                                                          XSPRM
1 CD5        XRIG                                              XSPRM
1 CD5        XRMOV                                             XSPRM
1 CD5        XRMSG                                             XSPRM
1 CD5        XRSET                                             XSPRM
1 CD5        XUDEG                                             XSPRM
1 CD5                                                          XSPRM
1 C*********                                                   XSPRM
```

```
120    1 BEGIN XSPRM                                                                                                    XSPRM
121    2    IF PROMPT MODE IS ALL, THEN                                                                                 XSPRM
122    3       IF NUMBER OF ENTRIES (NUMENT) > 0, AND                                                                   XSPRM
123    4          ENTRIES EXIST BEYOND TABLE ENTRY INDEX (TABNDX), THEN                                                 XSPRM
124    4          DO UNTIL A NON-DELETED ENTRY IS FOUND                                                                 XSPRM
125    5             INCREMENT TABLE ENTRY INDEX (TABNDX) TO NEXT ENTRY (+7)                                            XSPRM
126    4          ENDDO                                                                                                 XSPRM
127    4          BUILD PROMPT OF THE FORM ' NNNNN=PROC,INT'                                                            XSPRM
128    4          SET PROMPT SEQUENCE NUMBER (SEQNO) TO SEQUENCE NO. OF ENTRY                                           XSPRM
129    3       ELSE                                                                                                     XSPRM
130    4          SET PROMPT MODE TO CREATE                                                                             XSPRM
131    3       ENDIF                                                                                                    XSPRM
132    2    ENDIF                                                                                                       XSPRM
133    2    IF PROMPT MODE IS CREATE, THEN                                                                              XSPRM
134    3       SET TABLE ENTRY INDEX (TABNDX) TO NEXT ENTRY (NUMENT * 7 + 1)                                            XSPRM
135    3       IF NUMBER OF TABLE ENTRIES (NUMENT) > 0, THEN                                                            XSPRM
136    4          IF SEQUENCE NO. OF LAST ENTRY > 32699, THEN                                                           XSPRM
137    5             CALL XPMSG - 'UNABLE TO BUILD SEQUENCE NO. > 32700'                                                XSPRM
138    5             SET PROMPT MODE TO UPDATE                                                                          XSPRM
139    4          ELSE                                                                                                  XSPRM
140    5             SET PROMPT SEQUENCE NUMBER (SEQNO) TO NEXT MULTIPLE OF 100                                         XSPRM
141    5             BEYOND SEQUENCE NUMBER OF LAST TABLE ENTRY                                                         XSPRM
142    4          ENDIF                                                                                                 XSPRM
143    3       ELSE                                                                                                     XSPRM
144    4          SET PROMPT SEQ. NO. (SEQNO) TO BE 100                                                                 XSPRM
145    3       ENDIF                                                                                                    XSPRM
146    3       IF PROMPT MODE IS NOT UPDATE, THEN                                                                       XSPRM
147    4          BUILD PROMPT OF THE FORM ' NNNNN='                                                                    XSPRM
148    3       ENDIF                                                                                                    XSPRM
149    2    ENDIF                                                                                                       XSPRM
150    2    IF PROMPT MODE IS UPDATE, THEN                                                                              XSPRM
151    3       BUILD PROMPT OF THE FORM                                                                                 XSPRM
152    3       SET PROMPT LENGTH TO 0 CAUSING #: PROMPT TO BE ISSUED                                                    XSPRM
153    2    ENDIF                                                                                                       XSPRM
154    1 END XSPRM                                                                                                      XSPRM
```

```
156   1 CD0                                                                  XSNPT
157   1 CD0       FORTRAN CALLING PROCEEDURE                                 XSNPT
158   1 CD0                                                                  XSNPT
159   1 CD0     CALL XSNPT                                                   XSNPT
160   1 CD0                                                                  XSNPT
161   1 C**********                                                          XSNPT
162   1 CD1                                                                  XSNPT
163   1 CD1     XSNPT PROCESSES THE INPUT RESPONSES OF THE SEQUENCE          XSNPT
164   1 CD1     TABLE EDITOR                                                 XSNPT
165   1 CD1                                                                  XSNPT
166   1 C**********                                                          XSNPT
167   1 CD2                                                                  XSNPT
168   1 CD2       INPUT                                                      XSNPT
169   1 CD2                                                                  XSNPT
170   1 CD2     COMMON XE - COMBUF, COMPTR, LU, TOKENS                       XSNPT
171   1 CD2                                                                  XSNPT
172   1 CD2     COMMON XB - DEBUG,  DIRECT, NUMDIR, NUMENT, PRMTMD           XSNPT
173   1 CD2                 SEQNO,  TABNDX, WKBUF                            XSNPT
174   1 CD2                                                                  XSNPT
175   1 C**********                                                          XSNPT
176   1 CD3                                                                  XSNPT
177   1 CD3       OUTPUT                                                     XSNPT
178   1 CD3                                                                  XSNPT
179   1 CD3     COMMON XE - COMPTR                                           XSNPT
180   1 CD3                                                                  XSNPT
181   1 CD3     COMMON XB - INSERT, IRETC,  NUMENT, PRMTMD,  SEQNO,          XSNPT
182   1 CD3                 TABNDX, TABSIZ, WKBUF                            XSNPT
183   1 CD3                                                                  XSNPT
184   1 C**********                                                          XSNPT
185   1 CD5                                                                  XSNPT
186   1 CD5       NOTES                                                      XSNPT
187   1 CD5                                                                  XSNPT
188   1 CD5     USES ROUTINES                                                XSNPT
189   1 CD5                                                                  XSNPT
190   1 CD5       XRMSG                                                      XSNPT
191   1 CD5       XSDEL                                                      XSNPT
192   1 CD5       XSENT                                                      XSNPT
193   1 CD5       XSLIS                                                      XSNPT
194   1 CD5       XSNUM                                                      XSNPT
195   1 CD5       XSPMT                                                      XSNPT
196   1 CD5       XUDBG                                                      XSNPT
197   1 CD5                                                                  XSNPT
198   1 C**********                                                          XSNPT
```

```
200   1 BEGIN XSNPT                                                                                                    XSNPT
201   2    IF PROMPT MODE IS UPDATE, THEN                                                                              XSNPT
202   3       IF TOKEN INPUT IS AN INTEGER, THEN                                                                       XSNPT
203   4          ERREXIT IF INTEGER < 1 :ERR06:                                                                        XSNPT
204   4          RETAIN INTEGER AS SEQUENCE NO. (SEQNO)                                                                XSNPT
205   4          INCREMENT TO NEXT TOKEN                                                                               XSNPT
206   4          ERREXIT IF TOKEN IS NOT '=' :ERR01:                                                                   XSNPT
207   4          INCREMENT TO NEXT TOKEN                                                                               XSNPT
208   4          START SEARCH UNTIL NUMBER OF TABLE ENTRIES (NUMENT) SEARCHED                                          XSNPT
209   4          EXITIF SEQUENCE NO. OF ENTRY = SEQUENCE NO. INPUT (SEQNO.)                                            XSNPT
210   5             SET TABLE ENTRY INDEX (TABNDX) TO THIS ENTRY                                                       XSNPT
211   5             SET INSERT FLAG TO ZERO INDICATING REPLACEMENT OF ENTRY                                            XSNPT
212   4          EXITIF SEQUENCE NO. OF ENTRY > SEQUENCE NO. INPUT (SEQNO)                                             XSNPT
213   5             SET TABLE ENTRY INDEX (TABNDX) TO THIS ENTRY                                                       XSNPT
214   5             SET INSERT FLAG TO 1 INDICATING INSERT NECESSARY                                                   XSNPT
215   4          ORELSE                                                                                                XSNPT
216   5             INCREMENT TO NEXT TABLE ENTRY                                                                      XSNPT
217   4          ENDLOOP                                                                                               XSNPT
218   5             SET TABLE ENTRY INDEX (TABNDX) TO NEXT ENTRY OF TABLE                                              XSNPT
219   5             SET INSERT FLAG TO 2 INDICATING EXTENSION TO END OF TABLE                                          XSNPT
220   4          ENDSEARCH                                                                                             XSNPT
221   4          CALL XSENT TO REPLACE/INSERT/ADD TABLE ENTRY BASED ON INSERT FLAG                                     XSNPT
222   3       ELSE                                                                                                     XSNPT
223   4          ERREXIT IF TOKEN IS NOT A NAME :ERR01:                                                                XSNPT
224   4          START SEARCH UNTIL LIST OF SEQ. EDIT. DIRECTIVES SEARCHED                                             XSNPT
225   4          EXITIF NAME INPUT IS DIRECTIVE                                                                        XSNPT
226   5             SET INDEX TO DIRECTIVE LIST ENTRY                                                                  XSNPT
227   4          ORELSE                                                                                                XSNPT
228   5             INCREMENT TO NEXT DIRECTIVE                                                                        XSNPT
229   5             ENLOOP                                                                                             XSNPT
230   5             ERREXIT :ERR02:                                                                                    XSNPT
231   4          ENDSEARCH                                                                                             XSNPT
232   4          CASE (XSLIS, XSDEL, XSPMT, XSNUM, XSMRG), INDEX                                                       XSNPT
233   4          ENDIF                                                                                                 XSNPT
234   4          ELSE, PROMPT MODE IS NOT UPDATE                                                                       XSNPT
235   4          IF TOKEN IS #, THEN                                                                                   XSNPT
236   5             SET PROMPT MODE TO UPDATE                                                                          XSNPT
237   4          ELSE                                                                                                  XSNPT
238   5             IF AN & IS INPUT, THEN                                                                             XSNPT
239   6                ERREXIT IF PROMPT MODE IS NOT UPDATE :ERR01:                                                    XSNPT
240   6                MARK THIS TABLE ENTRY AS DELETED                                                               XSNPT
241   6                DO FROM END OF TABLE UNTIL A NONDELETED ENTRY IS FOUND                                         XSNPT
242   7                   IF TABLE ENTRY IS MARKED FOR DELETION, THEN                                                 XSNPT
243   8                      DECREMENT NUMBER OF TABLE ENTRIES BY ONE                                                 XSNPT
244   7                   ENDIF                                                                                       XSNPT
245   6                ENDDO                                                                                          XSNPT
246   5             ELSE                                                                                               XSNPT
247   6                IF PROMPT MODE IS ALL, THEN                                                                    XSNPT
248   7                   SET INSERT FLAG TO TWO  TO INDICATE EXTENSION OF TABLE                                      XSNPT
249   6                ELSE                                                                                           XSNPT
250   7                   SET INSERT FLAG TO ZERO TO INDICATE REPLACEMENT OF TABLE ENTRY                              XSNPT
251   6                ENDIF                                                                                          XSNPT
252   6                CALL XSENT TO BUILD ENTRY BASE ON INSERT FLAG                                                  XSNPT
253   5             ENDIF                                                                                              XSNPT
254   4          ENDIF                                                                                                 XSNPT
255   3       ENDIF                                                                                                    XSNPT
256   2    EXIT XSNPT                                                                                                  XSNPT

257   3       :ERR01: CALL XRMSG - 'SYNTAX ERROR - MISSING OR EXTRANEOUS FIELD'                                        XSNPT
```

```
258   3       :ERR02: CALL XRMSG - 'INVALID DIRECTIVE'                    XSNPT
259   3       :ERR06: CALL XRMSG - 'INVALID SEQUENCE NUMBER ENTERED'      XSNPT
260   2     END XSNPT                                                     XSNPT
```

```
263   2   CD0                                                                    XSENT
264   2   CD0       FORTRAN CALLING PROCEEDURE                                   XSENT
265   2   CD0                                                                    XSENT
266   2   CD0    CALL XSENT                                                      XSENT
267   2   CD0                                                                    XSENT
268   2   C*********                                                             XSENT
269   2   CD1                                                                    XSENT
270   2   CD1                                                                    XSENT
271   2   C*********                                                             XSENT
272   2   CD1    XSENT CONSTRUCTS SEQUENCE TABLE ENTRIES                         XSENT
273   2   CD2                                                                    XSENT
274   2   CD2    COMMON XE - COMBUF, COMPTR, TOKENS, LU                          XSENT
275   2   CD2                                                                    XSENT
276   2   CD2    COMMON XB - DEBUG,  LIBDSZ, NUMENT, PRMTMD,                     XSENT
277   2   CD2                SEQNO,  TABNDX, WKBUF                               XSENT
278   2   CD2                                                                    XSENT
279   2   C*********                                                             XSENT
280   2   CD3                                                                    XSENT
281   2   CD3       OUTPUT                                                       XSENT
282   2   CD3                                                                    XSENT
283   2   CD3    COMMON XE - COMPTR                                              XSENT
284   2   CD3                                                                    XSENT
285   2   CD3    COMMON XB - INTNAM, NUMENT, PRCNAM, PRMTMD,                     XSENT
286   2   CD3                SEQNO,  TABNDX, WKBUF                               XSENT
287   2   CD3                                                                    XSENT
288   2   C*********                                                             XSENT
289   2   CD5                                                                    XSENT
290   2   CD5       NOTES                                                        XSENT
291   2   CD5                                                                    XSENT
292   2   CD5    USES ROUTINES                                                   XSENT
293   2   CD5                                                                    XSENT
294   2   CD5       XRCPR                                                        XSENT
295   2   CD5       XREXT                                                        XSENT
296   2   CD5       XRMOV                                                        XSENT
297   2   CD5       XRMSG                                                        XSENT
298   2   CD5       XRSET                                                        XSENT
299   2   CD5       XRPCK                                                        XSENT
300   2   CD5       XUDMG                                                        XSENT
301   2   CD5                                                                    XSENT
302   2   C*********                                                             XSENT
303   2   BEGIN XSENT                                                            XSENT
304   3      ERREXIT IF TOKEN INPUT IS NOT A NAME :ERR01:                        XSENT
305   3      RETAIN THIS NAME AS PROC. NAME                                      XSENT
306   3      INCREMENT TO NEXT TOKEN                                             XSENT
307   3      START SEARCH UNTIL ALL ENTRIES OF    XLIBD SEARCHED                 XSENT
308   3      EXITIF   XLIBD ENTRY = PROC. NAME                                   XSENT
309   3      ORELSE                                                              XSENT
310   4         INCREMENT TO NEXT ENTRY                                          XSENT
311   3      ENDLOOP                                                             XSENT
312   4         ERREXIT :ERR03:                                                  XSENT
313   3      ENDSEARCH                                                           XSENT
314   3      IF COMMA IS NEXT TOKEN, THEN                                        XSENT
315   4         ERREXIT IF INT. TABLE NOT REQUIRED FOR THIS PROCESSOR :ERR04:    XSENT
316   4         INCREMENT TO NEXT TOKEN                                          XSENT
317   4         ERREXIT IF NEXT TOKEN IS NOT A NAME :ERR01:                      XSENT
318   4         RETAIN THIS NAME AS INTERFACE TABLE NAME                         XSENT
319   4         INCREMENT TO NEXT TOKEN                                          XSENT
320   3      ELSE                                                                XSENT
321   4         SET INTERFACE TABLE NAME TO ZERO                                 XSENT
```

```
322   3       ENDIF                                                                                                    XSENT
323   3       ERREXIT IF NEXT TOKEN IS NOT EOS :ERR01:                                                                 XSENT
324   3       IF INSERT FLAG DOES NOT INDICATE REPLACE, THEN (I.E. INSERT OR ADD)                                      XSENT
325   4          IF NUMBER OF TABLE ENTRIES (NUMENT) = 150, THEN                                                       XSENT
326   5             CALL XSPCK TO PACK TABLE BUFFER (REMOVE DELETED ENTRIES)                                           XSENT
327   5             IF NUMBER OF TABLE ENTRIES STILL = 150, THEN                                                       XSENT
328   6                SET PROMPT MODE TO UPDATE                                                                       XSENT
329   6                ERREXIT :ERR05:                                                                                 XSENT
330   5             ENDIF                                                                                              XSENT
331   4          ENDIF                                                                                                 XSENT
332   4          IF INSERT FLAG INDICATES INSERT (=1), THEN                                                            XSENT
333   5             IF ENTRY ABOVE INDEXED ENTRY (TABNDX) IS MARKED DELETED, THEN                                      XSENT
334   6                SET TABLE ENTRY INDEX (TABNDX) TO BE THIS DELETED ENTRY                                         XSENT
335   6                SET INSERT FLAG TO 0 INDICATING ENTRY REPLACEMENT                                               XSENT
336   5             ELSE                                                                                               XSENT
337   6                SET MOVLEN = MIN (5, 150-NUMENT) * 7                                                            XSENT
338   6                DO FOR ALL TABLE ENTRIES FROM BOTTOM OF TABLE TO TABNDX                                         XSENT
339   7                   MOVE THE ENTRY DOWN MOVLEN WORDS                                                             XSENT
340   6                ENDDO                                                                                           XSENT
341   6                IF MOVLEN > 7 (I.E. MORE THAN 1 ENTRY), THEN                                                    XSENT
342   7                   MARK FOLLOWING ENTRIES AS DELETED                                                            XSENT
343   6                ENDIF                                                                                           XSENT
344   5             ENDIF                                                                                              XSENT
345   4          ENDIF                                                                                                 XSENT
346   4          SET SEQUENCE NO. FIELD OF ENTRY TO SEQUENCE NO. (SEQNO) INPUT/PROMPTED                                XSENT
347   3       ENDIF                                                                                                    XSENT
348   3       MOVE PROC. NAME AND INT. NAME INTO TABLE ENTRY AT TABNDX                                                  XSENT
349   3       IF INSERT FLAG DOES NOT INDICATE ENTRY REPLACEMENT, THEN                                                 XSENT
350   4          INCREMENT NUMBER OF ENTRIES (NUMENT) BY 1                                                             XSENT
351   3       ENDIF                                                                                                    XSENT
352   2    EXIT XSENT                                                                                                  XSENT

353   3       :ERR01: CALL XRMSG - 'SYNTAX ERROR - MISSING OR EXTRANEOUS FIELD'                                        XSENT

354   3       :ERR03: CALL XRMSG - '.... IS NOT A VALID PROCESSOR NAME'                                                XSENT

355   3       :ERR04: CALL XRMSG - '... DOES NOT USE AN INTERFACE TABLE'                                               XSENT

356   3       :ERR05: CALL XRMSG - 'MAX. SIZE OF 150 SEQUENCE ENTRIES ALREADY REACHED'                                 XSENT
357   2    END XSENT                                                                                                   XSENT
```

```
C     2    CD0
C     2    CD0       FORTRAN CALLING PROCEEDURE
C     2    CD0
C     2    CD0    CALL XSPCK
C     2    CD0
C     2    C*********
C     2    CD1
C     2    CD1    XSPCK COMPACTS THE WORKING BUFFER BY REMOVING ALL SEQUENCE
C     2    CD1    TABLE ENTRIES MARKED FOR DELETION
C     2    CD1
C     2    C*********
C     2    CD2
C     2    CD2       INPUT
C     2    CD2
C     2    CD2    COMMON XE - LU
C     2    CD2
C     2    CD2    COMMON XB - DEBUG,  NUMENT, TABNDX, WKBUF
C     2    CD2
C     2    C*********
C     2    CD3
C     2    CD3       OUTPUT
C     2    CD3
C     2    CD3    COMMON XB - NUMENT, TABNDX, WKBUF
C     2    CD3
C     2    C*********
C     2    CD5
C     2    CD5       NOTES
C     2    CD5
C     2    CD5    USES ROUTINES
C     2    CD5
C     2    CD5       XRMOV
C     2    CD5       XRMSG
C     2    CD5       XUDBG
C     2    CD5
C     2    C*********
```

XSPCK

```
395   2   BEGIN XSPCK                                                                                    XSPCK
396   3      IF THE TABLE IS NOT EMPTY, THEN                                                             XSPCK
397   4         DO UNTIL NUMBER OF ENTRIES (NUMENT) PROCESSED                                            XSPCK
398   5            IF THIS ENTRY IS MARKED DELETED, THEN                                                 XSPCK
399   6               SET MOVE LENGTH (MOVLEN) TO 7                                                      XSPCK
400   6               DO UNTIL A NON-DELETED ENTRY IS FOUND                                              XSPCK
401   7                  INCREMENT MOVLEN BY 7                                                           XSPCK
402   6               ENDDO                                                                              XSPCK
403   6               MOVE MOVLEN WORDS BEGINNING WITH THE NON-DELETED ENTRY TO                          XSPCK
404   6               THE DELETED ENTRY                                                                  XSPCK
405   6               DECREMENT NUMENT BY MOVLEN/7                                                       XSPCK
406   6               IF TABLE INDEX (TABNDX) > INDEX TO DELETED ENTRY, THEN                             XSPCK
407   7                  DECREMENT TABLE INDEX (TABNDX) BY MOVLEN                                        XSPCK
408   6               ENDIF                                                                              XSPCK
409   5            ENDIF                                                                                 XSPCK
410   4         ENDDO                                                                                    XSPCK
411   3      ENDIF                                                                                       XSPCK
412   2   END XSPCK                                                                                      XSPCK
```

```
414  2  CD0                                                                    XSLST
415  2  CD0       FORTRAN CALLING PROCEEDURE                                   XSLST
416  2  CD0                                                                    XSLST
417  2  CD0    CALL XSLST                                                      XSLST
418  2  CD0                                                                    XSLST
419  2  C*********                                                             XSLST
420  2  CD1                                                                    XSLST
421  2  CD1    XSLST WILL LIST TO A SPECIFIED DEVICE THE SEQUENCE TABLE        XSLST
422  2  CD1    CONTAINED IN THE WORKING BUFFER                                 XSLST
423  2  CD1                                                                    XSLST
424  2  C*********                                                             XSLST
425  2  CD2                                                                    XSLST
426  2  CD2       INPUT                                                        XSLST
427  2  CD2                                                                    XSLST
428  2  CD2    COMMON XE - LU, REQBUF, SUBSTA                                  XSLST
429  2  CD2                                                                    XSLST
430  2  CD2    COMMON XB - BEGNO,  DEBUG,  ENDNO,  LISTLU,                     XSLST
431  2  CD2                NEWTAB, NUMENT, WKBUF                             XSLST
432  2  CD2                                                                    XSLST
433  2  C*********                                                             XSLST
434  2  CD5                                                                    XSLST
435  2  CD5       NOTES                                                        XSLST
436  2  CD5                                                                    XSLST
437  2  CD5    USES ROUTINES                                                   XSLST
438  2  CD5                                                                    XSLST
439  2  CD5       XRIG                                                         XSLST
440  2  CD5       XRMOV                                                        XSLST
441  2  CD5       XRSET                                                        XSLST
442  2  CD5       XUDBG                                                        XSLST
443  2  CD5                                                                    XSLST
444  2  C*********                                                             XSLST
445  2  BEGIN XSLST                                                            XSLST
446  3     IF SUBSTATE FLAG INDICATES THAT SEQ. EDITOR NOT MAKING THIS CALL, THEN XSLST
447  4        DETERMINE SIZE OF SEQ. TAB FROM AWA REQUEST BUFFER ENTRY         XSLST
448  4        SET LIMITS (BEGNO AND ENDNO) OF SEQ. ENTRIES LISTED              XSLST
449  4        SET TABLE NAME (NEWTAB) FROM AWA REQUEST BUFFER ENTRY            XSLST
450  3     ENDIF                                                             XSLST
451  3     WRITE HEADER LINE - 'SEQUENCE TABLE XXXXXX'                       XSLST
452  3     IF SEQ. TABLE ENTRY IS NOT MARKED AS DELETED, THEN                XSLST
453  4        DO FROM BEGNO TO ENDNO                                         XSLST
454  5           MOVE PROC. NAME AND INT. NAME FROM ENTRY TO PRINT BUFFER     XSLST
455  5           IF INT. TABLE NAME = 0, THEN                                XSLST
456  6              SET LENGTH OF PRINT TO BE  7 WORDS (14 CHARS.)           XSLST
457  5           ELSE                                                        XSLST
458  6              SET LENGTH OF PRINT LINE TO BE 10 WORDS (20 CHARS.)      XSLST
459  5           ENDIF                                                       XSLST
460  5           CALL XRI6 TO CONVERT SEQ. NO. OF TABLE ENTRY AND PLACE IN BUFFER XSLST
461  5           WRITE PRINT BUFFER                                          XSLST
462  4        ENDIF                                                          XSLST
463  3     ENDDO                                                             XSLST
464  2  END XSLST                                                           XSLST
```

```
466   2   CD0
467   2   CD0       FORTRAN CALLING PROCEEDURE                                    XSCAN
468   2   CD0                                                                     XSCAN
469   2   CD0     CALL XSCAN                                                      XSCAN
470   2   CD0                                                                     XSCAN
471   2   C*********                                                             XSCAN
472   2   CD1                                                                     XSCAN
473   2   CD1     XSCAN PERFORMS SYNTACTICAL PROCESSING FOR THE LIST AND          XSCAN
474   2   CD1     DELETE DIRECTIVES OF THE SEQUENCE TABLE EDITOR                  XSCAN
475   2   CD1                                                                     XSCAN
476   2   C*********                                                             XSCAN
477   2   CD2                                                                     XSCAN
478   2   CD2       INPUT                                                         XSCAN
479   2   CD2                                                                     XSCAN
480   2   CD2     COMMON XE - COMBUF, COMPTR, LU, TOKENS                          XSCAN
481   2   CD2                                                                     XSCAN
482   2   CD2     COMMON XB - DEBUG,  NUMENT, TABSIZ, WKBUF                       XSCAN
483   2   CD2                                                                     XSCAN
484   2   C*********                                                             XSCAN
485   2   CD3                                                                     XSCAN
486   2   CD3       OUTPUT                                                        XSCAN
487   2   CD3                                                                     XSCAN
488   2   CD3     COMMON XB - BEGNO,  ENDNO,  IRETC                               XSCAN
489   2   CD3                                                                     XSCAN
490   2   C*********                                                             XSCAN
491   2   CD5                                                                     XSCAN
492   2   CD5       NOTES                                                         XSCAN
493   2   CD5                                                                     XSCAN
494   2   CD5     USES ROUTINES                                                   XSCAN
495   2   CD5                                                                     XSCAN
496   2   CD5       XRMSG                                                         XSCAN
497   2   CD5       XUDBG                                                         XSCAN
498   2   CD5                                                                     XSCAN
499   2   C*********                                                             XSCAN
```

```
501   2   BEGIN XSCAN                                                                    XSCAN
502   3      SET LIST LIMITS (BEGNO AND ENDNO) TO ZERO                                   XSCAN
503   3      IF NEXT TOKEN IS A COMMA, THEN                                              XSCAN
504   4         INCREMENT TO NEXT TOKEN                                                  XSCAN
505   4         IF NEXT TOKEN IS AN INTEGER, THEN                                        XSCAN
506   5            ERREXIT IF VALUE IS < 1 :ERR06:                                       XSCAN
507   5            SET BEGIN LIMIT (BEGNO) TO THIS VALUE                                 XSCAN
508   5            INCREMENT TO NEXT TOKEN                                               XSCAN
509   4         ENDIF                                                                    XSCAN
510   4         IF NEXT TOKEN IS A COMMA, THEN                                           XSCAN
511   5            INCREMENT TO NEXT TOKEN                                               XSCAN
512   5            IF NEXT TOKEN IS AN INTEGER, THEN                                     XSCAN
513   6               ERREXIT IF VALUE IS < 1 :ERR06:                                    XSCAN
514   6               SET END LIMIT (ENDNO) TO THIS VALUE                                XSCAN
515   6               INCREMENT TO NEXT TOKEN                                            XSCAN
516   5            ENDIF                                                                 XSCAN
517   4         ENDIF                                                                    XSCAN
518   3      ENDIF                                                                       XSCAN
519   3      ERREXIT IF NEXT TOKEN IS NOT EOS :ERR01:                                    XSCAN
520   3      IF BEGIN LIMIT (BEGNO) = 0, THEN                                            XSCAN
521   4         SET BEGIN LIMIT (BEGNO) TO 1 (BEGNO IS NOW A TABLE INDEX)                XSCAN
522   3      ELSE                                                                        XSCAN
523   4         START SEARCH FROM FIRST TO LAST SEQ. TABLE ENTRY                         XSCAN
524   4         EXITIF SEQ. NO. OF THIS ENTRY = BEGIN LIMIT (BEGNO)                      XSCAN
525   5            SET BEGIN LIMIT (BEGNO) TO INDEX OF THIS ENTRY                        XSCAN
526   4         ORELSE                                                                   XSCAN
527   5            INCREMENT INDEX TO NEXT TABLE ENTRY                                   XSCAN
528   4         ENDLOOP                                                                  XSCAN
529   5            ERREXIT :ERR06:                                                       XSCAN
530   4         END SEARCH                                                               XSCAN
531   3      ENDIF                                                                       XSCAN
532   3      IF END LIMIT (ENDNO) = 0, THEN                                              XSCAN
533   4         SET END LIMIT (ENDDO) TO INDEX OF LAST TABLE ENTRY                       XSCAN
534   3      ELSE                                                                        XSCAN
535   4         START SEARCH FROM BEGIN LIMIT (BEGNO) TO LAST TABLE ENTRY                XSCAN
536   4         EXITIF SEQ. NO. OF THIS ENTRY = END LIMIT (ENDNO)                        XSCAN
537   5            SET END LIMIT (ENDNO) TO INDEX OF THIS ENTRY                          XSCAN
538   4         ORELSE                                                                   XSCAN
539   5            INCREMENT INDEX TO NEXT TABLE ENTRY                                   XSCAN
540   4         ENDLOOP                                                                  XSCAN
541   5            ERREXIT :ERR08:                                                       XSCAN
542   4         END SEARCH                                                               XSCAN
543   3      ENDIF                                                                       XSCAN
544   3      SET RETURN CODE TO INDICATE NO ERROR                                        XSCAN
545   3      SET RETURN CODE TO INDICATE AN ERROR                                        XSCAN
546   2   EXIT XSLIS                                                                     XSCAN

547   3      :ERR01: CALL XRMSG - 'SYNTAX ERROR - MISSING OR EXTRANEOUS FIELD'           XSCAN

548   3      :ERR06: CALL XRMSG - 'INVALID SEQUENCE NUMBER'                              XSCAN

549   3      :ERR08: CALL XRMSG - 'INVALID SEQUENCE NUMBER RANGE'                        XSCAN
550   2   END XSCAN                                                                      XSCAN
```

```
552   2   CD0                                                                    XSPMT
553   2   CD0       FORTRAN CALLING PROCEEDURE                                   XSPMT
554   2   CD0                                                                    XSPMT
555   2   CD0   CALL XSPMT                                                       XSPMT
556   2   CD0                                                                    XSPMT
557   2   C*********                                                             XSPMT
558   2   CD1                                                                    XSPMT
559   2   CD1   XSPMT PROCESSES THE SEQUENCE TABLE EDITOR PROMPT DIRECTIVE       XSPMT
560   2   CD1                                                                    XSPMT
561   2   C*********                                                             XSPMT
562   2   CD2                                                                    XSPMT
563   2   CD2       INPUT                                                        XSPMT
564   2   CD2                                                                    XSPMT
565   2   CD2   COMMON XE - COMBUF, COMPTR, LU, TOKENS                           XSPMT
566   2   CD2                                                                    XSPMT
567   2   CD2   COMMON XB - DEBUG                                                XSPMT
568   2   CD2                                                                    XSPMT
569   2   C*********                                                             XSPMT
570   2   CD3                                                                    XSPMT
571   2   CD3       OUTPUT                                                       XSPMT
572   2   CD3                                                                    XSPMT
573   2   CD3   COMMON XB - PRMTMD, TABNDX                                       XSPMT
574   2   CD3                                                                    XSPMT
575   2   C*********                                                             XSPMT
576   2   CD5                                                                    XSPMT
577   2   CD5       NOTES                                                        XSPMT
578   2   CD5                                                                    XSPMT
579   2   CD5   USES ROUTINES                                                    XSPMT
580   2   CD5                                                                    XSPMT
581   2   CD5       XRMSG                                                        XSPMT
582   2   CD5       XUDBG                                                        XSPMT
583   2   C*********                                                             XSPMT
584   2   BEGIN XSPMT                                                            XSPMT
585   3      ERREXIT IF TOKEN IS NOT COMMA :ERR01:                               XSPMT
586   3      INCREMENT TO NEXT TOKEN                                             XSPMT
587   3      ERREXIT IF TOKEN IS NOT A NAME :ERR01:                              XSPMT
588   3      INCREMENT TO NEXT TOKEN                                             XSPMT
589   3      ERREXIT IF TOKEN IS NOT EOS :ERR01:                                 XSPMT
590   3      IF NAME IS 'N', THEN                                                XSPMT
591   4         SET PROMPT MODE TO CREATE                                        XSPMT
592   3      ELSE                                                                XSPMT
593   4         ERREXIT IF NAME IS NOT 'A' :ERR09:                               XSPMT
594   4         SET PROMPT MODE TO ALL                                           XSPMT
595   4         SET TABLE ENTRY INDEX (TABNDX) TO 0                              XSPMT
596   3      ENDIF                                                               XSPMT
597   2   EXIT XSPMT                                                             XSPMT

598   3      :ERR01: CALL XRMSG - 'SYNTAX ERROR - MISSING OR EXTRANEOUS FIELD'   XSPMT

599   3      :ERR09: CALL XRMSG - 'SYNTAX ERROR - INVALID QUALIFIER              XSPMT
600   2   END XSPMT                                                              XSPMT
```

```
602  2  CD0
603  2  CD0       FORTRAN CALLING PROCEEDURE
604  2  CD0
605  2  CD0    CALL XSDEL
606  2  CD0
607  2  C*********
608  2  CD1
609  2  CD1    XSDEL IS THE SEQUENCE TABLE EDITOR DELETE DIRECTIVE PROCESSOR
610  2  CD1
611  2  C*********
612  2  CD2
613  2  CD2       INPUT
614  2  CD2
615  2  CD2    COMMON XE - LU
616  2  CD2
617  2  CD2    COMMON XB - BEGNO,  DEBUG,  ENDNO,  IRETC, NUMENT
618  2  CD2
619  2  C*********
620  2  CD3
621  2  CD3       OUTPUT
622  2  CD3
623  2  CD3    COMMON XB - NUMENT, WKBUF
624  2  CD3
625  2  C*********
626  2  CD5
627  2  CD5       NOTES
628  2  CD5
629  2  CD5    USES ROUTINES
630  2  CD5
631  2  CD5       XSCAN
632  2  CD5       XUDBG
633  2  CD5
634  2  C*********
635  2  BEGIN XSDEL
636  3     CALL XSCAN TO SCAN AND INTERPRET SEQ. LIMITS ON THE DIRECTIVE
637  3     IF NO ERROR INDICATED, THEN
638  4        DO FROM THE BEGIN LIMIT (BEGNO) TO THE END LIMIT (ENDNO)
639  5           MARK THIS SEQ. TABLE ENTRY AS DELETED
640  4        ENDDO
641  4        DO FROM LAST TABLE ENTRY TO FIRST ENTRY, OR
642  5           UNTIL A NON-DELETED ENTRY FOUND
643  5           IF ENTRY IS MARKED DELETED, THEN
644  6              DECREMENT NUMBER OF TABLE ENTRIES (NUMENT) BY ONE
645  5           ENDIF
646  4        ENDDO
647  3     ENDIF
648  2  END XSDEL
```

XSDEL

```
650   2   BEGIN XSLIS                                                    XSLIS
651   3      CALL XSCAN TO SCAN AND INTERPRET SEQ. LIMITS ON THE DIRECTIVE  XSLIS
652   3      IF NO ERROR INDICATED, THEN                                 XSLIS
653   4         CALL XSLST TO LIST TABLE ENTRIES DEFINED BY LIMITS       XSLIS
654   3      ENDIF                                                       XSLIS
655   2   END XSLIS                                                      XSLIS
```


C-4

```
657   2   CD0
658   2   CD0       FORTRAN CALLING PROCEEDURE
659   2   CD0
660   2   CD0    CALL XSLIS
661   2   CD0
662   2   C**********
663   2   CD1
664   2   CD1    XSLIS IS THE SEQUENCE TABLE EDITOR LIST DIRECTIVE PROCESSOR
665   2   CD1
666   2   C**********
667   2   CD2
668   2   CD2       INPUT
669   2   CD2
670   2   CD2    COMMON XE - LU
671   2   CD2
672   2   CD2    COMMON XB - DEBUG, PRETC
673   2   CD2
674   2   C**********
675   2   CD5
676   2   CD5       NOTES
677   2   CD5
678   2   CD5    USES ROUTINES
679   2   CD5
680   2   CD5       XSCAN
681   2   CD5       XSLST
682   2   CD5       XUDBG
683   2   CD5
684   2   C**********
685   2   CD0
686   2   CD0       FORTRAN CALLING PROCEEDURE
687   2   CD0
688   2   CD0    CALL XSNUM
689   2   CD0
690   2   C**********
691   2   CD1
692   2   CD1    XSNUM IS THE SEQUENCE TABLE EDITOR NUMBER DIRECTIVE PROCESSOR
693   2   CD1
694   2   C**********
695   2   CD2
696   2   CD2       INPUT
697   2   CD2
698   2   CD2    COMMON XE - COMBUF, COMPTR, LU, TOKENS
699   2   CD2
700   2   CD2    COMMON XB - DEBUG, NUMENT
701   2   CD2
702   2   C**********
703   2   CD3
704   2   CD3       OUTPUT
705   2   CD3
706   2   CD3    COMMON XB - WKBUF
707   2   CD3
708   2   C**********
709   2   CD5
710   2   CD5       NOTES
711   2   CD5
712   2   CD5    USES ROUTINES
713   2   CD5
714   2   CD5       XRMSG
715   2   CD5       XUDBG
```

```
716   2   CD5                                                                                    XSNUM
717   2   C*********                                                                             XSNUM
718   2   BEGIN XSNUM                                                                            XSNUM
719   3      ERREXIT IF TOKEN IS NOT EOS :ERR01:                                                 XSNUM
720   3      IF NUMBER OF TABLE ENTRIES (NUMENT) > 0, THEN                                       XSNUM
721   4         SET SEQUENCE NUMBER (SEQNO) TO 100                                               XSNUM
722   4         DO FOR ALL ENTRIES IN TABLE                                                      XSNUM
723   5            IF TABLE ENTRY IS NOT MARKED DELETED, THEN                                    XSNUM
724   6               SET SEQ. NO. FIELD OF ENTRY TO SEQUENCE NUMBER (SEQNO)                     XSNUM
725   6               INCREMENT SEQUENCE NUMBER (SEQNO) BY 100                                   XSNUM
726   5            ENDIF                                                                         XSNUM
727   4         ENDDO                                                                            XSNUM
728   3      ENDIF                                                                               XSNUM
729   2   EXIT XSNUM                                                                             XSNUM

730   3      :ERR01: CALL XRMSG - 'SYNTAX ERROR - MISSING OR EXTRANEOUS FIELD'                   XSNUM
731   2   END XSNUM                                                                              XSNUM
```


C-4

```
SYMBOL DEFINITION TABLE
::ERR01 :: 730
::ERR01 :: 598
::ERR01 :: 547
::ERR01 :: 353
::ERR01 :: 257
::ERR02 :: 258
::ERR03 :: 354
::ERR04 :: 355
::ERR05 :: 356
::ERR06 :: 548
::ERR06 :: 259
::ERR08 :: 549
::ERR09 :: 599
::ERR10 ::  78
   XSCAN    501
   XSDEL    635
   XSENT    303
   XSEQE     47
   XSLIS    650
   XSLST    445
   XSNPT    200
   XSNUM    718
   XSPCK    395
   XSPMT    584
   XSPRM    120

@XQT F.PDLIST
```

```
      1 CD*************                                                      XTCOM
      1 CDO                                                                 XTCOM
      1 CDO      FORTRAN CALLING PROCEDURE FOR TERMINAL COMMUNICATIONS:     XTCOM
      1 CDO                                                                 XTCOM
      1 CDO           CALL XTCOM (PROMPT, PRMLEN, RETCOD)                   XTCOM
      1 CDO                                                                 XTCOM
      1 CD*************                                                      XTCOM
      1 CD1                                                                 XTCOM
      1 CD1      WRITE 'PRMLEN' WORDS OF CHARACTERS 'PROMPT' TO THE USER,   XTCOM
      1 CD1      READ THE USER'S RESPONSE, CALL XTLAN TO ANALYZE THE RESPONSE,XTCOM
      1 CD1      AND RETURN A 'RETCOD' TO THE USER INDICATING THE SUCCESS OF XTCOM
      1 CD1      XTCOM AND THE USEFULNESS OF THE RESPONSE.                  XTCOM
      1 CD1                                                                 XTCOM
      1 CD*************                                                      XTCOM
      1 CD2                                                                 XTCOM
      1 CD2      INPUTS FROM CALLING SEQUENCE:                              XTCOM
      1 CD2                                                                 XTCOM
      1 CD2           PROMPT - (INTEGER, PRMLEN WORDS)  AN ARRAY OF PRMLEN WORDS XTCOM
      1 CD2                    USED AS A PROMPT TO THE USER                 XTCOM
      1 CD2           PRMLEN - (INTEGER, 1 WORD) THE LENGTH IN WORDS OF     XTCOM
      1 CD2                    THE PROMPT ARRAY. MAXIMUM LENGTH IS 38 WORDS.XTCOM
      1 CD2                                                                 XTCOM
      1 CD*************                                                      XTCOM
      1 CD3                                                                 XTCOM
      1 CD3      OUTPUTS IN CALLING SEQUENCE:                               XTCOM
      1 CD3                                                                 XTCOM
      1 CD3           RETCOD - (INTEGER, 1 WORD` IS A COMPLETION CODE DEFINED XTCOM
      1 CD3                    AS FOLLOWS:                                  XTCOM
      1 CD3                 0 - NORMAL RETURN. BUFFER CONTAINS RESPONSE     XTCOM
      1 CD3                 1 - USER RESPONDED %. BUFFER CONTAINS RESPONSE UP XTCOM
      1 CD3                     TO AND INCLUDING %.                         XTCOM
      1 CD3                 2 - USER ENTERED A CR. THERE IS NO RESPONSE     XTCOM
      1 CD3                 3 - PROMPT WAS TOO LONG. MAXIMUM LENGTH IS 76   XTCOM
      1 CD3                     CHARACTERS (38 WORDS)                       XTCOM
      1 CD3                 5 - USER REQUESTED A CONTINUATION               XTCOM
      1 CD3                                                                 XTCOM
      1 CD*************                                                      XTCOM
      1 CD4                                                                 XTCOM
      1 CD4      INTERNAL VARIABLES                                         XTCOM
      1 CD4                                                                 XTCOM
      1 CD4           CONMSG - 6 WORD ARRAY CONTAINING THE CONTINUATION     XTCOM
      1 CD4                    MESSAGE                                      XTCOM
      1 CD4           PREFIX - 4 CHARACTERS USED AS PREFIXES TO PROMPT,     XTCOM
      1 CD4                    CORRESPONDING TO CODES IN XE(5)              XTCOM
      1 CD4           RETCEP - RETURN CODE FROM XTPRM EXTENDED PROMPTING    XTCOM
      1 CD4           RETCLA - RETURN CODE FROM XTLAN LEXICAL ANALYSIS      XTCOM
      1 CD4           SUFFIX - :_ CHARACTERS APPENDED TO END OF PROMPT       XTCOM
      1 CD4                                                                 XTCOM
      1 CD*************                                                      XTCOM
      1 CD5                                                                 XTCOM
      1 CD5      XE COMMON USED:                                            XTCOM
      1 CD5                                                                 XTCOM
      1 CD5           EQUIVALENCE (XE(1),  LU),      (XE(2),   ICLASS),     XTCOM
      1 CD5          +            (XE(5),  LEVL),    (XE(145), COMBUF),     XTCOM
      1 CD5          +            (XE(145),NOTOK),   (XE(146), NOWDS)       XTCOM
      1 CD5                                                                 XTCOM
      1 CD5      XS COMMON USED:                                            XTCOM
      1 CD5                                                                 XTCOM
      1 CD5                                                                 XTCOM
```

```
1 CD5           EQUIVALENCE (XS(1),  INBUF),     (XS(81),  NOIN),               XTCOM
1 CD5          +            (XS(82), OUTBUF),    (XS(122), SSFLAG),             XTCOM
1 CD5          +            (XS(123),INLEN)                                     XTCOM
1 CD5                                                                           XTCOM
1 CD5           INBUF - 80 WORD ARRAY THAT CONTAINS THE USER'S RESPONSE         XTCOM
1 CD5                   IN R1 FORMAT                                            XTCOM
1 CD5           INLEN - LENGTH OF ORIGINAL PROMPT PASSED TO XTPRM               XTCOM
1 CD5           NOIN  - NUMBER OF WORDS OF INBUF ACTUALLY USED FOR THE          XTCOM
1 CD5                   USER'S RESPONSE                                         XTCOM
1 CD5           OUTBUF - 40 WORD AREA WHERE PROMPT IS CONSTRUCTED AND           XTCOM
1 CD5                   WRITTEN FROM.                                           XTCOM
1 CD5           SSFLAG - SYMBOLIC STRING INDICATOR FLAG                         XTCOM
1 CD5                   0 = NOT WITHIN A SYMBOLIC STRING                        XTCOM
1 CD5                   NONZERO = VALUE REPRESENTING THE INDEX INTO COMBUF      XTCOM
1 CD5                             WHERE LENGTH OF SS IS TO BE STORED            XTCOM
1 CD5                                                                           XTCOM
1 CD5       SUBROUTINES AND FUNCTIONS CALLED:                                   XTCOM
1 CD5                                                                           XTCOM
1 CD5           EXEC, KCVT, XRMOV, XRMSG, XRUPK, XTLAN, XTPRM                   XTCOM
1 CD5                                                                           XTCOM
1 CD5       PDL ROUTINES INCLUDED:                                              XTCOM
1 CD5                                                                           XTCOM
1 CD5           XTCOM, READSEG                                                  XTCOM
1 CD5                                                                           XTCOM
1 CD*************                                                               XTCOM
```

```
 87    1 BEGIN XTCOM                                                                                    XTCOM
 88    2    IF PROMPT IS NOT TOO LONG (76 CHARS) THEN                                                   XTCOM
 89    3       MOVE PREFIX CHARACTER FOR EXEC LEVEL INTO OUTPUT AREA                                    XTCOM
 90    3       MOVE PROMPT INTO OUTPUT AREA                                                             XTCOM
 91    3       MOVE SUFFIX CHARACTER INTO OUTPUT AREA                                                   XTCOM

 92    3       :LOOP:                                                                                   XTCOM
 93    3       ISSUE WRITE TO PROMPT USER                                                               XTCOM
 94    3       INITIALIZE COMMUNICATIONS BUFFER                                                         XTCOM
 95    3       TURN SYMBOLIC STRING FLAG OFF                                                            XTCOM
 96    3       INITIALIZE LA RETURN CODE TO NORMAL RETURN                                               XTCOM
 97    3       PERFORM READSEG TO READ INPUT AND CALL LEXICAL ANALYSIS                                  XTCOM
 98    3       DO WHILE LEXICAL ANALYSIS (LA) RETURN CODE SAYS CONTINUE AND                             XTCOM
 99    4          (EXEC LEVEL IS NOT INTERFACE TABLE EDITOR OR                                          XTCOM
100    4          SYMBOLIC STRING FLAG IS ON)                                                           XTCOM
101    4          CALL EXEC TO WRITE CONTINUATION MESSAGE                                               XTCOM
102    4          PERFORM READSEG TO READ INPUT AND CALL LEXICAL ANALYSIS                               XTCOM
103    3       ENDDO                                                                                    XTCOM
104    3       IF LA RETURN CODE SAYS ERROR IN RESPONSE THEN                                            XTCOM
105    4          CALL KCVT TO CONVERT OCTAL TO ASKII                                                   XTCOM
106    4          CALL XRMSG TO WRITE ERROR MESSAGE                                                     XTCOM
107    4          GO TO :LOOP: TO DISPLAY ORIGINAL PROMPT                                               XTCOM
108    3       ENDIF                                                                                    XTCOM
109    3       IF LA RETURN CODE SAYS OVERFLOW/UNDERFLOW THEN                                           XTCOM
110    4          CALL KCVT TO CONVERT OCTAL TO ASKII                                                   XTCOM
111    4          CALL XRMSG TO WRITE ERROR MESSAGE                                                     XTCOM
112    4          GO TO :LOOP: TO DISPLAY ORIGINAL PROMPT                                               XTCOM
113    3       ENDIF                                                                                    XTCOM
114    3       IF LA RETURN CODE SAYS EXTENDED PROMPTING WAS REQUESTED THEN                             XTCOM
115    4          CALL XTPRM FOR EXTENDED PROMPT                                                        XTCOM
116    4          IF EXTENDED PROMPT (EP) RETURN CODE SAYS INVALID REQUEST THEN                         XTCOM
117    5             CALL XRMSG TO WRITE ERROR MESSAGE                                                  XTCOM
118    4          ENDIF                                                                                 XTCOM
119    4          GO TO :LOOP: TO DISPLAY ORIGINAL PROMPT                                               XTCOM
120    3       ENDIF                                                                                    XTCOM
121    3       IF LA RETURN CODE SAYS COMBUF IS FULL THEN                                               XTCOM
122    4          CALL XRMSG TO WRITE ERROR MESSAGE                                                     XTCOM
123    4          GO TO :LOOP: TO DISPLAY ORIGINAL PROMPT                                               XTCOM
124    3       ENDIF                                                                                    XTCOM
125    3       SET XTCOM RETURN CODE = LA RETURN CODE                                                   XTCOM
126    2    ELSE                                                                                        XTCOM
127    3       SET XTCOM RETURN CODE = PROMPT IS TOO LONG                                               XTCOM
128    2    ENDIF                                                                                       XTCOM
129    1 END XTCOM
```

```
131    1 BEGIN READSEG
132    2    CALL XRMOV TO INITIALIZE INPUT BUFFER TO BLANKS                    XTCOM
133    2    CALL EXEC TO READ RESPONSE TO PROMPT                               XTCOM
134    2    CALL XRUPK ROUTINE TO CONVERT A2 FORMAT BUFFER TO R1               XTCOM
135    2    IF NUMBER OF WORDS READ IS NOT ZERO THEN                           XTCOM
136    3       CALL XTLAN ROUTINE TO BUILD COMMUNICATIONS BUFFER               XTCOM
137    2    ELSE                                                               XTCOM
138    3       IF LAST LA RETURN CODE WAS A CONTINUE THEN                      XTCOM
139    4          REMOVE TRAILING COMMAS FROM COMBUF                           XTCOM
140    4          SET LA RETURN CODE TO NORMAL RETURN                          XTCOM
141    3       ELSE                                                            XTCOM
142    4          SET LA RETURN CODE TO SAY USER ENTERED CR                    XTCOM
143    3       ENDIF                                                           XTCOM
144    2    ENDIF                                                              XTCOM
145    1 END READSEG                                                          XTCOM
```

```
147   1 CD************                                                                  XTLAN
148   1 CDO                                                                           XTLAN
149   1 CDO      FORTRAN CALLING PROCEDURE FOR LEXICAL ANALYSIS:                      XTLAN
150   1 CDO                                                                           XTLAN
151   1 CDO           CALL XTLAN (RETC)                                               XTLAN
152   1 CDO                                                                           XTLAN
153   1 CD************                                                                XTLAN
154   1 CD1                                                                           XTLAN
155   1 CD1      CONVERT 'INBUF' USER'S RESPONSE TO 'COMBUF' OF TOKENS                XTLAN
156   1 CD1      INDICATING  CHARACTERS, INTEGERS, REALS, ETC.                        XTLAN
157   1 CD1                                                                           XTLAN
158   1 CD************                                                                XTLAN
159   1 CD3                                                                           XTLAN
160   1 CD3      OUTPUTS IN CALLING SEQUENCE:                                         XTLAN
161   1 CD3                                                                           XTLAN
162   1 CD3           RETC - (INTEGER, 1 WORD) IS A COMPLETION CODE PASSED            XTLAN
163   1 CD3                  BACK TO CALLER AS FOLLOWS:                               XTLAN
164   1 CD3                                                                           XTLAN
165   1 CD3                0 - NORMAL RETURN. BUFFER CONTAINS RESPONSE.               XTLAN
166   1 CD3                1 - USER RESPONDED %. BUFFER CONTAINS RESPONSE             XTLAN
167   1 CD3                    UP TO AND INCLUDING %.                                 XTLAN
168   1 CD3                5 - USER REQUESTED A CONTINUATION.                         XTLAN
169   1 CD3                6 - EXTENDED PROMPTING REQUEST WWS RECEIVED. BUFFER        XTLAN
170   1 CD3                    CONTAINS RESPONSE UP TO AND INCLUDING THE REQUEST.     XTLAN
171   1 CD3                7 - COMMUNICATIONS BUFFER IS FULL.                         XTLAN
172   1 CD3              1XX - ERROR IN RESPONSE AT OR BEYOND CHARACTER XX.           XTLAN
173   1 CD3              2XX - OVERFLOW/UNDERFLOW DETECTED AT OR BEYOND               XTLAN
174   1 CD3                    CHARACTER XX,                                          XTLAN
175   1 CD3                                                                           XTLAN
176   1 CD************                                                                XTLAN
177   1 CD4                                                                           XTLAN
178   1 CD4      INTERNAL VARIABLES                                                   XTLAN
179   1 CD4                                                                           XTLAN
180   1 CD4           COMLEN - LENGTH IN WORDS OF COMBUF  =256                        XTLAN
181   1 CD4           DBLINT - DOUBLE PRECISION LOCATION TO ACCUMULATE AN             XTLAN
182   1 CD4                    INTEGER VALUE                                          XTLAN
183   1 CD4           DBLWD - DOUBLE PRECISION LOCATION TO ACCUMULATE AN INTEGER      XTLAN
184   1 CD4                   AND FRACTIONAL VALUE FOR DOUBLE PRECISION OR REAL       XTLAN
185   1 CD4           FLGCOM - COMMA FLAG                                             XTLAN
186   1 CD4                0 - LAST CHARACTER WAS NOT A COMMA                         XTLAN
187   1 CD4                1 - LAST CHARACTER WAS A COMMA                             XTLAN
188   1 CD4           FLGEND - END LOOP FLAG                                          XTLAN
189   1 CD4                0 - CONTINUE LOOP                                          XTLAN
190   1 CD4                1 - END LOOP                                               XTLAN
191   1 CD4           FLGNEG - NEGATIVE EXPONENT FLAG                                 XTLAN
192   1 CD4                0 - EXPONENT WAS POSITIVE                                  XTLAN
193   1 CD4                1 - EXPONENT WAS NEGATIVE                                  XTLAN
194   1 CD4           FLGTYP - TYPE OF REAL VALUE                                     XTLAN
195   1 CD4                0 - SINGLE PRECISION                                       XTLAN
196   1 CD4                1 - DOUBLE PRECISION                                       XTLAN
197   1 CD4           POWER - EXPONENT PART OF A REAL NUMBER                          XTLAN
198   1 CD4           RELWD - SINGLE PRECISION LOCATION FOR REAL VALUE                XTLAN
199   1 CD4           SPCHAR - 29 SPECIAL CHARACTER ARRAY CONTAINING                  XTLAN
200   1 CD4                    THE A1 FORMAT REPRESENTATION FOR:                      XTLAN
201   1 CD4                    "+-*/<>#@=&?()''$Z$.YXR,; DEW                          XTLAN
202   1 CD4                                              W IS AN EXCLAMATION POINT    XTLAN
203   1 CD4                                          X IS A CLOSED BRACKET            XTLAN
204   1 CD4                                        Y IS AN OPEN BRACKET               XTLAN
205   1 CD4                                    Z IS A BACK SLASH                      XTLAN
```

```
206   1 CD4
207   1 CD*************                                                                                    XTLAN
208   1 CD5                                                                                               XTLAN
209   1 CD5       XE COMMON USED:                                                                         XTLAN
210   1 CD5                                                                                               XTLAN
211   1 CD5            EQUIVALENCE (XE(85), TOKENS),  (XE(145), COMBUF),                                  XTLAN
212   1 CD5           +            (XE(145),NOTOK),   (XE(146), NOWDS),                                  XTLAN
213   1 CD5                                                                                               XTLAN
214   1 CD5       XS COMMON USED                                                                          XTLAN
215   1 CD5                                                                                               XTLAN
216   1 CD5            EQUIVALENCE (XS(1),  INBUF),   (XS(81),  NOCHAR),                                  XTLAN
217   1 CD5           +            (XS(122),FLAGSS),  (XS(186), SCRATCH)                                  XTLAN
218   1 CD5                                                                                               XTLAN
219   1 CD5       SUBROUTINES AND FUNCTIONS CALLED                                                        XTLAN
220   1 CD5                                                                                               XTLAN
221   1 CD5            XRPCK, XRMOV                                                                       XTLAN
222   1 CD5                                                                                               XTLAN
223   1 CD5       PDL ROUTINES INCLUDED:                                                                  XTLAN
224   1 CD5                                                                                               XTLAN
225   1 CD5            XTLAN, COMMA, ALPHA, DIGIT, DCOL,  DECPT, EORD                                     XTLAN
226   1 CD5            INTEGR,REAL,  DBL,   REPET, INVAL, SCHARS,QUOTE                                    XTLAN
227   1 CD5                                                                                               XTLAN
228   1 CD*************                                                                                    XTLAN
```

```
1 BEGIN XTLAN                                                                  XTLAN
2     SET END FLAG OFF                                                         XTLAN
2     SET LAST CHARACTER WAS A COMMA ON                                        XTLAN
2     INITIALIZE INDEX INTO INPUT BUFFER                                       XTLAN
2     INITIALIZE RETURN CODE TO NORMAL RETURN                                  XTLAN
2     DO WHILE END FLAG IS OFF                                                 XTLAN
3        IF INPUT BUFFER HAS BEEN COMPLETELY SCANNED THEN                      XTLAN
4           IF LAST CHARACTER WAS A COMMA OR A SYMBOLIC STRING IS STILL OPEN THEN  XTLAN
5              SET RETURN CODE TO SAY CONTINUATION REQUESTED                   XTLAN
4           ENDIF                                                              XTLAN
4           SET END FLAG ON                                                    XTLAN
3        ELSE                                                                  XTLAN
4           ERREXIT IF COMBUF IS FULL PERFORM COMFUL                           XTLAN
4           IF INPUT CHARACTER IS A COMMA THEN                                 XTLAN
5              PERFORM COMPA                                                   XTLAN
4           ELSE                                                               XTLAN
5              SET LAST CHARACTER WAS A COMMA OFF                              XTLAN
5              IF INPUT CHARACTER IS AN ALPHABETIC CHARACTER THEN              XTLAN
6                 PERFORM ALPHA                                                XTLAN
5              ELSE                                                            XTLAN
6                 IF INPUT CHARACTER IS A DIGIT THEN                           XTLAN
7                    PERFORM DIGIT                                             XTLAN
6                 ELSE                                                         XTLAN
7                    PERFORM SCHARS                                            XTLAN
6                 ENDIF                                                        XTLAN
5              ENDIF                                                           XTLAN
4           ENDIF                                                              XTLAN
3        ENDIF                                                                 XTLAN
2     ENDDO                                                                    XTLAN
2     STORE END OF BUFFER TOKEN IN COMBUF                                      XTLAN
2     INCREMENT #TOKENS BY 1                                                   XTLAN
1 END XTLAN                                                                    XTLAN
1 *                                                                            XTLAN
1 *                                                                            XTLAN
1 *                                                                            XTLAN
1 BEGIN COMMA                                                                  XTLAN
2     SET LAST CHARACTER WAS A COMMA ON                                        XTLAN
2     ERREXIT IF THERE IS NO ROOM IN COMBUF FOR THIS TOKEN PERFORM COMFUL      XTLAN
2     STORE COMMA TOKEN IN COMBUF                                              XTLAN
2     INCREMENT #WORDS IN COMBUF BY 1                                          XTLAN
2     INCREMENT #TOKFNS BY 1                                                   XTLAN
2     GET NEXT INPUT CHARACTER                                                 XTLAN
1 END COMMA                                                                    XTLAN
```

```
1 BEGIN ALPHA                                                          XTLAN
  2   ERREXIT IF THERE IS NO ROOM IN COMBUF FOR THIS TOKEN PERFORM COMFUL  XTLAN
  2   CALL XRMOV TO INITIALIZE TEMPORARY BUFFER WITH 6 BLANKS           XTLAN
  2   DO WHILE (INPUT CHARACTER IS AN ALPHA CHARACTER OR                XTLAN
  3     INPUT CHARACTER IS A NUMERIC OR                                 XTLAN
  3     INPUT CHARACTER IS AN EXCLAMATION POINT) AND                    XTLAN
  3     INPUT BUFFER HAS NOT BEEN COMPLETELY SCANNED                    XTLAN
  3     MOVE CHARACTER INTO TEMPORARY BUFFER                            XTLAN
  3     GET NEXT INPUT CHARACTER                                        XTLAN
  2   ENDDO                                                             XTLAN
  2   SET CHARACTER COUNT = 6                                           XTLAN
  2   STORE CHARACTER NAME TOKEN IN COMBUF                              XTLAN
  2   CALL XRPCK ROUTINE TO PACK CHARACTERS INTO COMBUF                 XTLAN
  2   INCREMENT #WORDS IN COMBUF BY 4                                   XTLAN
  2   INCREMENT #TOKENS BY 1                                            XTLAN
1 END ALPHA                                                            XTLAN
1 *                                                                    XTLAN
1 *                                                                    XTLAN
1 *                                                                    XTLAN
1 BEGIN DIGIT                                                          XTLAN
  2   INITIALIZE POWER TO ZERO                                          XTLAN
  2   PERFORM DCOL                                                      XTLAN
  2   IF INPUT BUFFER IS NOT EXHAUSTED THEN                             XTLAN
  3     IF INPUT CHARACTER IS A . THEN                                  XTLAN
  4       PERFORM DECPT                                                 XTLAN
  3     ELSE                                                            XTLAN
  4       IF INPUT CHARACTER IS AN "E" OR A "D" THEN                    XTLAN
  5         CONVERT INTEGER VALUE TO DOUBLE PRECISION VALUE             XTLAN
  5         PERFORM EORD                                                XTLAN
  4       ELSE                                                          XTLAN
  5         IF INPUT CHARACTER IS AN "R" THEN                           XTLAN
  6           PERFORM REPET                                             XTLAN
  5         ELSE                                                        XTLAN
  6           PERFORM INTEGR                                            XTLAN
  5         ENDIF                                                       XTLAN
  4       ENDIF                                                         XTLAN
  3     ENDIF                                                           XTLAN
  2   ELSE                                                              XTLAN
  3     PERFORM INTEGR                                                  XTLAN
  2   ENDIF                                                             XTLAN
1 END DIGIT                                                            XTLAN
```

```
316    1 BEGIN DCOL                                                                                                    XTLAN
317    2    SET INTEGER = 0                                                                                            XTLAN
318    2    SET COUNTER = 0                                                                                            XTLAN
319    2    DO WHILE CHARACTER IS A DIGIT AND                                                                          XTLAN
320    3       INPUT BUFFER IS NOT EXHAUSTED                                                                           XTLAN
321    3       SET INTEGER =(INTEGER * 10 ) + CURRENT CHARACTER - 48                                                   XTLAN
322    3       ERREXIT IF OVERFLOW/UNDERFLOW IS DETECTED PERFORM OVFLOW                                                XTLAN
323    3       INCREMENT COUNTER BY1                                                                                   XTLAN
324    3       GET NEXT CHARACTER                                                                                      XTLAN
325    2    ENDDO                                                                                                      XTLAN
326    1 END DCOL                                                                                                      XTLAN
327    1 *                                                                                                             XTLAN
328    1 *                                                                                                             XTLAN
329    1 *                                                                                                             XTLAN
330    1 BEGIN DECPT                                                                                                   XTLAN
331    2    CONVERT INTEGER VALUE TO DOUBLE PRECISION VALUE                                                            XTLAN
332    2    GET NEXT INPUT CHARACTER                                                                                   XTLAN
333    2    IF INPUT BUFFER IS NOT EXHAUSTED THEN                                                                      XTLAN
334    3       IF INPUT CHARACTER IS A DIGIT THEN                                                                      XTLAN
335    4          PERFORM DCOL                                                                                         XTLAN
336    4          ADD FRACTIONAL PART TO DOUBLE PRECISION VALUE                                                        XTLAN
337    4          ERREXIT IF OVERFLOW/UNDERFLOW IS DETECTED PERFORM OVFLOW                                             XTLAN
338    3       ENDIF                                                                                                   XTLAN
339    3       IF INPUT CHARACTER IS AN "E" OR A "D" THEN                                                              XTLAN
340    4          PERFORM EORD                                                                                         XTLAN
341    3       ELSE                                                                                                    XTLAN
342    4          PERFORM REAL                                                                                         XTLAN
343    3       ENDIF                                                                                                   XTLAN
344    2    ELSE                                                                                                       XTLAN
345    3       PERFORM REAL                                                                                            XTLAN
346    2    ENDIF                                                                                                      XTLAN
347    1 END DECPT                                                                                                     XTLAN
```

```
1 BEGIN EORD
2   IF INPUT CHARACTER IS AN "E" THEN
3     SET TYPE FLAG TO "E"
2   ELSE
3     SET TYPE FLAG TO "D"
2   ENDIF
2   GET NEXT CHARACTER
2   ERREXIT IF INPUT BUFFER IS EXHAUSTED
2   SET NEGATIVE FLAG OFF
2   IF CHARACTER IS A - THEN
3     SET NEGATIVE FLAG ON
3     GET NEXT CHARACTER
2   ELSE
2     IF CHARACTER IS A + THEN
3       GET NEXT CHARACTER
2     ENDIF
2   ENDIF
2   ERREXIT IF INPUT BUFFER IS EXHAUSTED OR
2   ERREXIT IF CHARACTER IS NOT A DIGIT PERFORM INVAL
2   PERFORM DCOL
2   IF NEGATIVE FLAG IS ON THEN
3     SET POWER = -POWER
2   ENDIF
2   IF TYPE FLAG IS "E" THEN
3     PERFORM REAL
2   ELSE
3     PERFORM DBL
2   ENDIF
1 END EORD
```

```
379    1 BEGIN INTEGR                                                                                                  XTLAN
380    2    ERREXIT IF THERE IS NO ROOM IN COMBUF FOR THIS TOKEN PERFORM COMFUL                                         XTLAN
381    2    CONVERT NUMBER TO INTEGER                                                                                  XTLAN
382    2    ERREXIT IF OVERFLOW/UNDERFLOW IS DETECTED PERFORM OVFLOW                                                   XTLAN
383    2    STORE INTEGER TOKEN IN COMBUF                                                                              XTLAN
384    2    INCREMENT #WORDS IN COMBUF BY 2                                                                            XTLAN
385    2    INCREMENT #TOKENS BY 1                                                                                     XTLAN
386    1 END INTGR                                                                                                     XTLAN
387    1 *                                                                                                             XTLAN
388    1 *                                                                                                             XTLAN
389    1 *                                                                                                             XTLAN
390    1 BEGIN REAL                                                                                                    XTLAN
391    2    ERREXIT IF THERE IS NO ROOM IN COMBUF FOR THIS TOKEN PERFORM COMFUL                                         XTLAN
392    2    SET REAL = DOUBLE PRECISION * 10 ** POWER                                                                  XTLAN
393    2    ERREXIT IF OVERFLOW/UNDERFLOW IS DETECTED PERFORM OVFLOW                                                   XTLAN
394    2    STORE REAL TOKEN IN COMBUF                                                                                 XTLAN
395    2    INCREMENT #WORDS IN COMBUF BY 3                                                                            XTLAN
396    2    INCREMENT #TOKENS BY 1                                                                                     XTLAN
397    1 END REAL                                                                                                      XTLAN
398    1 *                                                                                                             XTLAN
399    1 *                                                                                                             XTLAN
400    1 *                                                                                                             XTLAN
401    1 BEGIN DBL                                                                                                     XTLAN
402    2    ERREXIT IF THERE IS NO ROOM IN COMBUF FOR THIS TOKEN PERFORM COMFUL                                         XTLAN
403    2    SET DOUBLE = DOUBLE PRECISION * 10 ** POWER                                                                XTLAN
404    2    ERREXIT IF OVERFLOW/UNDERFLOW IS DETECTED PERFORM OVFLOW                                                   XTLAN
405    2    STORE DOUBLE TOKEN IN COMBUF                                                                               XTLAN
406    2    INCREMENT #WORDS IN COMBUF BY 4                                                                            XTLAN
407    2    INCREMENT #TOKENS BY 1                                                                                     XTLAN
408    1 END DBL                                                                                                       XTLAN
409    1 *                                                                                                             XTLAN
410    1 *                                                                                                             XTLAN
411    1 *                                                                                                             XTLAN
412    1 BEGIN REPET                                                                                                   XTLAN
413    2    ERREXIT IF THERE IS NO ROOM IN COMBUF FOR THIS TOKEN PERFORM COMFUL                                         XTLAN
414    2    ERREXIT IF NUMBER OF TIMES TO REPEAT IS ZERO PERFORM INVAL                                                  XTLAN
415    2    STORE REPEAT TOKEN IN COMBUF                                                                               XTLAN
416    2    INCREMENT #WORDS IN COMBUF BY 2                                                                            XTLAN
417    2    INCREMENT #TOKENS BY 1                                                                                     XTLAN
418    2    GET NEXT CHARACTER                                                                                         XTLAN
419    1 END REPET                                                                                                     XTLAN
```

```
421   1 BEGIN SCHARS                                                                                                XTLAN
422   2 *         CHARACTER TABLE:                                                                                  XTLAN
423   2 *             "+-*/<>#@=&?()''XZ$.YXNN;                                                                     XTLAN
424   2 *                                      NN ARE INVALID CHARACTERS HERE                                       XTLAN
425   2 *                                    X IS A CLOSED BRACKET                                                  XTLAN
426   2 *                                  Y IS AN OPEN BRACKET                                                     XTLAN
427   2 *                                Z IS A BACK SLASH                                                          XTLAN
428   2     SET J=1                                                                                                 XTLAN
429   2     STARTSEARCH WHILE J<= #CHARACTERS IN TABLE                                                              XTLAN
430   2     EXITIF INPUT CHARACTER MATCHES CHARACTER (J) IN TABLE                                                   XTLAN
431   3        INCREMENT J BY 1                                                                                     XTLAN
432   2     ENDLOOP                                                                                                 XTLAN
433   3        PERFORM INVAL - NO RETURN                                                                            XTLAN
434   2     ENDSEARCH                                                                                               XTLAN
435   2     SET NEXT FLAG ON                                                                                        XTLAN
436   2     *       "   +   -   *   /   <   >   #   @   =   &   ?                                                 XTLAN
437   2     CASE J(:A:,:F:,:F:,:F:,:F:,:F:,:F:,:F:,:F:,:F:,:F:,:F:,:F:,:@:,                                        XTLAN
438   3        * (   )   '   '   ~   Z   $   .   Y   X       N       N     ;                                      XTLAN
439   3        (:F:,:F:,:C:,:C:,:D:,:F:,:F:,:E:,:F:,:F:,:INVAL:,:INVAL:,:F:)                                        XTLAN

440   3        :A:                                                                                                  XTLAN
441   3        SET NEXT FLAG OFF                                                                                    XTLAN
442   3        PERFORM QUOTE                                                                                        XTLAN

443   3        :@:                                                                                                  XTLAN
444   3        SET RETURN CODE TO SAY EXTENDED PROMPT REQUESTED                                                     XTLAN
445   3        SET END FLAG ON                                                                                      XTLAN

446   3        :C:                                                                                                  XTLAN
447   3        IF SYMBOLIC STRING FLAG IS OFF THEN                                                                  XTLAN
448   4           SET SYMBOLIC STRING FLAG TO CURRENT COMBUF INDEX + 1                                              XTLAN
449   3        ELSE                                                                                                 XTLAN
450   4           SET COMBUF(SYMBOLIC STRING FLAG) =CURRENT COMBUF INDEX - SYMBOLIC STRING FLAG                     XTLAN
451   4           SET J = J+1 TO STORE SYMBOLIC STRING CLOSE TOKEN                                                  XTLAN
452   4           SET SYMBOLIC STRING FLAG OFF                                                                      XTLAN
453   3        ENDIF                                                                                                XTLAN

454   3        :D:                                                                                                  XTLAN
455   3        SET RETURN CODE TO SAY X ENTERED                                                                     XTLAN
456   3        SET END FLAG ON                                                                                      XTLAN

457   3        :E:                                                                                                  XTLAN
458   3        IF FOLLOWING CHARACTER IS A DIGIT THEN                                                               XTLAN
459   4           SET NEXT FLAG OFF                                                                                 XTLAN
460   4           SET INTEGER = 0                                                                                   XTLAN
461   4           SET POWER = 0                                                                                     XTLAN
462   4           PERFORM DECPT                                                                                     XTLAN
463   3        ENDIF                                                                                                XTLAN

464   3        :F:                                                                                                  XTLAN
465   2     ENDCASE                                                                                                 XTLAN
466   2     IF NEXT FLAG IS ON                                                                                      XTLAN
467   3        STORE TOKEN (J) IN COMBUF                                                                            XTLAN
468   3        INCREMENT #WORDS IN COMBUF BY 1                                                                      XTLAN
469   3        IF TOKEN IS BEGIN SYMBOLIC STRING THEN                                                               XTLAN
470   4           INCREMENT #WORDS IN COMBUF BY 1                                                                   XTLAN
471   3        ENDIF                                                                                                XTLAN
472   3        INCREMENT #TOKENS BY 1                                                                               XTLAN
473   3        GET NEXT CHARACTER                                                                                   XTLAN
```

```
474    2    ENDIF
475    1 END SCHARS
```

```
477   1 BEGIN QUOTE                                                                                                  XTLAN
478   2    GET NEXT CHARACTER                                                                                        XTLAN
479   2    SET #CHARACTERS =0                                                                                        XTLAN
480   2    DO WHILE (INPUT CHARACTER IS NOT A QUOTE AND                                                              XTLAN
481   3       INPUT BUFFER HAS NOT BEEN COMPLETELY SCANNED) OR                                                       XTLAN
482   3       (INPUT CHARACTER IS A QUOTE AND                                                                        XTLAN
483   3       NEXT CHARACTER IS A QUOTE AND                                                                          XTLAN
484   3       INPUT BUFFER HAS NOT BEEN COMPLETELY SCANNED)                                                          XTLAN
485   3       INCREMENT #CHARACTERS BY 1                                                                             XTLAN
486   3       MOVE CHARACTER INTO TEMPORARY BUFFER                                                                   XTLAN
487   3       IF INPUT CHARACTER IS A QUOTE THEN                                                                     XTLAN
488   4          GET NEXT CHARACTER                                                                                  XTLAN
489   3       ENDIF                                                                                                  XTLAN
490   3       GET NEXT CHARACTER                                                                                     XTLAN
491   2    ENDDO                                                                                                     XTLAN
492   2    ERREXIT IF THERE IS NO ROOM IN COMBUF FOR THIS TOKEN PERFORM COMFUL                                       XTLAN
493   2    ERREXIT IF LENGTH OF CHARACTER STRING IS 0 OR                                                             XTLAN
494   2    ERREXIT IF INPUT CHARACTER IS NOT A QUOTE PERFORM INVAL                                                   XTLAN
495   2    STORE CHARACTER STRING TOKEN IN COMBUF                                                                    XTLAN
496   2    STORE #CHARACTERS IN COMBUF                                                                               XTLAN
497   2    CALL XRPCK TO PACK CHARACTERS INTO COMBUF                                                                 XTLAN
498   2    INCREMENT #WORDS IN COMBUF BY 2+((#CHARACTERS+1)/2)                                                       XTLAN
499   2    INCREMENT #TOKENS BY 1                                                                                    XTLAN
500   2    GET NEXT CHARACTER                                                                                        XTLAN
501   1 END QUOTE                                                                                                    XTLAN
```

```
503   1 BEGIN INVAL                                                          XTLAN
504   2    SET RETURN CODE TO SAY INVALID RESPONSE                           XTLAN
505   2    SET END FLAG ON                                                   XTLAN
506   1 EXIT XTLAN                                                           XTLAN
507   1 END INVAL                                                            XTLAN
508   1 *                                                                    XTLAN
509   1 *                                                                    XTLAN
510   1 *                                                                    XTLAN
511   1 BEGIN COMFUL                                                         XTLAN
512   2    SET RETURN CODE TO SAY COMBUF IS FULL                             XTLAN
513   2    SET END FLAG ON                                                   XTLAN
514   1 EXIT XTLAN                                                           XTLAN
515   1 END COMFUL                                                           XTLAN
516   1 *                                                                    XTLAN
517   1 *                                                                    XTLAN
518   1 *                                                                    XTLAN
519   1 BEGIN OVFLOW                                                         XTLAN
520   2    SET RETIRN CODE TO SAY OVERFLCW/UNDERFLOW                         XTLAN
521   2    SET END FLAG ON                                                   XTLAN
522   1 EXIT XTLAN                                                           XTLAN
523   1 END OVFLOW                                                           XTLAN
```

```
1 CD0      FORTRAN CALLING PROCEDURE                                         XTPRM
1 CD0                                                                        XTPRM
1 CD0   CALL XTPRM                                                           XTPRM
1 CD0                                                                        XTPRM
1 C*********                                                                 XTPRM
1 CD1                                                                        XTPRM
1 CD1      XTPRM IS CALLED BY XTCOM TO PROVIDE EXTENDED PROMPTING WHEN A     XTPRM
1 CD1      ? IS ENTERED FROM THE USER TERMINAL.                              XTPRM
1 CD1                                                                        XTPRM
1 C*********                                                                 XTPRM
1 CD2      INPUT                                                             XTPRM
1 CD2   XE COMMON - CARTRG, COMBUF, FLAGS, LU, NOPROC, PRCKAM, SUBSTA,       XTPRM
1 CD2               TOKENS                                                   XTPRM
1 CD2   XB COMMON - ARGPTR, WKBUF (FROM INTERFACE TABLE EDITOR)              XTPRM
1 CD2   XS COMMON - PRMLEN (FROM XTCOM)                                      XTPRM
1 CD2                                                                        XTPRM
1 CD2   VARIOUS FDS PROMPT FILES (SEE INTERNAL VARIABLE TABLE)               XTPRM
1 CD2                                                                        XTPRM
1 C*********                                                                 XTPRM
1 CD3      OUTPUT                                                            XTPRM
1 CD3   XE COMMON - COMBUF (USED FOR SCRATCH SPACE)                          XTPRM
1 CD3                                                                        XTPRM
1 CD3   LISTING OF APPROPRIATE EXTENDED PROMPTS                              XTPRM
1 CD3                                                                        XTPRM
1 C*********                                                                 XTPRM
1 CD4      INTERNAL VARIABLES                                                XTPRM
1 CD4   CONTIN - CONTINUATION INDICATOR (1) FOR CURRENT TABLE ENTRY          XTPRM
1 CD4   FILE   - FILE NAME OF CURRENT TABLE ENTRY                            XTPRM
1 CD4   INDEX  - INDEX TO CURRENT TABLE ENTRY                                XTPRM
1 CD4   L      - RECORD NUMBER OF TEXT OR SYNTAX BLOCK CORRESPONDING TO      XTPRM
1 CD4            FIRST LIST ITEM IN RECORD 1 (SEE SDD SECTION 6.2.4.3)       XTPRM
1 CD4   MESAGE - MESSAGE NUMBER, IF ANY, FOR CURRENT TABLE ENTRY             XTPRM
1 CD4   N      - NUMBER OF LIST ITEMS IN RECORD(S) 1 (AND 2) (SEE SDD        XTPRM
1 CD4            SECTION 6.2.4.3)                                            XTPRM
1 CD4   RECORD - RECORD NUMBER WITH WHICH TO BEGIN PROCESSING FOR CURRENT    XTPRM
1 CD4            TABLE ENTRY                                                 XTPRM
1 CD4   SEARCH - LIST SEARCH FLAG (1) FOR CURRENT TABLE ENTRY                XTPRM
1 CD4   SIZE   - LIST ITEM SIZE OF CURRENT TABLE ENTRY (NEGATIVE INDICATES   XTPRM
1 CD4            LAST CHARACTER TO BE MASKED)                                XTPRM
1 CD4   TABLE  - PROCESSING CONTROL TABLE FOR VARIOUS SYNTAX CONDITIONS      XTPRM
1 CD4            AS FOLLOWS                                                  XTPRM
1 CD4--------------------------------------------------------------------   XTPRM
1 CD4ENTRY   SYNTAX CONDITION   INDEX    FILE  REC SIZE SRCH MESSAGE CONTINUE XTPRM
1 CD4I  1 I X:?                  I  1 I >XDPRM  1    2   NO   NONE    NO  I  XTPRM
1 CD4I  2 I X:R?                 I  9 I >XDPRM  1    2   YES  XT06    NO  I  XTPRM
1 CD4I  3 I #:?                  I 17 I >XSPRM  1    2   NO   NONE    YES I  XTPRM
1 CD4I  4 I $:?, $P:?, #P:?      I 25 I  XLIBD  2   -3   NO   NONE    NO  I  XTPRM
1 CD4I  5 I #:R?                 I 33 I >XSPRM  1    2   YES  NONE    YES I  XTPRM
1 CD4I  6 I $:R?, $P:R?, #P:R?   I 41 I >PPOC   3  128   NO   XT07    NO  I  XTPRM
1 CD4I  7 I /:?                  I 49 I >XIPRM  1    2   NO   NONE    YES I  XTPRM
1 CD4I  8 I                      I 57 I >PROC   1    3   NO   NONE    NO  I  XTPRM
1 CD4I  9 I /:R?                 I 65 I >XIPRM  1    2   YES  NONE    YES I  XTPRM
1 CD4I 10 I /P:?, /P:P?          I 73 I >PRO?   1    3   YES  XT08    NO  I  XTPRM
1 CD4--------------------------------------------------------------------   XTPRM
1 CD4   WHERE P AND R INDICATE A PROMPT AND RESPONSE                         XTPRM
1 CD4--------------------------------------------------------------------   XTPRM
1 CD4                                                                        XTPRM
1 C*********                                                                 XTPRM
1 CD5      NOTES                                                             XTPRM
```

```
584    1 CD5   USES CLOSE, EXEC, IAND, OPEN, POSNT, READF, XPCPR, XRI6, XRMON,                    XTPRM
585    1 CD5        XRMSG, XRPCK, XRSET, XRUPK, XUDBG                                           XTPRM
586    1 CD5                                                                                    XTPRM
587    1 CD5   ANY SYNTAX CONDITION OTHER THAN THAT LISTED IS AN ERROR.  THE ONLY               XTPRM
588    1 CD5   VALID RESPONSE IS '?' OR 'NAME?'.                                                XTPRM
589    1 CD5                                                                                    XTPRM
590    1 C*********                                                                             XTPRM
```

```
1 BEGIN XTPRM                                                                      XTPRM
2    PERFORM SETUP TO COMPLETE CONTROL TABLE AND INDEX TO APPROPRIATE ENTRY        XTPRM
2    DO UNTIL 'NO CONTINUE' FOUND (0 IN ENTRY CONTINUE FIELD)                      XTPRM
3       OPEN DESIGNATED FILE                                                       XTPRM
3       IF OPEN SUCCESSFUL, THEN                                                   XTPRM
4          POSITION TO INDICATED STARTING RECORD AND READ                          XTPRM
3       EXIT TO :ERR09: IF FAILURE                                                 XTPRM
4          IF TABLE SIZE FIELD < 128 (NOT A LIST RECORD), THEN                     XTPRM
5             IF SIZE > 0 (NO LAST CHARACTER MASKING & POSSIBLE SPANNING), THEN    XTPRM
6                IF RECORD SPANNED (N*SIZE > 126), THEN                            XTPRM
7                   READ SECOND RECORD AND APPEND TO FIRST RECORD DATA             XTPRM
6                EXIT TO :ERR09: IF FAILURE                                        XTPRM
6                ENDIF                                                             XTPRM
5             ELSE                                                                 XTPRM
6                SET SIZE POSITIVE                                                 XTPRM
6                DO FOR EACH LIST ITEM (1-N)                                       XTPRM
7                   BLANK LAST CHARACTER                                           XTPRM
6                ENDDO                                                             XTPRM
5             ENDIF                                                                XTPRM
5             IF LIST SEARCHING IS INDICATED (SEARCH FIELD = 1), THEN              XTPRM
6                START SEARCH WHILE LIST ITEMS REMAIN TO BE EXAMINED               XTPRM
6                EXIT IF TOKEN LOCATED IN LIST                                     XTPRM
7                   POSITION TO APPROPRIATE RECORD (I+L-1) AND READ                XTPRM
6                EXIT TO :ERR09: IF FAILURE                                        XTPRM
7                   SET SIZE TO 128 (ENTIRE RECORD TO BE DISPLAYED)                XTPRM
6                END LOOP                                                          XTPRM
7                   IF TABLE MESSAGE NUMBER FIELD > 0, THEN                        XTPRM
8                      CALL XRMSG TO DISPLAY 'NOT VALID ...' MESSAGE               XTPRM
7                   ENDIF                                                          XTPRM
6                EXIT TO ENDDO                                                     XTPRM
6                END SEARCH                                                        XTPRM
5             ENDIF                                                                XTPRM
4          ENDIF                                                                   XTPRM
4          PERFORM DSPLAY                                                          XTPRM
3       ELSE OPEN ERROR                                                            XTPRM
4          IF FILE NOT FOUND AND TABLE MESSAGE NUMBER FIELD > 0                    XTPRM
5             CALL XRMSG TO DISPLAY 'NOT VALID ...' MESSAGE                        XTPRM
4          ELSE                                                                    XTPRM

5             :ERR09: CALL XRMSG TO DISPLAY 'FILE MANAGER ERROR ...' MESSAGE       XTPRM
5             CLOSE FILE                                                           XTPRM
4          ENDIF                                                                   XTPRM
3       ENDIF                                                                      XTPRM
2    ENDDO                                                                         XTPRM
1 END XTPRM                                                                        XTPRM
```

```
637   1 BEGIN SETUP                                                                                                    XTPRM
638   1 EXIT TO :ERR02: IF FIRST TOKEN NOT ? OR NAME FOLLOWED BY ?                                                     XTPRM
639   2    CASE (:%:, :$:, :#:, :/:) SUBSTA                                                                            XTPRM

640   3       :%:                                                                                                    XTPRM
641   3       IF FIRST TOKEN = ?, THEN                                                                               XTPRM
642   4          SET TABLE INDEX TO FIRST ENTRY                                                                      XTPRM
643   3       ELSE                                                                                                   XTPRM
644   4          SET TABLE INDEX TO SECOND ENTRY                                                                     XTPRM
645   3       ENDIF                                                                                                  XTPRM

646   3       :$:                                                                                                    XTPRM
647   3       IF FIRST TOKEN = ?, THEN                                                                               XTPRM
648   4          SET TABLE INDEX TO FOURTH ENTRY                                                                     XTPRM
649   3       ELSE                                                                                                   XTPRM
650   4          SET TABLE INDEX TO SIXTH ENTRY                                                                      XTPRM
651   4          FORM SIXTH ENTRY FILE NAME FROM > AND TOKEN                                                         XTPRM
652   3       ENDIF                                                                                                  XTPRM

653   3       :#:                                                                                                    XTPRM
654   3       IF PROMPT LENGTH = 0, THEN                                                                             XTPRM
655   4          ASSUME TABLE INDEX OF THIRD ENTRY                                                                   XTPRM
656   3       ELSE                                                                                                   XTPRM
657   4          ASSUME TABLE INDEX OF FOURTH ENTRY                                                                  XTPRM
658   3       ENDIF                                                                                                  XTPRM
659   3       IF FIRST TOKEN IS A NAME, THEN                                                                         XTPRM
660   4          INCREMENT TABLE INDEX BY TWO ENTRIES (NOW FIFTH OR SIXTH)                                           XTPRM
661   4          FORM SIXTH ENTRY FILE NAME FROM > AND TOKEN                                                         XTPRM
662   3       ENDIF                                                                                                  XTPRM

663   3       :/:                                                                                                    XTPRM
664   3       FORM TENTH ENTRY FILE NAME FROM > AND IT EDITOR CURRENT PROCESSOR NAME                                  XTPRM
665   3       ASSUME TABLE INDEX OF TENTH ENTRY                                                                      XTPRM
666   3       IF FIRST TOKEN = ?, THEN                                                                               XTPRM
667   4          IF PROMPT LENGTH = 0, THEN                                                                          XTPRM
668   5             CHANGE TABLE INDEX TO SEVENTH ENTRY                                                              XTPRM
669   5             COPY FILE NAME FROM ENTRY TEN TO ENTRY EIGHT                                                     XTPRM
670   4          ELSE                                                                                                XTPRM
671   5             CHANGE TOKEN TO KEYWORD CURRENTLY BEING PROCESSED BY IT EDITOR                                   XTPRM
672   4          ENDIF                                                                                               XTPRM
673   3       ELSE                                                                                                   XTPRM
674   4          IF PROMPT LENGTH = 0, THEN                                                                          XTPRM
675   5             CHANGE TABLE INDEX TO NINETH ENTRY                                                               XTPRM
676   4          ENDIF                                                                                               XTPRM
677   3       ENDIF                                                                                                  XTPRM
678   2    END CASE                                                                                                  XTPRM
679   1 EXIT SETUP                                                                                                   XTPRM

680   2    :ERR02:  EXIT XTPRM WITH INVALID REQUEST FOR EXTENDED PROMPTING                                           XTPRM
681   1 END SETUP                                                                                                    XTPRM
```

```
683  1 BEGIN DSPLAY                                                  XTPRM
684  2    IF SIZE < 128, THEN                                        XTPRM
685  3       DO UNTIL ALL LIST ITEMS DISPLAYED                       XTPRM
686  4             BLANK LINE BUFFER                                 XTPRM
687  4             MOVE EIGHT (OR REMAING) ITEMS INTO BUFFER         XTPRM
688  4             DISPLAY LINE                                      XTPRM
689  3          ENDDO                                                XTPRM
690  2       ELSE                                                    XTPRM
691  3          DISPLAY EXTENDED PROMPT                              XTPRM
692  2       ENDIF                                                   XTPRM
693  1 END DSPLAY                                                    XTPRM
```

```
 SYMBOL DEFINITION TABLE
:#      :   653
:A      :   440
 ALPHA      274
:B      :   443
:C      :   446
 COMFUL     511
 COMMA      265
:D      :   454
 DBL        401
 DCOL       316
 DECPT      330
 DIGIT      293
 DSPLAY     683
:E      :   457
 EORD       349
:ERR02  :   680
:ERR09  :   630
:F      :   464
 INTEGR     379
 INVAL      503
:LOOP   :    92
 OVFLOW     519
 QUOTE      477
 READSE     131
 REAL       390
 REPET      412
 SCHARS     421
 SETUP      637
 XTCOM       87
 XTLAN      230
 XTPRM      592
:$      :   646
:%      :   640
:/      :   663

@XQT F.PDLIST
```

```
1 CD0      FORTRAN CALLING PROCEDURE                                     XUDBG
1 CD0                                                                    XUDBG
1 CD0   CALL XUDBG (LU, ID)                                              XUDBG
1 CD0                                                                    XUDBG
1 C*********                                                             XUDBG
1 CD1                                                                    XUDBG
1 CD1   XUDBG PROVIDES THE PROGRAMMER WITH A CALLABLE INTERACTIVE MEMORY XUDBG
1 CD1   DUMP AND/OR MODIFICATION FACILITY                                XUDBG
1 CD1                                                                    XUDBG
1 C*********                                                             XUDBG
1 CD2      INPUT (CALLING SEQUENCE)                                      XUDBG
1 CD2   LU      - LOGICAL UNIT TO INTERACT WITH IN INVOKING XUDBG OPTIONS XUDBG
1 CD2   ID      - THREE WORD ASCII ARRAY USED AS A HEADER TO IDENTIFY XUDBG XUDBG
1 CD2             CALLER                                                 XUDBG
1 CD2                                                                    XUDBG
1 CD2      INPUT (INTERACTIVE)                                           XUDBG
1 CD2   OPTION (S,M,E): - ONE ASCII CHARACTER IDENTIFYING XUDBG OPTION   XUDBG
1 CD2                     S = SNAP OUT (DUMP) MEMORY                     XUDBG
1 CD2                     M = MODIFY MEMORY                              XUDBG
1 CD2                     E = EXIT XUDBG                                 XUDBG
1 CD2   START:          - OCTAL MEMORY ADDRESS IN THE USERS MAP OF FIRST XUDBG
1 CD2                     WORD TO BE SNAPPED OR MODIFIED                 XUDBG
1 CD2   STOP:           - OCTAL MEMORY ADDRESS OF LAST WORD TO BE SNAPPED XUDBG
1 CD2   OUTPUT UNIT:    - LOGICAL UNIT NUMBER TO WHICH SNAP IS TO BE     XUDBG
1 CD2                     WRITTEN                                        XUDBG
1 CD2   VALUES:         - ARRAY OF OCTAL VALUES TO BE STORED IN MEMORY   XUDBG
1 CD2                     BEGINNING AT LOCATION 'START'.  WHEN INPUT     XUDBG
1 CD2                     'VALUES' MUST NOT EXCEED 50 CHARACTERS.  NULL  XUDBG
1 CD2                     FIELDS, I.E., SUCCESSIVE COMMAS, INDICATE WORDS XUDBG
1 CD2                     OF ZERO TO BE STORED.                          XUDBG
1 CD2                                                                    XUDBG
1 C*********                                                             XUDBG
1 CD3      OUTPUT (TO 'LU')                                              XUDBG
1 CD3   HEADER          - '*** XUDBG FROM IDIDID'                        XUDBG
1 CD3   PROMPTS         - (SEE INPUT)                                    XUDBG
1 CD3                                                                    XUDBG
1 CD3      OUTPUT (TO 'OUTPUT UNIT')                                     XUDBG
1 CD3   HEADER          - '*** XUDBG FROM IDIDID'                        XUDBG
1 CD3   SNAP            - 102 WORD DUMP FORMATTED LINE (SEE XUDPL)       XUDBG
1 CD3                                                                    XUDBG
1 C*********                                                             XUDBG
1 CD4      INTERNALS                                                     XUDBG
1 CD4   BASE   - NUMBER BASE FOR PROMPT AND CONVERSION PROCEDURE         XUDBG
1 CD4   CLASS  - CLASS I/O NUMBER FOR TERMINAL INPUT                     XUDBG
1 CD4   LENGTH - LENGTH OF CHARACTER STRING BEING MANIPULATED            XUDBG
1 CD4   LINE   - EIGHT WORD BUFFER OF WORDS TO BE SNAPPED                XUDBG
1 CD4   LUI    - TERMINAL LOGICAL UNIT WITH INPUT ECHO BIT SET           XUDBG
1 CD4   LUO    - LOGICAL UNIT FOR SNAP OUTPUT                            XUDBG
1 CD4   OPTN   - ONE CHARACTER EXECUTION OPTION CODE                     XUDBG
1 CD4   ORIGIN - REFERENCE POINT FOR MEMORY ACCESS OFFSET COMPUTATION    XUDBG
1 CD4   ORG    - ADDRESS OF 'ORIGIN'                                     XUDBG
1 CD4   STP    - 'ORIGIN' RELATIVE END OF DATA TO BE SNAPPED             XUDBG
1 CD4   STRT   - 'ORIGIN' RELATIVE BEGINNING OF MEMORY TO BE ACCESSED    XUDBG
1 CD4   RTN    - RETURN POINT INDICATOR FOR INTERNAL PROCEDURE 'PROMPT AND XUDBG
1 CD4            CONVERSION'                                             XUDBG
1 CD4                                                                    XUDBG
1 C*********                                                             XUDBG
1 CD5      NOTES                                                         XUDBG
1 CD5                                                                    XUDBG
```

```
61    1 CDS   USES EXEC, IAND, XRLOC, XRMOV, XRUPK, XUDPL                    XUDBG
62    1 CDS                                                                 XUDBG
63    1 CDS   WITH THE EXCEPTION OF 'OPTION' AND 'OUTPUT UNIT' ALL INPUTS ARE IN  XUDBG
64    1 CDS   OCTAL.  NO ERROR CHECKING IS PERFORMED THEREFORE CARE SHOULD BE  XUDBG
65    1 CDS   TAKEN TO ASSURE VALID DATA.                                   XUDBG
66    1 CS                                                                  XUDBG
67    1 C*********                                                          XUDBG
```

```
1 BEGIN XUD8G
2   WRITE SNAP HEADER TO TERMINAL
2   DO UNTIL OPTION IS EXIT (E)
3     PROMPT TERMINAL FOR OPTION
3     IF OPTION IS NOT EXIT (E)
3     THEN
4       PERFORM PROMPT AND CONVERSION FOR STARTING ADDRESS
4         IF OPTION IS MODIFY (M)
4         THEN
5           PERFORM PROMPT AND CONVERSION FOR VALUES AND STORE IN MEMORY
5         ELSE
5           PERFORM PROMPT AND CONVERSION FOR ENDING ADDRESS
5           PERFORM PROMPT AND CONVERSION FOR OUTPUT UNIT
5           WRITE SNAP HEADER TO OUTPUT UNIT
6           DO FOR EACH EIGHT WORD BLOCK OF DUMP AREA
6             CALL XUDPL TO FORMAT LINE
6             OUTPUT LINE
5           END DO
4         ENDIF
3       ENDIF
2     END DO
1   END XUD8G
1   BEGIN PROMPT AND CONVERSION
2     ISSUE PROMPT TO TERMINAL, SOLICIT RESPONSE AND WAIT
2     RETRIEVE RESPONSE
2     CLEAR SUM
2     DO FOR EACH CHARACTER
3       SHIFT SUM AND ADD NEXT DIGIT
2     END DO
1   END PROMPT AND CONVERSION
```

```
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
XUD8G
```

```
100   1 CD1   GENERAL FILE DUMP PROGRAM FOR FILE MANAGER FILES                                                    XUDPF
101   1 CD1                                                                                                      XUDPF
102   1 C*********                                                                                               XUDPF
103   1 CD2       INPUT                                                                                          XUDPF
104   1 CD2   NAME    - NAME OF FM FILE TO BE DUMPED                                                             XUDPF
105   1 CD2   IREC    - LOGICAL RECORD NUMBER OF FIRST RECORD TO BE DUMPED                                        XUDPF
106   1 CD2             (FIRST RECORD IS RECORD NUMBER ONE)                                                      XUDPF
107   1 CD2   NREC    - NUMBER OF LOGICAL RECORDS TO DUMP                                                        XUDPF
108   1 CD2   FMT     - RUN TIME FORMAT FOR RECORDS (MAXIMUM OF 72 CHARACTERS) OR                                XUDPF
109   1 CD2             BLANK INDICATING THE DEFAULT OF OCTAL AND ASCII DUMP TYPE                                XUDPF
110   1 CD2             FORMATTING OR THE CHARACTERS UF INDICATING UNFORMATTED                                    XUDPF
111   1 CD2             OUTPUT                                                                                   XUDPF
112   1 CD2   LU      - LOGICAL UNIT NUMBER OF OUTPUT DEVICE                                                     XUDPF
113   1 CD2                                                                                                      XUDPF
114   1 C*********                                                                                               XUDPF
115   1 CD3       OUTPUT                                                                                         XUDPF
116   1 CD3   FORMATTED DUMP OF THE INDICATE PORTION OF THE INDICATED FILE                                       XUDPF
117   1 CD3                                                                                                      XUDPF
118   1 C*********                                                                                               XUDPF
119   1 CD5       NOTES                                                                                          XUDPF
120   1 CD5                                                                                                      XUDPF
121   1 CD5   USES EXEC, MAXO, OPEN, POSNT, READF, RMPAR, XPRBS, XQNOV, XUDPL                                    XUDPF
122   1 CD5                                                                                                      XUDPF
123   1 CD5                                                                                                      XUDPF
124   1 CD5   ANY FILE WITH VARIABLE LENGTH RECORDS WILL BE DUMPED USING A                                       XUDPF
125   1 CD5   RECORD BUFFER OF 1024 WORDS THUS LIMITING THE MAXIMUM DUMPABLE                                     XUDPF
126   1 CD5   RECORD LENGTH.                                                                                     XUDPF
127   1 CD5                                                                                                      XUDPF
128   1 C*********                                                                                               XUDPF
129   1 *                                                                                                        XUDPF
130   1 *                                                                                                        XUDPF
131   1 *                                                                                                        XUDPF
132   1 BEGIN XUDPF                                                                                              XUDPF
133   2    DO FOREVER                                                                                            XUDPF
134   3       READ FILE NAME                                                                                     XUDPF
135   2    EXIT XUDPF IF NAME IS NULL                                                                            XUDPF
136   3       READ INITIAL RECORD NUMBER                                                                         XUDPF
137   3       READ NUMBER OF RECORDS TO DUMP                                                                     XUDPF
138   3       READ DUMP FORMAT                                                                                   XUDPF
139   3       IF FORMAT IS NULL                                                                                  XUDPF
140   3       THEN                                                                                               XUDPF
141   4          SET DEFAULT OCTAL/ASCII FORMAT                                                                  XUDPF
142   3       ENDIF                                                                                              XUDPF
143   3       READ LU OF PRINT DEVICE                                                                            XUDPF
144   3       OPEN FILE                                                                                          XUDPF
145   3       IF SUCCESSFUL                                                                                      XUDPF
146   3       THEN                                                                                               XUDPF
147   4          DO FOR NUMBER OF RECORDS TO DUMP                                                                XUDPF
148   5             READ RECORD                                                                                  XUDPF
149   4          EXIT TO :ERROR: IF FAILED                                                                       XUDPF
150   5             FORMAT AND PRINT RECORD                                                                      XUDPF
151   4          ENDDO                                                                                           XUDPF
152   3       ELSE                                                                                               XUDPF
153   4          :ERROR: OUTPUT MESSAGE                                                                          XUDPF
154   3       ENDIF                                                                                              XUDPF
155   2    ENDDO                                                                                                 XUDPF
156   1 END XUDPF                                                                                                XUDPF
```

```
158   1 CD0      FORTRAN CALLING PROCEDURE                                                    XUDPL
159   1 CD0                                                                                   XUDPL
160   1 CD0   CALL XUDPL (ADDRES, LINE, BUFFER)                                              XUDPL
161   1 CD0                                                                                   XUDPL
162   1 C*********                                                                            XUDPL
163   1 CD1                                                                                   XUDPL
164   1 CD1   PRODUCE AN OCTAL AND ASCII PRINT FORMATTED MEMORY DUMP LINE IMAGE               XUDPL
165   1 CD1                                                                                   XUDPL
166   1 C*********                                                                            XUDPL
167   1 CD2      INPUT                                                                        XUDPL
168   1 CD2   ADDRES - TWO WORD INTEGER ARRAY CONTAINING THE ABSOLUTE AND                     XUDPL
169   1 CD2            RELATIVE ADDRESS TO BE FORMATTED WITH THE LINE                         XUDPL
170   1 CD2   LINE   - EIGHT WORD ARRAY TO BE CONVERTED TO OCTAL AND ASCII                    XUDPL
171   1 CD2            FORMATTED AND SPACED INTO A LINE IMAGE                                 XUDPL
172   1 CD2   BUFFER - FIFTY-ONE WORD BUFFER TO HOLD FORMATTED PRINT LINE.  MUST             XUDPL
173   1 CD2            BE BLANKED PRIOR TO FIRST CALL TO XUDPL AND NOT STORED                 XUDPL
174   1 CD2            INTO BETWEEN CALLS TO XUDPL.                                           XUDPL
175   1 CD2                                                                                   XUDPL
176   1 C*********                                                                            XUDPL
177   1 CD3      OUTPUT                                                                       XUDPL
178   1 CD3   BUFFER - FIFTY-ONE WORD BUFFER CONTAINING FORMATTED LINE                       XUDPL
179   1 CD3   COLUMNS  CONTENTS                                                              XUDPL
180   1 CD3       3-8  FIRST ADDRESS                                                         XUDPL
181   1 CD3     11-16  SECOND ADDRESS                                                        XUDPL
182   1 CD3     21-82  OCTAL REPRESENTATION OF 'LINE'                                       XUDPL
183   1 CD3    87-102  ASCII REPRESENTATION OF 'LINE'                                       XUDPL
184   1 CD3                                                                                 XUDPL
185   1 C*********                                                                          XUDPL
186   1 CD5      NOTES                                                                      XUDPL
187   1 CD5   USES XREXT, XR06, XRSET                                                       XUDPL
188   1 CD5                                                                                 XUDPL
189   1 C*********                                                                          XUDPL
190   1 *                                                                                   XUDPL
191   1 *                                                                                   XUDPL
192   1 *                                                                                   XUDPL
193   1 *                                                                                   XUDPL
194   1 BEGIN XUDPL                                                                         XUDPL
195   2    CALL XR06 TO CONVERT EACH WORD OF ADDRES TO OCTAL                                XUDPL
196   2    DO FOR EACH WORD OF LINE                                                         XUDPL
197   3       CALL XR06 TO CONVERT WORD TO OCTAL                                            XUDPL
198   3       DO FOR EACH BYTE OF WORD                                                      XUDPL
199   4          IF BYTE < 40B OR BYTE > 136B                                               XUDPL
200   4          THEN                                                                       XUDPL
201   5             REPLACE BYTE WITH ASCII PERIOD                                          XUDPL
202   4          ENDIF                                                                      XUDPL
203   3       END DO                                                                        XUDPL
204   2    ENDDO                                                                            XUDPL
205   1 END XUDPL                                                                           XUDPL
```

```
207   1 CD1      FORTRAN MAIN PROGRAM XUFMT IS SCHEDULED BY XUDMP TO PRINT                    XUFMT
208   1 CD1      A PARTITION DUMP WHICH HAS BEEN WRITTEN TO DISK                              XUFMT
209   1 C******                                                                               XUFMT
210   1 CD2      INPUT                                                                        XUFMT
211   1 CD2         ICNTL - CONTROL WORD FOR EXEC CALL READS.  CONTAINS THE                   XUFMT
212   1 CD2                 LU NUMBER OF WHERE DUMP IS ON DISK                               XUFMT
213   1 CD2         TRACK - TRACK NUMBER OF A 4 TRACK GLOBALLY ALLOCATED AREA                XUFMT
214   1 CD2                 CONTAINING THE DUMP                                              XUFMT
215   1 C******                                                                               XUFMT
216   1 CD3      OUTPUT                                                                       XUFMT
217   1 CD3         FORMATTED DUMP TO LU 6                                                    XUFMT
218   1 C******                                                                               XUFMT
219   1 CD5      EXTERNAL REFERENCES                                                          XUFMT
220   1 CD5         EXEC                                                                      XUFMT
221   1 CD5         RMPAR                                                                     XUFMT
222   1 CD5         XRCPR                                                                     XUFMT
223   1 CD5         XRMOV                                                                     XUFMT
224   1 CD5         XUDPL                                                                     XUFMT
225   1 BEGIN XUFMT                                                                           XUFMT
226   2    CALL RMPAR TO GET LU AND STARTING TRACK NOS.                                       XUFMT
227   2    READ 1ST TRACK -- 1ST 12 WORDS ARE HEADER                                          XUFMT
228   2    .COUNT OF ID-SEGS TO BE DUMPED                                                     XUFMT
229   2    .UP TO 7 ID-SEGMENT ADDRESSES                                                      XUFMT
230   2    .LOW AND HIGH BASE PAGE ADDRESSES                                                  XUFMT
231   2    .LOW AND HIGH MAIN MEMORY ADDRESSES                                                XUFMT
232   2    POINT TO 2ND SECTOR OF DUMP DATA                                                   XUFMT
233   2    DO UNTIL ALL ID-SEGS PRINTED                                                       XUFMT
234   3       DO UNTIL 4 EIGHT-WORD LINES PRINTED                                             XUFMT
235   4          PRINT 1 LINE AND INCREMENT POINTER AND ADDRESSES TO NEXT                     XUFMT
236   3       ENDDO                                                                           XUFMT
237   3       INCREMENT TO NEXT SECTOR OF DUMP DATA                                           XUFMT
238   2    ENDDO                                                                              XUFMT
239   2    COMPUTE N, THE NO. OF 8-WORD LINES IN THE BASE PAGE DUMP                           XUFMT
240   2    PERFORM COMPARE AND PRINT FUNCTION                                                 XUFMT
241   2    COMPUTE NTRKS, NO. OF DISK TRACKS OF MAIN MEMORY TO BE READ                        XUFMT
242   2    DO UNTIL NTRKS ARE READ                                                            XUFMT
243   3       READ NEXT TRACK FROM DISK                                                       XUFMT
244   3       COMPUTE N, THE NUMBER OF 8-WORD LINES TO DUMP                                   XUFMT
245   3       PERFORM COMPARE AND PRINT FUNCTION                                              XUFMT
246   2    ENDDO                                                                              XUFMT
247   2    RELEASE THE GLOBALLY ALLOCATED TRACKS                                              XUFMT
248   1 EXIT XUFMT                                                                            XUFMT
249   2    BEGIN COMPARE AND PRINT FUNCTION                                                   XUFMT
250   3       DO UNTIL N LINES PROCESSED                                                      XUFMT
251   4          IF NOT 1ST LINE, THEN                                                        XUFMT
252   5             CALL XRCPR TO COMPARE WITH PREVIOUS LINE                                  XUFMT
253   5             IF LINES ARE IDENTICAL, THEN                                              XUFMT
254   6                IF THIS IS 1ST OF A SERIES, THEN                                       XUFMT
255   7                   WRITE 'DUPLICATE LINE'                                              XUFMT
256   6                ENDIF                                                                  XUFMT
257   5             ELSE                                                                      XUFMT
258   6                CALL XUDPL TO FORMAT THE DUMP LINE                                     XUFMT
259   6                WRITE FORMATTED DUMP LINE                                              XUFMT
260   5             ENDIF                                                                     XUFMT
261   4          ELSE                                                                         XUFMT
262   5             CALL XUDPL TO FORMAT THE DUMP LINE                                        XUFMT
263   5             WRITE FORMATTED DUMP LINE                                                 XUFMT
264   4          ENDIF                                                                        XUFMT
265   3       ENDDO                                                                           XUFMT
```

```
XUFNT
XUFNT
```

```
266     2    END COMPARE AND PRINT FUNCTION
267     1 END XUFNT
```

```
  SYMBOL DEFINITION TABLE
  COMPAR    249
 :ERROR :   153
  PROMPT     91
  XUDBG      69
  XUDPF     132
  XUDPL     194
  XUFMT     225

@XQT F.PDLIST
```

```
1 *D1 ENTRY XVABN                                                                XVABN
1 *D1  TYPE 14 ROUTINE TO ABORT CURRENT ID AND                                   XVABN
1 *D1 REQUEST A DUMP FOR THE ID.                                                 XVABN
1 *D1 ENTRY XUDMP                                                                XVABN
1 *D1  TYPE 14 ROUTINE TO DUMP FROM CURRENT ID                                   XVABN
1 *D1 THE REQUESTED DATA AREAS AND THE CURRENT                                   XVABN
1 *D1 ID SEGMENT AND ITS FATHER CHAIN SEGMENTS                                   XVABN
1 *                                                                              XVABN
1 BEGIN  XVABN   ABEND                                                           XVABN
2 *D2 INPUTS: NONE                                                               XVABN
2 *                                                                              XVABN
2 *D3 OUTPUTS: SETS CURRENT ID TO ABORTED STATE                                   XVABN
2 *                                                                              XVABN
2 *   EXTERNALS: $ABRT,$ABRE,$XEQ,$LIBR                                          XVABN
2    CALL $LIBR     REENTRANT ROUTINE CALL                                       XVABN
2    GET CURRENT ID FROM XEQT (LOCATION 1717)                                    XVABN
2    INCREMENT TO ADDRESS BOUNDRIES                                              XVABN
2    GET ADDRESS BOUNDRIES                                                       XVABN
2    STORE IN XUDMP PARAMTER LIST                                                XVABN
2    PERFORM XUDMP        DUMP THIS PARITION                                     XVABN
2    CALL    $ABRT        ABORT THIS PARTITION                                   XVABN
2    CALL    $ABRE        FREE CURRENT REENTRANT DATA BLOCK                      XVABN
1 EXIT TO :$XEQ       GO TO DISPATCHER                                           XVABN
1 END    XVABN                                                                   XVABN
1 *                                                                              XVABN
1 *                                                                              XVABN
1 *                                                                              XVABN
1 *                                                                              XVABN
1 BEGIN XUDMP                                                                    XVABN
2 *D1 ENTRY XUDMP                                                                XVABN
2 *D2 INPUTS: ADDRESS OF DOUBLE WORD CONTAINING START AND END ADDRESSES          XVABN
2 *D2                 DEFINING AREA TO BE DUMPED                                 XVABN
2 *D3 OUTPUTS: OUTPUTS TO DISK                                                   XVABN
2 *D3            .HEADER DATA CONTAINING NO. OF ID SEGMENTS TO FOLLOW            XVABN
2 *D3            .CURRENT ID SEGMENT AND ALL FATHER CHAIN ID SEGMENTS            XVABN
2 *D3            .MEMORY IMAGE OF BASE PAGE SEGMENT FOR THIS PARTITION           XVABN
2 *D3            .MEMORY IMAGE AS SPECIFIED BY INPUT ADDRESS RANGE               XVABN
2 *D3          SCHEDULES XVFMT TO FORMAT DATA FROM DISK TO PRINT                 XVABN
2 *                                                                              XVABN
2 *D3 EXTERNALS: $LIBR, $LIBX, EXEC, XUFMT                                       XVABN
2    CALL $LIBR TO NOTIFY OF REENTRANCY AND BECOME PRIVILEDGED                   XVABN
2    RUN FATHER ID SEGMENT CHAIN SAVING ADDRESS OF EACH                          XVABN
2    CALL $LIBX TO BECOME NON-PRIVILEDGED                                        XVABN
2    CALL EXEC FOR A GLOBAL ALLOCATION OF DISK                                   XVABN
2    CALL EXEC TO WRITE HEADER DATA USING CLASS I/O                              XVABN
2    DO UNTIL ALL ID SEGMENTS PROCESSED                                          XVABN
3       WRITE ID SEGMENT TO DISK                                                 XVABN
2    ENDDO                                                                       XVABN
2    WRITE BASE PAGE SEGMENT TO REMAINING PROTION OF THIS TRACK                  XVABN
2    DO UNTIL END ADDRESS REACHED                                                XVABN
3       WRITE ONE SECTOR OF DATA                                                 XVABN
2    ENDDO                                                                       XVABN
2    SCHEDULE XUFMT WITH LU AND STARTING TRACK NOS.                              XVABN
2    CALL $LIBX TO RETURN FROM REENTRANT ROUTINE                                 XVABN
1 END XVDMP                                                                      XVABN
```

```
 58    1 *D1 TYPE 14 ROUTINE TO CONTROL COMMUNICATION BETWEEN AND                                                    XVPAW
 59    1 *D1EXECUTION OF FDS MANAGER AND IT'S ASSOCIATED TASKS                                                       XVPAW
 60    1 *D1      (EXECUTIVE, PROCESSORS, AND UTILITIES).                                                            XVPAW
 61    1 *  ENTRY XVPAW AND XVSTB                                                                                   XVPAW
 62    1 *D2 INPUTS                                                                                                 XVPAW
 63    1 *D2   FROM AN ASSOCIATED TASK                                                                              XVPAW
 64    1 *D2      CALL XVPAW(PARMS)                                                                                 XVPAW
 65    1 *D2                                                                                                        XVPAW
 66    1 *D2      ASSEMBLY FORM                                                                                     XVPAW
 67    1 *D2      JSB   XVPAW                                                                                       XVPAW
 68    1 *D2      DEF   *+2        RETURN ADDRESS                                                                   XVPAW
 69    1 *D2      DEF   PARMS      A(PARMS)                                                                         XVPAW
 70    1 *D2 WHERE PARMS ARE P1,P2,P3,P4,P5                                                                         XVPAW
 71    1 *D2      P1 IS THE SERVICE REQUEST                                                                         XVPAW
 72    1 *D2   0= NORMAL TERMINATION (P2-P5 NOT USED)                                                               XVPAW
 73    1 *D2   1= WORK AREA REQUEST (P2-P5 NOT USED)                                                                XVPAW
 74    1 *D2   2= EXECUTE A SEQUENCE TABLE                                                                          XVPAW
 75    1 *D2      (P2-P4 HAS TABLE NAME)                                                                            XVPAW
 76    1 *D2      (P5 INDICATES EXECUTION CONTROL IN CLASS I/O BUFFER)                                              XVPAW
 77    1 *D2   3= RESET SEQUENCE POINTER                                                                            XVPAW
 78    1 *D2      (P2 HAS SEQUENCE NUMBER)(P3-P5 NOT USED)                                                          XVPAW
 79    1 *D2   8= TERMINATE SEQUENCE (P2-P5 NOT USED)                                                               XVPAW
 80    1 *D2   9= TERMINATE FDS FUNCTION (P2-P5 NOT USED)                                                           XVPAW
 81    1 *D2 -32767= ABNORMAL TERMINATION OF ASSOCIATED TASK                                                        XVPAW
 82    1 *D2                                                                                                        XVPAW
 83    1 *D2   FROM AN FDS MANAGER                                                                                  XVPAW
 84    1 *D2      ASSEMBLY FORM                                                                                     XVPAW
 85    1 *D2      JSB   XVPAW                                                                                       XVPAW
 86    1 *D2      DEF   (RETURN POINT)                                                                              XVPAW
 87    1 *D2      OCT   0                                                                                           XVPAW
 88    1 *D2      DEF   PARMS      A(FDS MANAGER RESPONSE)                                                          XVPAW
 89    1 *D2                                                                                                        XVPAW
 90    1 *D3 OUTPUTS                                                                                                XVPAW
 91    1 *D3   TO AN FDS MANAGER                                                                                    XVPAW
 92    1 *D3      REQUEST PARMS (P1-P5) MOVED INTO ASSOCIATED TASK ID-SEGMENT                                       XVPAW
 93    1 *D3      RETURN ADDRESS IS MOVED INTO ID-SEGMENT WORD 9(XSUSP)                                             XVPAW
 94    1 *D3      CURRENT ID-SEGMENT IS MOVED INTO GENERAL WAIT VIA $LIST                                           XVPAW
 95    1 *D3      MANAGER IS ACTIVATED VIA $LIST                                                                    XVPAW
 96    1 *D3   TO AN ASSOCIATED TASK                                                                                XVPAW
 97    1 *D3      RESPONSE PARMS (P1-P5) MOVED INTO ASSOCIATED TASK ID-SEGMENT                                      XVPAW
 98    1 *D5                                                                                                        XVPAW
 99    1 *D5      RETURN ADDRESS IS MOVED INTO ID-SEGMENT WORD 9(XSUSP)                                             XVPAW
100    1 *D5      CURRENT ID-SEGMENT IS MOVED INTO GENERAL WAIT VIA $LIST                                           XVPAW
101    1 *D5      ASSOCIATE TASK IS ACTIVATED VIA $LIST                                                             XVPAW
```

```
103    1 BEGIN XVPAW                                                                  XVPAW
104    2    CALL $LIBR   BECOME PRIVILEGED                                            XVPAW
105    2    SET STOP-ID FROM XEQT(OCT 1717)                                           XVPAW
106    2    IF THIS IS A MANAGER RESPONSE                                             XVPAW
107    2    THEN SET UP TO ACTIVATE ASSOCIATED TASK AND SUSPEND MANAGER               XVPAW
108    3 * CALL SEQUENCE IS RETURN,0,A(PARMS)                                         XVPAW
109    3        PERFORM XVPRAW  POST REQUESTOR AND WAIT                               XVPAW
110    2    ELSE SET UP TO ACTIVATE MANAGER AND SUSPEND ASSOCIATED TASK               XVPAW
111    3 * CALL SEQUENCE IS RETURN,A(PARMS)                                           XVPAW
112    3        PERFORM XVPMAW  POST MANAGER AND WAIT                                 XVPAW
113    2    ENDIF                                                                     XVPAW
114    2 * STOP-ID HAS ID-SEGMENT TO BE SUSPENED,                                     XVPAW
115    2 * AWAKEN-ID HAS ID-SEGMENT TO BE ACTIVATED.                                  XVPAW
116    2    CALL $LIST(SCHEDULE,AWAKEN-ID)                                            XVPAW
117    2    MOVE RETURN ADDRESS TO XSUSP OF STOP-ID.                                  XVPAW
118    2    CALL $LIST(WAIT,STOP-ID)                                                  XVPAW
119    1 EXIT TO :$XEQ  RTE DISPATCHER                                                XVPAW
120    1 END XVPAW                                                                    XVPAW
```

```
122   1 BEGIN XVPMAW                                                                XVPAW
123   2 *  DETERMINE REQUESTORS MANAGER BY USING FATHER ID NUMBER                   XVPAW
124   2 * FIELD IN ID SEGMENTS AS A BACKWARD CHAIN                                  XVPAW
125   2    SET TARGET-ID FROM CURRENT-ID-SEGMENT                                    XVPAW
126   2    DO WHILE FATHER-ID-NUMBER   .NE.  0 OR FATHER IS WAITING                 XVPAW
127   3       COMPUTE FATHER-ID-SEGMENT FROM FATHER-ID-NUMBER IN TARGET-ID          XVPAW
128   3       PERFORM MGRFND(FATHER-ID-SEGMENT,COUNT)                               XVPAW
129   2    EXITIF COUNT .GT.0                                                       XVPAW
130   3       SET TARGET-ID TO FATHER-ID-SEGMENT                                    XVPAW
131   2    ENDDO                                                                    XVPAW
132   2    IF FATHER-ID NUMBER .EQ. 0, OR FATHER NOT WAITING THEN                   XVPAW
133   3       CALL $SYMG (12,*XV03,SEGMENT-NAME)  '*XV03,NAME' REQUESTING PROG      XVPAW
134   3       PERFORM PDUMP                                                         XVPAW
135   2    EXIT TO $LIBX TO ENABLE AND REDISPATCH                                   XVPAW
136   2    ENDIF                                                                    XVPAW
137   2    SET AWAKEN-ID FROM FSD-ENTRY STBMG                                       XVPAW
138   2    GET REQUEST PARMS MOVE INTO ID-SEGMENT                                   XVPAW
139   2    SET STBAT FROM CURRENT-ID                                                XVPAW
140   1 END   XVPMAW                                                                XVPAW
141   1 *                                                                           XVPAW
142   1 *                                                                           XVPAW
143   1 *                                                                           XVPAW
144   1 *                                                                           XVPAW
145   1 BEGIN  XVPRAW                                                               XVPAW
146   2 *  DETERMINE IF CALLER IS A VALID FDS MANAGER                               XVPAW
147   2 *                                                                           XVPAW
148   2    PERFORM MGRFND (CURRENT-ID,COUNT)                                        XVPAW
149   2 *  COUNT WILL BE 0 FOR NO MATCH.                                            XVPAW
150   2 *  COUNT NOT EQUAL ZERO IMPLIES A MATCH                                     XVPAW
151   2 * AND FDS-ENTRY HAS MATCHING FDSTAB ENTRY ADDRESS                           XVPAW
152   2    IF COUNT .LT. 0 THEN  CALLER IS NOT A FDS MANAGER                        XVPAW
153   3       CALL $SYMG (12,*XV01,SEGMENT-NAME)  '*XV01 PROGRAM' REQUESTING PROG.  XVPAW
154   3       PERFORM PDUMP                                                         XVPAW
155   2    EXIT TO :$XEQ THE DISPATCHER                                             XVPAW
156   2    ENDIF                                                                    XVPAW
157   2 *  FDSTAB-ENTRY HAS ENTRY FOR RESPONDING MANAGER                            XVPAW
158   2 *  SET AWAKEN-ID-SEGMENT FROM CURRENT-ASSOCIATED-TASK                       XVPAW
159   2    IF CURRENT-ID NOT WAIT  THEN IT WAS NOT ON WAIT LIST                     XVPAW
160   3       CALL $SYMG (12,*XV02,SEGMENT-NAME) '*XV02 PROGRAM' ASSOCIATED PROG.   XVPAW
161   3       PERFORM PDUMP                                                         XVPAW
162   2    EXIT TO :$XEQ THE DISPATCHER                                             XVPAW
163   2    ENDIF                                                                    XVPAW
164   2    AWAKEN-ID-SEGMENT=STBAT                                                  XVPAW
165   2    IF MANAGER HAS REQUEST FOR ABORT, THEN                                   XVPAW
166   3       CALL $ABRT FOR CURRENT AT                                             XVPAW
167   2    ENDIF                                                                    XVPAW
168   2 *  MOVE FDS MANAGERS INPUT PARMS TO ASSOCIATED TASK ID SEGMENT              XVPAW
169   2    MOVE PARMS TO ID-SEGMENT WORDS 2-6                                       XVPAW
170   1 END  XVPRAW                                                                 XVPAW
```

```
172   1 BEGIN  PDUMP                                                          XVPAW
173   2 *                                                                     XVPAW
174   2 *                                                                     XVPAW
175   2    SET A CALL TO XVABN AT CALLER'S SUSPEND POINT                      XVPAW
176   2 *                                                                     XVPAW
177   1 END    PDUMP                                                          XVPAW
178   1 *                                                                     XVPAW
179   1 *                                                                     XVPAW
180   1 *                                                                     XVPAW
181   1 *                                                                     XVPAW
182   1 BEGIN  MGRFND  INPUT IS TARGET-ID                                     XVPAW
183   2    GET XVSTB    TABLE OF ACTIVE FDS MANAGERS                          XVPAW
184   2    SET COUNT TO NUMBER IN TABLE                                       XVPAW
185   2    SET FDSTAB-ENTRY TO FIRST-ENTRY(STBES)                             XVPAW
186   2    DO WHILE COUNT .GT. 0                                              XVPAW
187   2    EXIT IF  TARGET-ID .EQ. FDS-MANAGER-ID-SEGMENT                     XVPAW
188   3       SET FDSTAB-ENTRY TO FDSTAB-ENTRY + STBEL                        XVPAW
189   3       SET COUNT = COUNT - 1                                           XVPAW
190   2    ENDDO                                                              XVPAW
191   2 *  FDSTAB-ENTRY HAS FOUND MANAGER ENTRY IF COUNT .GT. 0               XVPAW
192   2 *  COUNT = 0 MEANS NO MATCH                                           XVPAW
193   1 END    MGRFND                                                         XVPAW
194   1 *                                                                     XVPAW
195   1 *                                                                     XVPAW
196   1 *                                                                     XVPAW
197   1 *                                                                     XVPAW
198   1 BEGIN  XVSTB                                                          XVPAW
199   2    N        EQU   3     EQUATE FOR NUMBER OF ENTRIES IN XVSTB         XVPAW
200   2    STBNM    DEF   N     NUMBER-IN-TABLE                               XVPAW
201   2    STBAC    DEF   0     NUMBER-ACTIVE                                 XVPAW
202   2    STBRN    DEF   0   STB RESOURCE NUMBER                            XVPAW
203   2    STBES    EQU   *               ENTRY START                         XVPAW
204   2    STBLU    DEF   0     LU(OCTAL)                                     XVPAW
205   2    STBLA    DEF   0     LU(ASCII)                                     XVPAW
206   2    STBMG    DEF   0     A(FDS-MANAGER1-ID-SEGMENT)                    XVPAW
207   2    STBEX    DEF   0     A(FDS-EXECUTIVE1-ID-SEGMENT)                  XVPAW
208   2    STBEC    DEF   0     CLASS NUMBER FOR EXEC                         XVPAW
209   2    STBAT    DEF   0     A(CURRENT-ASSOCIATED-TASK)                    XVPAW
210   2    STBPC    DEF   0     CLASS-NUMBER FOR PROCESSOR                    XVPAW
211   2    STBEQ    DEF   0     EQT SAVE AREA                                 XVPAW
212   2    STBEE    EQU   *               ENTRY END                           XVPAW
213   2    STBEL    EQU   STBEE-STBES  ENTRY LENGTH                           XVPAW
214   2    STBNE    EQU   N               NUMBER-OF-ENTRIES                   XVPAW
215   2    ORG   STBES                                                        XVPAW
216   2    REP   N             DO FOR N ENTRIES                              XVPAW
217   2    REP   STBEL         DEFINE STORAGE WORDS FOR AN ENTRY             XVPAW
218   2    DEF   0                                                           XVPAW
219   1 END    XVSTB                                                          XVPAW
```

```
  SYMBOL DEFINITION TABLE
  MGRFND     182
  PDUMP      172
  XUDMP       30
  XVABN       10
  XVPAW      103
  XVPMAW     122
  XVPRAW     145
  XVSTB      178

@XQT F.PDLIST
```

```
1 CD**************                                                          XXAUT
1 CD0                                                                       XXAUT
1 CD0      FORTRAN CALLING PROCEDURE:                                       XXAUT
1 CD0                                                                       XXAUT
1 CD0            CALL XXAUT                                                 XXAUT
1 CD0                                                                       XXAUT
1 CD**************                                                          XXAUT
1 CD1                                                                       XXAUT
1 CD1      XXAUT HANDLES AUTOMATIC EXECUTION WITHOUT TRACE                  XXAUT
1 CD1                                                                       XXAUT
1 CD**************                                                          XXAUT
1 CD2                                                                       XXAUT
1 CD2      INPUTS IN COMMON:                                                XXAUT
1 CD2                                                                       XXAUT
1 CD2            XE(5)  MASSTA,  XE(10)  SEQSTR, XE(11)  SEQEND, XE(12) SEQPTR,  XXAUT
1 CD2            XE(140)TABEND,  XB(1)   NOPROC, XB(2)   LIBD,              XXAUT
1 CD2            XB(249) SEQNO,  XB(250) SEQLEN, XB(251) SEQTAB             XXAUT
1 CD2                                                                       XXAUT
1 CD**************                                                          XXAUT
1 CD3                                                                       XXAUT
1 CD3      OUTPUTS IN COMMON:                                               XXAUT
1 CD3                                                                       XXAUT
1 CD3            XE(5) MASSTA, XS(13) FLGTAB                                XXAUT
1 CD3                                                                       XXAUT
1 CD**************                                                          XXAUT
1 CD5                                                                       XXAUT
1 CD5      COMMON USED:                                                     XXAUT
1 CD5                                                                       XXAUT
1 CD5            EQUIVALENCE         (XE(5),   MASSTA)                      XXAUT
1 CD5           +(XE(10), SEQSTR),  (XE(11),  SEQEND)                       XXAUT
1 CD5           +(XE(12), SEQPTR),  (XE(140),TABEND),                       XXAUT
1 CD5           +(XB(1),  NOPROC),  (XB(2),   LIBD  ),                      XXAUT
1 CD5           +(XB(235),RESIND),  (XB(249),SEQNO ),                       XXAUT
1 CD5           +(XB(250),SEQLEN),  (XB(251),SEQTAB),                       XXAUT
1 CD5           +(XS(13), FLGTAB)                                           XXAUT
1 CD5                                                                       XXAUT
1 CD5      FDS ROUTINES CALLED:                                             XXAUT
1 CD5                                                                       XXAUT
1 CD5            XRCPR, XREXT, XRMOV, XRMSG,                                XXAUT
1 CD5            XXEXE, XXSTO, XXTMP                                        XXAUT
1 CD5                                                                       XXAUT
1 CD5      RTE ROUTINES CALLED:                                             XXAUT
1 CD5                                                                       XXAUT
1 CD5            IOR                                                        XXAUT
1 CD5                                                                       XXAUT
1 CD**************                                                          XXAUT
```

```
49    1 BEGIN XXAUT                                                                  XXAUT
50    2    IF ENTRY IS FROM A DIRECTIVE THEN                                         XXAUT
51    3       SET MASTER STATE TO INDICATE REENTRY                                   XXAUT
52    3       DO FOR EACH ENTRY IN THIS SEQUENCE TABLE                               XXAUT
53    4          SEARCH LIBRARY DIRECTORY FOR THIS PROCESSOR                         XXAUT
54    4          ERREXIT IF PROCESSOR NOT FOUND TO :ERR1:                            XXAUT
55    4          STUFF INTERFACE TABLE BIT AND VERSION INTO SEQENCE TABLE ENTRY      XXAUT
56    3       ENDDO                                                                  XXAUT
57    3       CALL XXSTO TO STORE REVISED SEQUENCE TABLE IN AWA AS &SEQTB            XXAUT
58    2    ELSE  (I AM BEING REENTERED FROM INT.)                                    XXAUT
59    3       CALL XXTMP TO SET UP TEMPORARY ENTRY WITH &INTAB                       XXAUT
60    3       CALL XXEXE TO EXECUTE FROM TEMPORARY ENTRY                             XXAUT
61    3       IF RESET SEQUENCE NUMBER IS NOT REQUESTED THEN                         XXAUT
62    3       EXIT XXAUT IF TERMINAL ENTRY WAS JUST EXECUTED                         XXAUT
63    4          SET STARTING ENTRY TO NEXT ENTRY                                    XXAUT
64    3       ENDIF                                                                  XXAUT
65    2    ENDIF                                                                     XXAUT
66    2    DO UNTIL TERMINAL ENTRY IS EXECUTED AND IT IS NOT A REQUEST FOR RESET     XXAUT
67    3       CALL XXEXE TO EXECUTE REMAINDER OF TABLE                               XXAUT
68    2    ENDDO                                                                     XXAUT
69    1 EXIT XXAUT                                                                   XXAUT

70    2    :ERR1:                                                                    XXAUT
71    2    CALL XRMSG TO DISPLAY INVALID PROCESSOR NAME                              XXAUT
72    1 END XXAUT                                                                    XXAUT
```

```
1 CD*************                                                    XXCNT
1 CD0                                                                XXCNT
1 CD0      FORTRAN CALLING PROCEDURE FOR EXECUTION CONTROLLER:       XXCNT
1 CD0                                                                XXCNT
1 CD0            CALL XELDS ( XXCNT)                                 XXCNT
1 CD0                                                                XXCNT
1 CD*************                                                    XXCNT
1 CD1                                                                XXCNT
1 CD1      XXCNT IS THE MAIN PROGRAM FOR THE EXECUTION CONTROLLER.   XXCNT
1 CD1      IT GIVES CONTROL TO THE APPROPRIATE SUBROUTINE DEPENDING  XXCNT
1 CD1      ON THE MODE AND RETURNS TO DIRECTIVE LEVEL.               XXCNT
1 CD1                                                                XXCNT
1 CD*************                                                    XXCNT
1 CD2                                                                XXCNT
1 CD2      INPUTS FROM CALLING SEQUENCE:                             XXCNT
1 CD2                                                                XXCNT
1 CD2            XXCNT - (INTEGER, 3 WORDS) ARRAY CONTAINING THE     XXCNT
1 CD2                    NAME "XXCNT " USED BY XELDS TO CALL EXEC    XXCNT
1 CD2                    TO LOAD THE EXECUTION CONTROLLER SEGMENT.   XXCNT
1 CD2                                                                XXCNT
1 CD*************                                                    XXCNT
1 CD4                                                                XXCNT
1 CD4      INTERNAL VARIABLES:                                       XXCNT
1 CD4                                                                XXCNT
1 CD4            MODE - (INTEGER, 1 WORD) MODE IN WHICH THE EXECUTION XXCNT
1 CD4                   CONTROLLER WAS CALLED                        XXCNT
1 CD4                           0  -  MANUAL                         XXCNT
1 CD4                           1  -  SEMI AUTOMATIC                 XXCNT
1 CD4                           2  -  AUTOMATIC WITH TRACE           XXCNT
1 CD4                           3  -  AUTOMATIC                      XXCNT
1 CD4                                                                XXCNT
1 CD*************                                                    XXCNT
1 CD5                                                                XXCNT
1 CD5      COMMON USED:                                              XXCNT
1 CD5                                                                XXCNT
1 CD5            EQUIVALENCE   (XE(5),  MASSTA)                      XXCNT
1 CD5                                                                XXCNT
1 CD5      FDS ROUTINES USED                                         XXCNT
1 CD5                                                                XXCNT
1 CD5              XERTN, XKEXT, XRMSG, XXMAN                        XXCNT
1 CD5                                                                XXCNT
1 CD5      NOTE: CONTAINS DUMMY CALL TO XEXEC                        XXCNT
1 CD5                                                                XXCNT
1 CD*************                                                    XXCNT
```

```
119    1 BEGIN XXCNT                                                          XXCNT
120    2    SET MODE TO XREXT OF BITS 12 AND 13 OF MASSTA                     XXCNT
121    2    CASE MODE (:MANU:, :SEMI:, :AUTT:, :AUTO:)                        XXCNT

122    3       :MANU: CALL XXMAN                                              XXCNT

123    3       :SEMI: CALL XXSEM                                              XXCNT

124    3       :AUTT: CALL XRMSG TO DISPLAY INVALID OPTION                    XXCNT

125    3       :AUTO: CALL XXAUT                                              XXCNT
126    2    ENDCASE                                                           XXCNT
127    2    SET MASTER STATE TO DIRECTIVE LEVEL                               XXCNT
128    2    CALL XERTN TO RETURN TO XEXEC **NO RETURN**                       XXCNT
129    2    DUMMY CALL XEXEC                                                  XXCNT
130    1 END XXCNT                                                            XXCNT
```

```
132   1 CD**************                                                                  XXDEC
133   1 CD0                                                                             XXDEC
134   1 CD0       FORTRAN CALLING SEQUENCE:                                             XXDEC
135   1 CD0                                                                             XXDEC
136   1 CD0             CALL XXDEC (RETC)                                               XXDEC
137   1 CD0                                                                             XXDEC
138   1 CD**************                                                                  XXDEC
139   1 CD1                                                                             XXDEC
140   1 CD1       XXDEC DECODES A RESPONSE OF PROCESSOR NAME (,INT TABLE NAME)          XXDEC
141   1 CD1       INTO A SEQUENCE TABLE ENTRY.                                          XXDEC
142   1 CD1                                                                             XXDEC
143   1 CD**************                                                                  XXDEC
144   1 CD2                                                                             XXDEC
145   1 CD2       INPUTS IN COMMON:                                                     XXDEC
146   1 CD2                                                                             XXDEC
147   1 CD2             XE(85) TOKENS,  XE(145) COMBUF, XB(1) NOPROC, XB(2) LIBD        XXDEC
148   1 CD2                                                                             XXDEC
149   1 CD**************                                                                  XXDEC
150   1 CD3                                                                             XXDEC
151   1 CD3       OUTPUTS IN CALLING SEQUENCE:                                          XXDEC
152   1 CD3                                                                             XXDEC
153   1 CD3             RETC - RETURN CODE (0 IS NORMAL RETURN)                         XXDEC
154   1 CD3                                                                             XXDEC
155   1 CD3       OUTPUTS IN COMMON:                                                    XXDEC
156   1 CD3                                                                             XXDEC
157   1 CD3             XE(16) PRCNAM,  XB(251) SEQTAB                                  XXDEC
158   1 CD3                                                                             XXDEC
159   1 CD**************                                                                  XXDEC
160   1 CD5                                                                             XXDEC
161   1 CD5       COMMON USED:                                                          XXDEC
162   1 CD5                                                                             XXDEC
163   1 CD5             EQUIVALENCE          (XE(16), PRCNAM),                          XXDEC
164   1 CD5           + (XE(85),  TOKENS),  (XE(144),TOKPTR),                         XXDEC
165   1 CD5           + (XE(145), COMBUF),                                            XXDEC
166   1 CD5           + (XB(1),   NOPROC),  (XB(2),  LIBD  ),                         XXDEC
167   1 CD5           + (XB(251), SEQTAB)                                             XXDEC
168   1 CD5                                                                           XXDEC
169   1 CD5       FDS ROUTINES USED:                                                  XXDEC
170   1 CD5                                                                           XXDEC
171   1 CD5             XRCPR, XREXT, XRMOV, XRMSG                                    XXDEC
172   1 CD5                                                                           XXDEC
173   1 CD5       RTE ROUTINES USED:                                                  XXDEC
174   1 CD5                                                                           XXDEC
175   1 CD5             IAND                                                          XXDEC
176   1 CD5                                                                           XXDEC
177   1 CD**************                                                              XXDEC
```

```
179    1 BEGIN XXDEC                                                                    XXDEC
180    2    INITIALIZE RETURN CODE TO ZERO                                              XXDEC
181    2    SET SEQUENCE ENTRY TO ZEROS                                                 XXDEC
182    2    ERREXIT IF FIRST TOKEN IS NOT A PROCESSOR NAME TO :ERR1:                    XXDEC
183    2    SEARCH LIBRARY DIRECTORY FOR PROCESSOR                                      XXDEC
184    2    ERREXIT IF NAME NOT FOUND TO :ERR1:                                         XXDEC
185    2    MOVE PROCESSOR NAME, IT BIT AND VERSION INTO SEQUENCE ENTRY                 XXDEC
186    2    IF AN INTERFACE TABLE NAME WAS ENTERED THEN                                 XXDEC
187    3       MOVE INTERFACE TABLE NAME INTO SEQUENCE ENTRY                            XXDEC
188    2    ENDIF                                                                       XXDEC
189    2    ERREXIT IF LAST TOKEN IS NOT EOS TO :ERR1:                                  XXDEC
190    2    ERREXIT IF INTERFACE TABLE IS SPECIFIED WHEN NOT NEEDED TO :ERR1:           XXDEC
191    2    IF AN INTERFACE TABLE IS REQUIRED BUT NOT SPECIFIED THEN                    XXDEC
192    3       SET INTERFACE TABLE IN SEQUENCE ENTRY TO '&INTAB'                        XXDEC
193    2    ENDIF                                                                       XXDEC
194    1 EXIT XXDEC                                                                     XXDEC

195    2    :ERR1:                                                                      XXDEC
196    2    CALL XRNSG TO DISPLAY ERROR                                                 XXDEC
197    2    SET RETURN CODE TO SAY ERROR                                                XXDEC
198    1 END XXDEC                                                                      XXDEC
```

```
200   1 CD**************                                                                  XXDEF
201   1 CD0                                                                              XXDEF
202   1 CD0    FORTRAN CALLING PROCEDURE:                                                XXDEF
203   1 CD0                                                                              XXDEF
204   1 CD0        CALL XXDEF                                                            XXDEF
205   1 CD0                                                                              XXDEF
206   1 CD**************                                                                  XXDEF
207   1 CD1                                                                              XXDEF
208   1 CD1    XXDEF READS IN THE DEFAULT INTERFACE TABLE FOR A PROCESSOR                XXDEF
209   1 CD1    AND STORES IT IN THE AWA AS &INTAB                                        XXDEF
210   1 CD1                                                                              XXDEF
211   1 CD**************                                                                  XXDEF
212   1 CD2                                                                              XXDEF
213   1 CD2    INPUTS FROM COMMON:                                                       XXDEF
214   1 CD2                                                                              XXDEF
215   1 CD2            PRCNAM - (INTEGER, 3 WORDS) NAME OF PROCESSOR IN SEQTAB           XXDEF
216   1 CD2                     FOR WHICH NO INTERFACE TABLE WAS SUPPLIED                XXDEF
217   1 CD2                                                                              XXDEF
218   1 CD**************                                                                  XXDEF
219   1 CD4                                                                              XXDEF
220   1 CD4    INTERNAL VARIABLES:                                                       XXDEF
221   1 CD4                                                                              XXDEF
222   1 CD4            DEFTAB - (INTEGER, 1200 WORDS) ARRAY WHERE MAXIMUM SIZE           XXDEF
223   1 CD4                     DEFAULT INTERFACE TABLE CAN BE READ INTO                 XXDEF
224   1 CD4            DEFNAM - (INTEGER, 3 WORDS) ARRAY WHERE INTERFACE TABLE          XXDEF
225   1 CD4                     NAME IS CREATED FROM PROCESSOR NAME                      XXDEF
226   1 CD4                                                                              XXDEF
227   1 CD**************                                                                  XXDEF
228   1 CD5                                                                              XXDEF
229   1 CD5    COMMON USED:                                                              XXDEF
230   1 CD5                                                                              XXDEF
231   1 CD5             EQUIVALENCE  (XE(5),  MASSTA),  (XE(6),  SUBSTA),                XXDEF
232   1 CD5            +             (XE(13), INTNAM),  (XE(16), PRCNAM),                XXDEF
233   1 CD5            +             (XE(19), REQPTR),  (XE(20), REQBUF),                XXDEF
234   1 CD5            +             (XE(142),ICR   ),                                   XXDEF
235   1 CD5            +             (XS(6),  TMPTAB),  (XS(14), DEFNAM),                XXDEF
236   1 CD5            +             (XS(18), LEN1  ),  (XS(19), LEN2  ),                XXDEF
237   1 CD5            +             (XS(20), IDCB  ),  (XSS(200),RETC )                 XXDEF
238   1 CD5                                                                              XXDEF
239   1 CD5    RTE ROUTINES USED:                                                        XXDEF
240   1 CD5                                                                              XXDEF
241   1 CD5            CLOSE, EXEC, KCVT, OPEN, READF                                    XXDEF
242   1 CD5                                                                              XXDEF
243   1 CD5    FDS ROUTINES USED:                                                        XXDEF
244   1 CD5                                                                              XXDEF
245   1 CD5            XERTN, XREQ, XREXT, XRMOV, XRMSG, XRPCK, XRUPK                    XXDEF
246   1 CD5                                                                              XXDEF
247   1 CD**************                                                                  XXDEF
```

```
249   1 BEGIN XXDEF                                                                    XXDEF
250   2    CREATE DEFAULT INTERFACE TABLE NAME FROM PROCESSOR NAME                     XXDEF
251   2    CALL OPEN TO OPEN FILE                                                      XXDEF
252   2    ERREXIT IF OPEN ERROR TO :FILERR:                                           XXDEF
253   2    CALL READF TO READ RECORD 1                                                 XXDEF
254   2    ERREXIT IF READ  ERROR TO :FILERR:                                          XXDEF
255   2    IF THERE IS LITERAL DATA THEN                                               XXDEF
256   3       CALL READF TO READ RECORD 2                                              XXDEF
257   3       ERREXIT IF READ ERROR TO :FILERR:                                        XXDEF
258   2    ENDIF                                                                       XXDEF
259   2    CALL CLOSE TO CLOSE FILE                                                    XXDEF
260   2    ERREXIT IF CLOSE ERROR TO :FILERR:                                          XXDEF
261   2    CALL XREQ TO ALLOCATE & STORE &INTAB                                        XXDEF
262   2    ERREXIT IF RETURN CODE IS NOT ZERO TO :MGRERR:                              XXDEF
263   2    IF DEFAULT TABLE IS INCOMPLETE THEN                                         XXDEF
264   3       SET INTERFACE TABLE NAME IN XE TO &INTAB                                 XXDEF
265   3       SET SUBSTATE TO INTERFACE TABLE EDITOR                                   XXDEF
266   3       CALL XERTN TO IMPLICITLY CALL INT EDITOR  **NO RETURN**                  XXDEF
267   2    ENDIF                                                                       XXDEF
268   1 EXIT XXDEF                                                                     XXDEF

269   2    :FILERR:                                                                    XXDEF
270   2    CALL CLOSE TO CLOSE FILE                                                    XXDEF
271   2    CALL XRMSG TO DISPLAY FILE ACCESS ERROR                                     XXDEF
272   1 EXIT XXDEF                                                                     XXDEF

273   2    :MGRERR:                                                                    XXDEF
274   2    CALL XRMSG TO DISPLAY SPACE ERROR                                           XXDEF
275   2    CALL EXEC TO FREE CLASS NUMBER                                              XXDEF
276   2    SET MASTER STATE TO DIRECTIVE LEVEL                                         XXDEF
277   2    CALL XERTN TO RETURN TO EXEC **NO RETURN**                                  XXDEF
278   1 END XXDEF                                                                      XXDEF
```

```
C**********                                                                    XXEXE
C                                                                              XXEXE
C     FORTRAN CALLING PROCEDURE:                                               XXEXE
C                                                                              XXEXE
C          CALL XXEXE                                                          XXEXE
C                                                                              XXEXE
C**********                                                                    XXEXE
C                                                                              XXEXE
C     XXEXE REQUESTS THE MANAGER TO EXECUTE A SEQUENCE AND                     XXEXE
C     HANDLES ALL RETURN CODES FROM THE MANAGER                                XXEXE
C                                                                              XXEXE
C**********                                                                    XXEXE
C                                                                              XXEXE
C     INPUTS FROM COMMON:                                                      XXEXE
C                                                                              XXEXE
C          XE(10) SEQSTR, XE(11) SEQEND,  XE(12) SEQPTR,                       XXEXE
C          XB(250)SEQLEN, XB(251)SEQTAB,  XS(13) FLGTAB                        XXEXE
C                                                                              XXEXE
C**********                                                                    XXEXE
C                                                                              XXEXE
C     OUTPUTS TO COMMON:                                                       XXEXE
C                                                                              XXEXE
C          XE(6)  SUBSTA, XE(10) SEQSTR,  XE(12) SEQPTR,                       XXEXE
C          XE(13) INTNAM, XE(16) PRCNAM,  XB(235)RESIND                        XXEXE
C                                                                              XXEXE
C**********                                                                    XXEXE
C                                                                              XXEXE
C     COMMON USED:                                                             XXEXE
      EQUIVALENCE  (XE(5),   MASSTA),  (XE(6),   SUBSTA),                      XXEXE
     +             (XE(10),  SEQSTR),  (XE(11),  SEQEND),                      XXEXE
     +             (XE(12),  SEQPTR),  (XE(13),  INTNAM),                      XXEXE
     +             (XE(16),  PRCNAM),  (XE(20),  REQBUF),                      XXEXE
     +             (XB(235), RESIND),  (XB(249), SEQNO ),                      XXEXE
     +             (XB(250), SEQLEN),  (XB(251), SEQTAB),                      XXEXE
     +             (XS(1),   IPARM ),  (XS(13),  FLGTAB)                       XXEXE
C                                                                              XXEXE
C     FDS ROUTINES USED:                                                       XXEXE
C                                                                              XXEXE
C          XREQ,  XREXT, XRI6,  XRMOV, XRMSG,                                  XXEXE
C          XRSET, XVPAM, XXDEF, XXTMP                                          XXEXE
C                                                                              XXEXE
C     RTE ROUTINES USED:                                                       XXEXE
C                                                                              XXEXE
C          RMPAR                                                               XXEXE
C                                                                              XXEXE
C**********                                                                    XXEXE
```

```
327   1 BEGIN XXEXE                                                                                                    XXEXE
328   2    INITIALIZE RESET INDEX TO ZERO                                                                              XXEXE
329   2    IF TABLE FLAG SAYS SEQTAB IN XB THEN                                                                        XXEXE
330   3       CALL XVPAW WITH SEQ# START AND END                                                                       XXEXE
331   2    ELSE - AM  BEING CALLED TO EXECUTE A TEMPORARY SEQUENCE TABLE                                               XXEXE
332   3       CALL XVPAW WITH SEQUENCE POINTER AS START AND END                                                        XXEXE
333   2    ENDIF                                                                                                       XXEXE
334   2    CALL RMPAR TO RETRIEVE RETURN PARAMETERS                                                                    XXEXE
335   2    IF THE MANAGER DETECTED AN ERROR TRYING TO EXECUTE THE SEQUENCE THEN                                        XXEXE
336   3       SET SEQUENCE POINTER TO SEQUENCE # IN ERROR                                                              XXEXE
337   3       FIND ENTRY IN ERROR AND SAVE IN XE                                                                       XXEXE
338   3       CASE ERROR (:INT1:,:ERR1:,:INT2:,:ERR2:,:ERR3:,:RESET:,:ERR4:,:ERR5:,:ERR6:)                             XXEXE

339   4          :INT1: INTERFACE TABLE NOT SPECIFIED                                                                  XXEXE
340   4          CALL XXDEF TO READ UP DEFAULT TABLE                                                                   XXEXE
341   4          CALL XXTMP TO SET UP A TEMPORARY ENTRY                                                                XXEXE
342   4          PERFORM XXEXE TO EXECUTE  **NO RETURN**                                                               XXEXE

343   4          :ERR1: SET MESSAGE TO INTERFACE TABLE NOT FOUND                                                       XXEXE

344   4          :INT2: INTERFACE TABLE NOT COMPLETE                                                                   XXEXE
345   4          SET SUBSTATE TO INTERFACE TABLE EDITOR                                                                XXEXE
346   4          CALL XERTN TO RETURN TO EXEC  **NO RETURN**                                                           XXEXE

347   4          :ERR2: SET MESSAGE TO INT TABLE NOT FOR PROCESSOR                                                     XXEXE

348   4          :ERR3: SET MESSAGE TO VERSIONS DO NOT MATCH                                                           XXEXE

349   4          :RESET: RESET REQUESTED ON TERMINATION                                                                XXEXE
350   4          CONVERT SEQUENCE # TO INDEX                                                                           XXEXE
351   4          SAVE INDEX AND SEQUENCE NUMBER IN COMMON                                                              XXEXE
352   3       EXIT XXEXE                                                                                               XXEXE

353   4          :ERR4: SET MESSAGE TO RESET SEQ# NOT FOUND                                                            XXEXE

354   4          :ERR5:  SET MESSAGE TO PROCESSOR ABENDED                                                              XXEXE

355   4          :ERR6:  SET MESSAGE TO AWA OVERFLOW                                                                   XXEXE
356   3       ENDCASE                                                                                                  XXEXE
357   3       CALL XRMSG TO DISPLAY ERROR                                                                              XXEXE
358   3       FORMAT SEQUENCE ENTRY INTO ASKII                                                                         XXEXE
359   3       CALL XRMSG TO DISPLAY SEQUENCE TABLE ENTRY IN ERROR                                                      XXEXE
360   3       IF MODE IS SEMI-AUTO AND ENTRY IS NOT AN OVERRIDE THEN                                                   XXEXE
361   4          RESET OLD INDEX TO RE-EXECUTE THIS ENTRY                                                              XXEXE
362   3       ENDIF                                                                                                    XXEXE
363   3       IF MODE IS AUTO THEN                                                                                     XXEXE
364   4          SET MASTER STATE TO ZERO                                                                              XXEXE
365   4          CALL XERTN TO ABORT SEQUENCE  *** NO RETURN ***                                                       XXEXE
366   3       ENDIF                                                                                                    XXEXE
367   2    ENDIF                                                                                                       XXEXE
368   2    RETURN                                                                                                      XXEXE
369   1 END XXEXE                                                                                                      XXEXE
```

```
371   1 CD************                                                        XXMAN
372   1 CD0                                                                  XXMAN
373   1 CD0       PROCEDURE FOR CALLING XXMAN:                               XXMAN
374   1 CD0                                                                  XXMAN
375   1 CD0             CALL XXMAN                                           XXMAN
376   1 CD0                                                                  XXMAN
377   1 CD************                                                        XXMAN
378   1 CD1                                                                  XXMAN
379   1 CD1       XXMAN IS THE MANUAL EXECUTION CONTROLLER                   XXMAN
380   1 CD1                                                                  XXMAN
381   1 CD************                                                        XXMAN
382   1 CD2                                                                  XXMAN
383   1 CD2       INPUTS FROM COMMON:                                        XXMAN
384   1 CD2                                                                  XXMAN
385   1 CD2             XE(5) MASSTA,  XB(249) SEQNO                         XXMAN
386   1 CD2                                                                  XXMAN
387   1 CD************                                                        XXMAN
388   1 CD3                                                                  XXMAN
389   1 CD3       OUTPUTS TO COMMON:                                         XXMAN
390   1 CD3                                                                  XXMAN
391   1 CD3             XE(10) SEQSTR, XE(11) SEQEND, XB(249)SEQNO,          XXMAN
392   1 CD3             XB(250)SEQLEN, XB(251)SEQTAB, XS(13) FLGTAB          XXMAN
393   1 CD3                                                                  XXMAN
394   1 CD************                                                        XXMAN
395   1 CD5                                                                  XXMAN
396   1 CD5       COMMON USED:                                               XXMAN
397   1 CD5                                                                  XXMAN
398   1 CD5             EQUIVALENCE (XE(5),   MASSTA), (XE(10),  SEQSTR),    XXMAN
399   1 CD5            +            (XE(11),  SEQEND),                       XXMAN
400   1 CD5            +            (XB(249), SEQNO ), (XB(250), SEQLEN),    XXMAN
401   1 CD5            +            (XB(251), SEQTAB), (XS(13),  FLGTAB)     XXMAN
402   1 CD5                                                                  XXMAN
403   1 CD5       FDS ROUTINES USED:                                         XXMAN
404   1 CD5                                                                  XXMAN
405   1 CD5             XREXT, XRMSG, XTCOM, XXEXE,                          XXMAN
406   1 CD5             XXDEC, XXDEF, XXSTO, XXTMP                           XXMAN
407   1 CD5                                                                  XXMAN
408   1 CD5       RTE ROUTINES USED:                                         XXMAN
409   1 CD5                                                                  XXMAN
410   1 CD5             IOR                                                  XXMAN
411   1 CD5                                                                  XXMAN
412   1 CD************                                                        XXMAN
```

```
414    1 BEGIN XXMAN                                                                    XXMAN
415    2    IF ENTRY IS FROM A DIRECTIVE THEN                                           XXMAN
416    3       SET MASSTA TO INDICATE RE-ENTRY                                          XXMAN
417    3       DO UNTIL PERCENT IS ENTERED                                              XXMAN

418    4          :PROMPT: CALL XTCOM TO PROMPT FOR PRNAME,*ITNAME                      XXMAN
419    4          IF PERCENT IS NOT ENTERED THEN                                        XXMAN
420    5             ERREXIT IF CR ENTERED TO :PROMPT:                                  XXMAN
421    5             CALL XXDEC TO DECODE RESPONSE                                      XXMAN
422    5             ERREXIT IF INVALID RESPONSE TO :PROMPT:                            XXMAN
423    5             SET SEQ #S IN XE .) ZEROS                                          XXMAN
424    5             SET #ENTRIES IN SEQTAC IN XE TO 1                                  XXMAN
425    5             CALL XXSTO TO STORE SEQUENCE TABLE                                 XXMAN
426    5             IF IT NAME IN SEQTAB IS &INTAB THEN                                XXMAN
427    6                PROCESSOR REQUIRES AN IT THEN                                   XXMAN
428    6                CALL XXDEF TO READ UP DEFAULT INTERFACE TABLE                   XXMAN
429    5             ENDIF                                                              XXMAN
430    5             CALL XXEXE TO EXECUTE SEQTAB                                       XXMAN
431    4          ENDIF                                                                 XXMAN
432    3       ENDDO                                                                    XXMAN
433    2    ELSE  -  AM BEING REENTERED FROM INTERFACE TABLE EDITOR                     XXMAN
434    3       CALL XXTMP TO SET UP TO EXECUTE A TEMPORARY TABLE                        XXMAN
435    3       CALL XXEXE TO EXECUTE ENTRY                                              XXMAN
436    3       PERFORM XXMAN  **NO RETURN**                                             XXMAN
437    2    ENDIF                                                                       XXMAN
438    2    RETURN                                                                      XXMAN

439    2    :PRMERR: CALL XRMSG TO DISPLAY ERROR                                        XXMAN
440    2    PERFORM XXMAN  **NO RETURN**                                                XXMAN
441    1 END XXMAN                                                                      XXMAN
```

```
443   1 CD**************                                                                                  XXSEM
444   1 CD0                                                                                               XXSEM
445   1 CD0       FORTRAN CALLING SEQUENCE:                                                               XXSEM
446   1 CD0                                                                                               XXSEM
447   1 CD0            CALL XXSEM                                                                         XXSEM
448   1 CD0                                                                                               XXSEM
449   1 CD**************                                                                                  XXSEM
450   1 CD1                                                                                               XXSEM
451   1 CD1       XXSEM HANDLES SEMI AUTOMATIC EXECUTION                                                  XXSEM
452   1 CD1                                                                                               XXSEM
453   1 CD**************                                                                                  XXSEM
454   1 CD2                                                                                               XXSEM
455   1 CD2       INPUTS IN COMMON:                                                                       XXSEM
456   1 CD2                                                                                               XXSEM
457   1 CD2            XE(5)  MASSTA, XE(10) SEQSTR, XE(11) SEQEND, XE(12) SEQPTR,                        XXSEM
458   1 CD2            XE(140)TABEND, XB(1)  NOPROC, XB(2)  LIBD,                                         XXSEM
459   1 CD2            XB(249) SEQNO, XB(250)SEQLEN, XB(251)SEQTAB                                        XXSEM
460   1 CD2                                                                                               XXSEM
461   1 CD**************                                                                                  XXSEM
462   1 CD3                                                                                               XXSEM
463   1 CD3       OUTPUTS IN COMMON:                                                                      XXSEM
464   1 CD3                                                                                               XXSEM
465   1 CD3            XE(5) MASSTA, XS(13) FLGTAB                                                        XXSEM
466   1 CD3                                                                                               XXSEM
467   1 CD4**************                                                                                XXSEM
468   1 CD4                                                                                               XXSEM
469   1 CD4       INTERNAL COMMON USED:                                                                   XXSEM
470   1 CD4                                                                                               XXSEM
471   1 CD4            XE(139) EXEND  - ENDING SEQUENCE NUMBER USED TO                                    XXSEM
472   1 CD4                             TERMINATE SEQUENCE                                                XXSEM
473   1 CD4            XE(141) CURIND - INDEX TO THE CURRENT ENTRY BEING EXECUTED                         XXSEM
474   1 CD4            XB(235) RESIND - INDEX TO RESET CURRENT INDEX TO                                   XXSEM
475   1 CD4            XB(236) ASCENT - ASKII SEQUENCE TABLE ENTRY USED TO PROMPT                         XXSEM
476   1 CD4                             THE USER                                                          XXSEM
477   1 CD4            XB(246) OLDIND - INDEX TO THE LAST ENTRY EXECUTED IN THE                           XXSEM
478   1 CD4                             SEQUENCE TABLE                                                    XXSEM
479   1 CD4                                                                                               XXSEM
480   1 CD**************                                                                                  XXSEM
481   1 CD5                                                                                               XXSEM
482   1 CD5       COMMON USED:                                                                            XXSEM
483   1 CD5                                                                                               XXSEM
484   1 CD5            EQUIVALENCE          (XE(5),   MASSTA),                                            XXSEM
485   1 CD5           + (XE(10),  SEQSTR),  (XE(11),  SEQEND),                                           XXSEM
486   1 CD5           + (XE(12),  SEQPTR),  (XE(85),  EOS   ),                                           XXSEM
487   1 CD5           + (XE(86),  INTGR ),  (XE(100), AMPER ),                                           XXSEM
488   1 CD5           + (XE(139), EXEND ),  (XE(140), TABEND),                                           XXSEM
489   1 CD5           + (XE(141), CURIND),  (XE(145), COMBUF),                                           XXSEM
490   1 CD5           + (XB(1),   NOPROC),  (XB(2),   LIBD  ),                                           XXSEM
491   1 CD5           + (XB(235), RESIND),                                                               XXSEM
492   1 CD5           + (XB(236), ASCENT),  (XB(246), OLDIND),                                           XXSEM
493   1 CD5           + (XB(249), SEQNO ),  (XB(250), SEQLEN),                                           XXSEM
494   1 CD5           + (XB(251), SEQTAB),                                                               XXSEM
495   1 CD5           + (XS(13),  FLGTAB)                                                                XXSEM
496   1 CD5                                                                                              XXSEM
497   1 CD5       FDS ROUTINES USED:                                                                     XXSEM
498   1 CD5                                                                                              XXSEM
499   1 CD5            XRCPR, XREXT, XRIO,  XRMOV,                                                       XXSEM
500   1 CD5            XRMSG, XRSET, XXDEC, XXDEF, XXEXE,                                                XXSEM
501   1 CD5            XXSTO, XXTMP, XYCOM                                                               XXSEM
```

```
502    1 CD5
503    1 CD5       RTE ROUTINES USED:                                    XXSEM
504    1 CD5                                                             XXSEM
505    1 CD5             EXEC, IOR                                       XXSEM
506    1 CD5                                                             XXSEM
507    1 CD**************                                                XXSEM
```

```
509   1 BEGIN XXSEM                                                                                                    XXSEM
510   2    IF ENTRY IS FROM A DIRECTIVE THEN                                                                           XXSEM
511   3       SET MASTER STATE TO SAY REENTRY                                                                          XXSEM
512   3       DO FOR # ENTRIES IN SEQUENCE TABLE                                                                       XXSEM
513   4          SEARCH LIBD FOR PROCESSOR NAME                                                                        XXSEM
514   4          ERREXIT IF NOT FOUND TO :ERR1:                                                                        XXSEM
515   4          MOVE IT BIT AND VERSION INTO SEQUENCE TABLE ENTRY                                                     XXSEM
516   3       ENDDO                                                                                                    XXSEM
517   3       SAVE ORIGINAL ENDING SEQUENCE #                                                                          XXSEM
518   3       SET CURRENT SEQUENCE # TO BEGINNING SEQUENCE #                                                           XXSEM
519   3       SET FLGTAB TO SAY EXECUTE ENTIRE SEQTAB                                                                  XXSEM
520   3       CALL  XXSTO TO STORE &SEQTB IN AWA                                                                       XXSEM
521   2    ELSE  (I AM BEING REENTERED FROM INTE)                                                                      XXSEM
522   3       CALL XXTMP TO SET UP TEMPORARY EXECUTION                                                                 XXSEM
523   3       CALL XXEXE TO EXECUTE ONLY THE FIRST ENTRY OF SEQTAB                                                     XXSEM
524   3       IF RESET WAS REQUESTED THEN                                                                              XXSEM
525   4          RESET CURRENT SEQUENCE # TO NEW SEQUENCE #                                                            XXSEM
526   3       ELSE                                                                                                     XXSEM
527   3       EXIT XXAUT IF TERMINAL ENTRY WAS JUST EXECUTED                                                           XXSEM
528   4          SET CURRENT SEQUENCE # TO NEXT SEQUENCE #                                                             XXSEM
529   3       ENDIF                                                                                                    XXSEM
530   2    ENDIF                                                                                                       XXSEM
531   2    DO UNTIL TERMINAL ENTRY IS EXECUTED AND IT IS NOT A REQUEST TO RESET                                        XXSEM
532   3       CALL XTCGM TO PROMPT USER WITH CURRENT ENTRY                                                             XXSEM
533   2    EXIT XXSEM IF RESPONSE IS X                                                                                 XXSEM
534   3       IF RESPONSE IS CR THEN                                                                                   XXSEM
535   4          IF THIS IS AN OVERRIDE WITH DEFAULT INTERFACE TABLE THEN                                              XXSEM
536   5             CALL XXDEF TO READ UP DEFAULT TABLE                                                               XXSEM
537   4          ENDIF                                                                                                XXSEM
538   4          SET BEGINNING SEQUENCE # TO CURRENT SEQUENCE #                                                        XXSEM
539   4          SET ENDING SEQUENCE NUMBER TO BEGINNING SEQUENCE #                                                    XXSEM
540   4          CALL XXEXE TO EXECUTE                                                                                 XXSEM
541   4          SET CURRENT SEQUENCE # TO NEXT SEQUENCE #                                                             XXSEM
542   3       ELSE                                                                                                     XXSEM
543   4          IF RESPONSE WAS A SEQUENCE # THEN                                                                     XXSEM
544   5             ERREXIT IF NUMBER IS ZERO TO :ERR1:                                                                XXSEM
545   5             SEARCH SEQUENCE TABLE FOR SEQUENCE #                                                               XXSEM
546   5             ERREXIT IF NUMBER IS NOT FOUND TO :ERR1:                                                           XXSEM
547   5             SET CURRENT SEQUENCE # TO SEQUENCE # REQUESTED                                                     XXSEM
548   4          ELSE                                                                                                  XXSEM
549   5             IF RESPONSE IS AN AMPERSAND THEN                                                                   XXSEM
550   6                SET CURRENT SEQUENCE # TO NEXT SEQUENCE #                                                       XXSEM
551   5             ELSE    (RESPONSE MUST HAVE BEEN AN OVERRIDE)                                                      XXSEM
552   6                CALL XXDEC TO DECODE PROCESSOR NAME, IT NAME RESPONSE                                           XXSEM
553   6                IF RESPONSE IS VALID THEN                                                                       XXSEM
554   7                   CALL XXSTO TO STORE OVERRIDING ENTRY                                                         XXSEM
555   7                   SAVE CURRENT SEQUENCE # IN OLD SEQUENCE #                                                    XXSEM
556   7                   SET CURRENT SEQUENCE # TO FIRST ENTRY                                                        XXSEM
557   6                ENDIF                                                                                           XXSEM
558   5             ENDIF                                                                                              XXSEM
559   4          ENDIF                                                                                                 XXSEM
560   3       ENDIF                                                                                                    XXSEM
561   2    ENDDO                                                                                                       XXSEM
562   1 EXIT XXSEM                                                                                                     XXSEM

563   2    :ERR1:                                                                                                      XXSEM
564   2    CALL XXMSG TO DISPLAY ERROR MESSAGE                                                                         XXSEM
565   1 END XXSEM                                                                                                      XXSEM
```

```
567   1 CD**************                                                                  XXSTO
568   1 CDO                                                                             XXSTO
569   1 CDO       CALLING PROCEDURE FOR XXSTO:                                          XXSTO
570   1 CDO                                                                             XXSTO
571   1 CDO             CALL XXSTO                                                      XXSTO
572   1 CDO                                                                             XXSTO
573   1 CD**************                                                                XXSTO
574   1 CD1                                                                             XXSTO
575   1 CD1       XXSTO SETS UP TO REQUEST MANAGER TO STORE &SEQTB AND HANDLES          XXSTO
576   1 CD1       ANY ERRORS RETURNED BY THE MANAGER                                    XXSTO
577   1 CD1                                                                             XXSTO
578   1 CD**************                                                                XXSTO
579   1 CD2                                                                             XXSTO
580   1 CD2       INPUTS FROM COMMON:                                                   XXSTO
581   1 CD2                                                                             XXSTO
582   1 CD2             XB(250) SEQLEN, XB(251) SEQTAB, XS(13) FLGTAB                   XXSTO
583   1 CD2                                                                             XXSTO
584   1 CD**************                                                                XXSTO
585   1 CD3                                                                             XXSTO
586   1 CD3       OUTPUTS TO COMMON:                                                    XXSTO
587   1 CD3                                                                             XXSTO
588   1 CD3             XE(5) MASSTA                                                    XXSTO
589   1 CD3                                                                             XXSTO
590   1 CD**************                                                                XXSTO
591   1 CD5                                                                             XXSTO
592   1 CD5       COMMON USED:                                                          XXSTO
593   1 CD5                                                                             XXSTO
594   1 CD5             EQUIVALENCE (XE(5),    MASSTA),                                 XXSTO
595   1 CD5            +            (XE(19),   REQPTR), (XE(20),   REQBUF),            XXSTO
596   1 CD5            +            (XB(250),  SEQLEN), (XB(251),  SEQTAB),            XXSTO
597   1 CD5            +            (XS(13),   FLGTAB)                                 XXSTO
598   1 CD5                                                                             XXSTO
599   1 CD5       FDS ROUTINES USED:                                                    XXSTO
600   1 CD5                                                                             XXSTO
601   1 CD5             XREQ, XRMOV, XRMSG, XERTN                                       XXSTO
602   1 CD5                                                                             XXSTO
603   1 CD5       RTE ROUTINES USED:                                                    XXSTO
604   1 CD5                                                                             XXSTO
605   1 CD5             EXEC                                                            XXSTO
606   1 CD5                                                                             XXSTO
607   1 CD**************                                                                XXSTO
```

```
1 BEGIN XXSTO                                                          XXSTO
  2   SET CLASS NUMBER TO ZERO                                         XXSTO
  2   IF TABLE FLAG SAYS STORE ENTIRE TABLE THEN                       XXSTO
  3     CALL EXEC TO WRITE ENTIRE TABLE                                XXSTO
  3     SET LENGTHS IN REQUEST BUFFER TO LENGTHS IN XB                 XXSTO
  3     CALL XRMOV TO MOVE DELETE, ALLOCATE AND STORE INTO REQUEST BUFFER XXSTO
  2   ELSE                                                             XXSTO
  3     CALL EXEC TO WRITE ONLY FIRST ENTRY                            XXSTO
  3     SET LENGTHS IN REQUEST BUFFER TO 7 WORDS                       XXSTO
  3     CALL XRMOV TO MOVE STORE REQUEST INTO REQUEST BUFFER           XXSTO
  2   ENDIF                                                            XXSTO
  2   CALL XREQ TO REQUEST MANAGER TO STORE USERTB                     XXSTO
  2   IF RETURN CODE IS NOT ZERO THEN                                  XXSTO
  3     CALL XRMSG TO WRITE SPACE ERROR                                XXSTO
  3     CALL EXEC TO RELEASE CLASS #                                   XXSTO
  3     SET MASTER STATE TO ZERO                                       XXSTO
  3     CALL XERTN TO RETURN TO EXEC  **AND RETURN**                   XXSTO
  2   ENDIF                                                            XXSTO
  2   RETURN                                                           XXSTO
1 END XXSTO
```

```
630    1 CD**************                                                                  XXTMP
631    1 CDO                                                                              XXTMP
632    1 CDO       FORTRAN CALLING SEQUENCE:                                              XXTMP
633    1 CDO                                                                              XXTMP
634    1 CDO            CALL XXTMP                                                        XXTMP
635    1 CDO                                                                              XXTMP
636    1 CD**************                                                                 XXTMP
637    1 CD1                                                                              XXTMP
638    1 CD1       XXTMP SETS UP A ONE ENTRY SEQUENCE TABLE USING &INTAB AND               XXTMP
639    1 CD1       STORES IT IN THE AVA TO EXECUTE WHEN EXECUTION WITH A                  XXTMP
640    1 CD2      TEMPORARY ENTRY IS NECESSARY                                            XXTMP
641    1 CD1                                                                              XXTMP
642    1 CD**************                                                                 XXTMP
643    1 CD2                                                                              XXTMP
644    1 CD2       INPUTS FROM COMMON:                                                    XXTMP
645    1 CD2                                                                              XXTMP
646    1 CD2            XE(12) SEQPTR,  XB(250) SEQLEN, XB(251) SEQTAB                    XXTMP
647    1 CD2                                                                              XXTMP
648    1 CD**************                                                                 XXTMP
649    1 CD3                                                                              XXTMP
650    1 CD3       OUTPUTS TO COMMON:                                                     XXTMP
651    1 CD3                                                                              XXTMP
652    1 CD3            XE(6)  SUBSTA, X9(249) SEQNO,  XB(250) SEQLEN,                    XXTMP
653    1 CD3            XB(251)SEQTAB, XS(13, FLGTAB                                      XXTMP
654    1 CD3                                                                              XXTMP
655    1 CD**************                                                                 XXTMP
656    1 CD5                                                                              XXTMP
657    1 CD5       COMMON USED:                                                           XXTMP
658    1 CD5                                                                              XXTMP
659    1 CD5            EQUIVALENCE (XE(6),   SUBSTA),  (XE(12),  SEQPTR),                XXTMP
660    1 CD5           +            (XE(19),  REQPTR),  (XE(20),  REQBUF),                XXTMP
661    1 CD5           +            (XB(249), SEQNO ),  (XB(250), SEQLEN),                XXTMP
662    1 CD5           +            (XB(251), SEQTAB),  (XS(6),   TMPTAB),                XXTMP
663    1 CD5           +            (XS(13),  FLGTAB)                                     XXTMP
664    1 CD5                                                                              XXTMP
665    1 CD5       FDS ROUTINES USED:                                                     XXTMP
666    1 CD5                                                                              XXTMP
667    1 CD5            XREQ, XRMOV, XXSTO                                                XXTMP
668    1 CD5                                                                              XXTMP
669    1 CD5       RTE ROUTINES USED:                                                     XXTMP
670    1 CD5                                                                              XXTMP
671    1 CD5            EXEC                                                              XXTMP
672    1 CD5                                                                              XXTMP
673    1 CD**************                                                                 XXTMP
```

```
XXTMP
XXTMP
XXTMP
XXTMP
XXTMP
XXTMP
XXTMP
XXTMP
XXTMP
XXTMP
XXTMP
```

```
675  1 BEGIN XXTMP
676  2      SET SUBSTATE TO EXECUTION CONTROLLER
677  2      CALL XREQ TO RETRIEVE &SEQTB
678  2      CALL EXEC TO GET SEQUENCE TABLE
679  2      SEARCH SEQTAB TO FIND ENTRY IN ERROR
680  2      CREATE TEMPORARY ENTRY IN SEQTAB
681  2      SET IT NAME IN TEMP ENTRY TO &INTAB
682  2      SET TABLE FLAG TO SAY TEMPORARY ENTRY ONLY
683  2      CALL XXSTO TO STORE &SEQTB
684  2      RETURN
685  1 END XXTMP
```

```
 SYMBOL DEFINITION TABLE
:AUTO   :    125
:AUTT   :    124
:ERR1   :     70
:ERR1   :    343
:ERR1   :    195
:ERR1   :    563
:ERR2   :    347
:ERR3   :    348
:ERR4   :    353
:ERR5   :    354
:ERR6   :    355
:FILERR:     269
:INT1   :    339
:INT2   :    344
:MANU   :    122
:MGRERR:     273
:PRMERR:     439
:PROMPT:     418
:RESET  :    349
:SEMI   :    123
 XXAUT        49
 XXCNT       119
 XXDEC       179
 XXDEF       249
 XXEXE       327
 XXMAN       414
 XXSEM       509
 XXSTO       609
 XXTMP       675
```



```
@XQT F.POLIST
```

```
1 CD*************                                                        ASSGN
1 CD0                                                                    ASSGN
1 CD0      ASSGN - DATA ASSIGNMENT PROCESSOR                             ASSGN
1 CD0                                                                    ASSGN
1 CD0            - SCHEDULED BY FDS                                      ASSGN
1 CD0                                                                    ASSGN
1 C**************                                                        ASSGN
1 CD1                                                                    ASSGN
1 CD1      ASSGN ALLOWS THE FDS USER TO COMPUTE VALUES AND STORE THEM IN ASSGN
1 CD1      AN EXISTING DATA ELEMENT IN THE AWA.  ASSGN SUPPORTS EXTENDED ASSGN
1 CD1      FORTRAN TYPE MIXED-MODE EXPRESSIONS AND FUNCTIONS AND ALLOWS  ASSGN
1 CD1      REPETITIVE EVALUATIONS IN ORDER TO COMPUTE AND STORE MULTIPLE ASSGN
1 CD1      VALUES                                                        ASSGN
1 CD1                                                                    ASSGN
1 C**************                                                        ASSGN
1 CD2                                                                    ASSGN
1 CD2      INPUTS FROM THE MANAGER:                                      ASSGN
1 CD2                                                                    ASSGN
1 CD2            LU     - LOGICAL UNIT OF USER'S TERMINAL                ASSGN
1 CD2            DEBUG  - FLAGS FOR DEBUG                                ASSGN
1 CD2                                                                    ASSGN
1 CD2      INPUTS FROM THE INTERFACE TABLE:                              ASSGN
1 CD2                                                                    ASSGN
1 CD2            EXP    - SYMBOLIC STRING CONTAINING DATA ASSIGNMENT     ASSGN
1 CD2                     (SEE BELOW FOR BACKUS-NAUR DEFINITION OF VALID ASSGN
1 CD2                     SYNTAX)                                        ASSGN
1 CD2                                                                    ASSGN
1 C**************                                                        ASSGN
1 CD3                                                                    ASSGN
1 CD3      OUTPUTS TO THE AWA:                                           ASSGN
1 CD3                                                                    ASSGN
1 CD3            THE COMPUTED VALUE(S) IS STORED INTO THE SPECIFIED DATA ASSGN
1 CD3            ELEMENT                                                 ASSGN
1 CD3                                                                    ASSGN
1 C**************                                                        ASSGN
1 CD4                                                                    ASSGN
1 CD4      INTERNAL VARIABLES:                                           ASSGN
1 CD4                                                                    ASSGN
1 CD4      BLANK COMMON - ASGCOM DIMENSIONED BY 2300 WORDS DEFINED AS    ASSGN
1 CD4            FOLLOWS:                                                ASSGN
1 CD4                                                                    ASSGN
1 CD4             NAME   DIMENSION  START  DESCRIPTION                   ASSGN
1 CD4            ------  ---------  -----  -----------                   ASSGN
1 CD4            PARMS       5        1    PARMS(1)= LU, PARMS(3) = DEBUG FLAGS ASSGN
1 CD4            TOKENS     32        6    IDENTIFYING NUMBERS FOR TOKENS ASSGN
1 CD4            STWIDE      1       38    SYMBOL TABLE WIDTH            ASSGN
1 CD4            STLONG      1       39    SYMBOL TABLE LENGTH           ASSGN
1 CD4            LASTSY      1       40    LAST SYMBOL TABLE ENTRY DEFINED ASSGN
1 CD4            SYMTAB    12,81     41    SYMBOL TABLE (WORDS 1-8 = TOC ENTRY OR ASSGN
1 CD4                                      APPLICABLE INFORMATION, WORDS 9-11 = ASSGN
1 CD4                                      VALUE, WORD 12 = 1 FOR INDEX, = 2 FOR ASSGN
1 CD4                                      SUBSCRIPTED DATA ELEMENT)     ASSGN
1 CD4            SSTRNG    247     1013    SYMBOLIC STRING  (EXP)        ASSGN
1 CD4            RESULT     4,35   1260    RESULT STACK USED DURING POST- ASSGN
1 CD4                                      FIX STRING EVALUATION (EACH EN- ASSGN
1 CD4                                      TRY: WORDS 1-3 CONTAIN VALUE; ASSGN
1 CD4                                      WORD 4 = DATA TYPE). DATA TYPE= ASSGN
1 CD4                                       1,2,3  FIXED DATA            ASSGN
1 CD4                                          -1  SYMBOL TABLE INDEX    ASSGN
```

```
                                                -2  DISPLACEMENT                                  ASSGN
                                                -3  CHARACTER STRING INDEX                       ASSGN
OPRNDS      4,9      1400  STACK CONTAINING OPERANDS FOR                                         ASSGN
                           FUNCTIONS AND ARITHMETIC OPER-                                       ASSGN
                           ATIONS AND RESULTS FOR STORING                                       ASSGN
                           (EA. ENTRY: WORDS 1-3 CONTAIN                                        ASSGN
                           VALUE; WORD 4 = DATA TYPE)                                           ASSGN
                           SIZED FOR C72 CHARACTER STRING                                       ASSGN
SCRTCH      4,9      1436  SCRATCH AREA                                                          ASSGN
REQST       8        1472  MANAGER REQUEST FOR XPREQ                                             ASSGN
OPINFO      7        1480  CONTENTS OF FNCTBL OR SYNTAX FOR                                      ASSGN
                           FUNCTION OR MATHEMATICAL OPERA-                                      ASSGN
                           TION BEING EVALUATED                                                 ASSGN
NUMWDS      1        1487  NUMBER OF WORDS TO BE STORED IN                                       ASSGN
                           OBJECT DATA ELEMENT                                                  ASSGN
DATYPS      9        1488  DATA TYPES                                                            ASSGN
RSLTPT      1        1497  RESULT STACK POINTER                                                  ASSGN
EXPTR       1        1498  POST-FIX STRING(POLISH) POINTER                                       ASSGN
CLSREQ      1        1499  XPREQ OPTION WORD FOR QUEUE RE-                                       ASSGN
                           QUEST AND CLOSE BUFFER - NO                                          ASSGN
                           DATA TRANSFERED                                                      ASSGN
CLSTRN      1        1500  XPREQ OPTION WORD FOR QUEUE RE-                                       ASSGN
                           QUEST,CLOSE BUFFER AND TRANSFER                                      ASSGN
                           DATA                                                                 ASSGN
MAPWDS      9        1501  NUMBER OF WORDS PER LOGICAL                                           ASSGN
                           UNIT OF DATA FOR EACH DATA TYPE                                      ASSGN
DECLAS      1        1510  DATA ELEMENT CLASS                                                    ASSGN
RNGSTK      4,4      1511  BEGIN RANGE, END RANGE, INCRE-                                        ASSGN
                           MENT AND SYMBOL INDEX FOR EACH                                       ASSGN
                           RANGE SPECIFICATION                                                  ASSGN
POLISH      161      1527  POST-FIX REPRESENTATION OF EX-                                        ASSGN
                           PRESSION                                                             ASSGN
SYNTAX      7,40     1688  SYNTAX TABLE FOR VALIDITY TESTS                                       ASSGN
                           ON EXPRESSION (SEE BELOW)                                            ASSGN
FNCTBL      7,36     1968  FUNCTION TABLE CONTAINING DATA                                        ASSGN
                           REQUIREMENTS FOR EACH FUNCTION                                       ASSGN
                           (SEE BELOW)                                                          ASSGN
MRBUFF      64       2220  XPREQ BUFFER                                                          ASSGN
                                                                                                ASSGN
                                                                                                ASSGN
NOTE : STACKS USED IN THE ASSGN PROCESSOR ARE SIZED FOR                                         ASSGN
THE MAXIMUM POSSIBLE AND OVERFLOW IS NOT TESTED                                                 ASSGN
                                                                                                ASSGN
                                                                                                ASSGN
TABLE DEFINITIONS:                                                                              ASSGN
```

SYNTAX - 7X40 TABLE CONTAINING INFORMATION FOR PROCESSING EACH TOKEN

TOKEN	INPUT PRIORITY	OUTPUT PRIORITY	TOKEN LENGTH	#OPRNDS-1 /OPERATOR	DATA TYPE REQUIREMENTS	PRECEDENT TYPE	VALID PRECEDENTS
(EOS)	-	-	2	-	-	4	OP(LBRKT,=
INTEGER	-	-	3	-	-	4	OP(LBRKT,=
REAL	-	-	4	-	-	4	OP(LBRKT,=
DOUBLE	-	-	4	-	-	4	OP(LBRKT,=
NAME	-	-	4	-	-	2	OP(LB,=NUL
"	-	-	?	-	-	1000B	=
+	4	4	1	1	SAME	10B	)RB VR CN
-	4	4	1	1	SAME	10B	)RB VR CN
*	5	5	1	1	SAME	10B	)RB VR CN
/	5	5	1	1	SAME	10B	)RB VR CN
(<)							
(>)							
(#)							
(@)							
=	9	1	1	1	SAME	400B	) VR
(&)							
(?)							
(	8	2	1	-	-	20B	VR OP(LB,=
)	2	2	1	-	-	40B	)RB VR CN
(BSS)							
ESS	0	-	1	-	-	3000B	")RB VR CN
(%)							
(BKSLASH)							
($)							
(.)							
LB	8	2	1	-	-	20B	FUNCTION
RB	2	2	1	-	-	100B	)RB VR CN
(R)							
,	3	3	1	1	INTEGER	20B	)RB VR CN
;	0	-	1	-	-	3000B	")RB VR CN
SUBSCRPT	-	-	-	0	INTEGER	-	
**	6	6	2	1	SAME	10B	)RB VR CN
UNARY-	4	4	1	0	SAME	10B	(LB,=
FUNCTION	7	7	4	-	-	1	OP(LB,=

VR = VARIABLE OP = OPERATOR LB = LEFT BRACKET CN = CONSTANT
RB = RIGHT BRACKET

```
1 CD4                                                                                   ASSGN
1 CD4    FNCTBL - 7X36 TABLE CONTAINING INFORMATION FOR PROCESSING FUNCTIONS             ASSGN
1 CD4                                                                                   ASSGN
1 CD4                                         OUTPUT    FIRST    SECOND                  ASSGN
1 CD4    NAME(WORDS 1-3)   #OPERANDS-1        TYPE     OP TYPE   OP TYPE                 ASSGN
1 CD4    ---------------   -----------       ------    -------   -------                 ASSGN
1 CD4       ABS                 0              REAL     REAL                             ASSGN
1 CD4       AINT                0              REAL     REAL                             ASSGN
1 CD4       ALOG                0              REAL     REAL                             ASSGN
1 CD4       ALOGT               0              REAL     REAL                             ASSGN
1 CD4       AMOD                1              REAL     REAL      REAL                   ASSGN
1 CD4       ATAN                0              REAL     REAL                             ASSGN
1 CD4       ATAN2               1              REAL     REAL      REAL                   ASSGN
1 CD4       COS                 0              REAL     REAL                             ASSGN
1 CD4       DABS                0             DOUBLE   DOUBLE                            ASSGN
1 CD4       DATAN               0             DOUBLE   DOUBLE                            ASSGN
1 CD4       DATN2               1             DOUBLE   DOUBLE    DOUBLE                  ASSGN
1 CD4       DBLE                0             DOUBLE    REAL                             ASSGN
1 CD4       DCOS                0             DOUBLE   DOUBLE                            ASSGN
1 CD4       DDINT               0             DOUBLE   DOUBLE                            ASSGN
1 CD4       DLOG                0             DOUBLE   DOUBLE                            ASSGN
1 CD4       DLOGT               0             DOUBLE   DOUBLE                            ASSGN
1 CD4       DMOD                1             DOUBLE   DOUBLE    DOUBLE                  ASSGN
1 CD4       DSIGN               1             DOUBLE   DOUBLE    DOUBLE                  ASSGN
1 CD4       DSIN                0             DOUBLE   DOUBLE                            ASSGN
1 CD4       DSQRT               0             DOUBLE   DOUBLE                            ASSGN
1 CD4       DTAN                0             DOUBLE   DOUBLE                            ASSGN
1 CD4       DTANH               0             DOUBLE   DOUBLE                            ASSGN
1 CD4       FLOAT               0              REAL    INTEGER                           ASSGN
1 CD4       IABS                0             INTEGER  INTEGER                           ASSGN
1 CD4       IDINT               0             INTEGER  DOUBLE                            ASSGN
1 CD4       IFIX                0             INTEGER   REAL                             ASSGN
1 CD4       ISIGN               1             INTEGER  INTEGER   INTEGER                 ASSGN
1 CD4       MOD                 1             INTEGER  INTEGER   INTEGER                 ASSGN
1 CD4       SIGN                1              REAL     REAL      REAL                   ASSGN
1 CD4       SIN                 0              REAL     REAL                             ASSGN
1 CD4       SNGL                0              REAL    DOUBLE                            ASSGN
1 CD4       SQRT                0              REAL     REAL                             ASSGN
1 CD4       TAN                 0              REAL     REAL                             ASSGN
1 CD4       TANH                0              REAL     REAL                             ASSGN
1 CD4       EXP                 0              REAL     REAL                             ASSGN
1 CD4       DEXP                0             DOUBLE   DOUBLE                            ASSGN
1 CD************                                                                        ASSGN
```

```
1 CD5      ROUTINES REFERENCED - RMPAR, XPGET, XRMOV, XZ?S1, XZPS2, XPXIT                ASSGN
1 CD5                                                                                    ASSGN
1 CD5                                                                                    ASSGN
1 CD5      BACKUS-NAUR LANGUAGE DEFINITION                                               ASSGN
1 CD5                                                                                    ASSGN
1 CD5              <ASSIGNMENT>    ::=   <REPLACEMENT> <RANGE>  //                       ASSGN
1 CD5                                    <EVALUATION> <RANGE>                            ASSGN
1 CD5                                                                                    ASSGN
1 CD5             <REPLACEMENT>    ::=   <NON-NUMERIC DE> = <NON-NUMERIC DE>  //         ASSGN
1 CD5                                    <NON-NUMERIC DE> = "CHARACTER STRING"  //       ASSGN
1 CD5                                    <VARIABLE> = FREE DATA ELEMENT                  ASSGN
1 CD5                                                                                    ASSGN
1 CD5          <NON-NUMERIC DE>    ::=   FREE DATA ELEMENT  //                           ASSGN
1 CD5                                    FREE DATA ELEMENT(<SUBSCRIPT LIST>)  //         ASSGN
1 CD5                                    CHARACTER DATA ELEMENT  //                      ASSGN
1 CD5                                    CHARACTER DATA ELEMENT(<SUBSCRIPT LIST>)        ASSGN
1 CD5                                                                                    ASSGN
1 CD5              <EVALUATION>    ::=   <VARIABLE> = <EXPRESSION>  //                   ASSGN
1 CD5                                    FREE DATA ELEMENT = <EXPRESSION>  //            ASSGN
1 CD5                                    FREE DATA ELEMENT(<SUBSCRIPT LIST>) =           ASSGN
1 CD5                                    <EXPRESSION>                                    ASSGN
1 CD5                                                                                    ASSGN
1 CD5                <VARIABLE>    ::=   FIXED DATA ELEMENT  //                          ASSGN
1 CD5                                    FIXED DATA ELEMENT(<SUBSCRIPT LIST>)            ASSGN
1 CD5                                                                                    ASSGN
1 CD5          <SUBSCRIPT LIST>    ::=   <SUBSCRIPT LIST>,<EXPRESSION>  //               ASSGN
1 CD5                                    <EXPRESSION>                                    ASSGN
1 CD5                                                                                    ASSGN
1 CD5              <EXPRESSION>    ::=   <EXPRESSION> <ADDITIVE OPERATOR> <TERM>  //     ASSGN
1 CD5                                    <UNARY OPERATOR> <TERM>                         ASSGN
1 CD5                                                                                    ASSGN
1 CD5                    <TERM>    ::=   <TERM> <MULTIPLICATIVE OPERATOR> <FACTOR> //    ASSGN
1 CD5                                    <FACTOR>                                        ASSGN
1 CD5                                                                                    ASSGN
1 CD5                  <FACTOR>    ::=   <POWER>**<POWER>  //                            ASSGN
1 CD5                                    <POWER>                                         ASSGN
1 CD5                                                                                    ASSGN
1 CD5                   <POWER>    ::=   (<EXPRESSION>)  //                              ASSGN
1 CD5                                    <OPERAND>                                       ASSGN
1 CD5                                                                                    ASSGN
1 CD5          <UNARY OPERATOR>    ::=   <ADDITIVE OPERATOR>  //                         ASSGN
1 CD5                                    <>                                              ASSGN
1 CD5                                                                                    ASSGN
1 CD5       <ADDITIVE OPERATOR>    ::=   +  // -                                         ASSGN
1 CD5                                                                                    ASSGN
1 CD5 <MULTIPLICATIVE OPERATOR>    ::=   *  // /                                         ASSGN
1 CD5                                                                                    ASSGN
1 CD5                 <OPERAND>    ::=   FUNCTION NAME <LB> <FUNCTION LIST> <RB>  //     ASSGN
1 CD5                                    <VARIABLE>  //                                  ASSGN
1 CD5                                    <CONSTANT>                                      ASSGN
1 CD5                                                                                    ASSGN
1 CD5                      <LB>    ::=   LEFT BRACKET                                    ASSGN
1 CD5                                                                                    ASSGN
1 CD5                      <RB>    ::=   RIGHT BRACKET                                   ASSGN
1 CD5                                                                                    ASSGN
1 CD5           <FUNCTION LIST>    ::=   <FUNCTION LIST>,<EXPRESSION>  //                ASSGN
1 CD5                                    <EXPRESSION>                                    ASSGN
1 CD5                                                                                    ASSGN
```

```
260    1 CD5              <CONSTANT>   ::=   INTEGER  //                                    ASSGN
261    1 CD5                                SINGLE PRECISION REAL  //                       ASSGN
262    1 CD5                                DOUBLE PRECISION REAL                           ASSGN
263    1 CD5                                                                                ASSGN
264    1 CD5                 <RANGE>   ::=  <RANGE> <LIMITS>  //                           ASSGN
265    1 CD5                                <RANGE>  //                                     ASSGN
266    1 CD5                                <>                                              ASSGN
267    1 CD5                                                                                ASSGN
268    1 CD5                 <LIMITS>  ::=  ;INDEX=INTEGER,INTEGER                         ASSGN
269    1 CD5                                                                                ASSGN
270    1 CD**************                                                                   ASSGN
```

```
272    1 BEGIN ASSGN                                                                    ASSGN
273    2    INITIALIZE COMMON                                                           ASSGN
274    2    CALL XZPS1 TO BUILD POST--FIX STRING                                        ASSGN
275    2    CALL XZPS2 TO EVALUATE EXPRESSION AND STORE VALUE(S)                        ASSGN
276    2    CALL XPXIT TO EXIT PROCESSOR                                                ASSGN
277    1 END ASSGN                                                                      ASSGN
```

```
279   1 CD *******                                                                  DBDSP
280   1 CD 0                                                                        DBDSP
281   1 CD 0      DBDSP - DATA BOX DISPLAY PROCESSOR                                DBDSP
282   1 CD 0                                                                        DBDSP
283   1 CD C            - SCHEDULED BY FDS                                          DBDSP
284   1 CD 0                                                                        DBDSP
285   1 CD *******                                                                  DBDSP
286   1 CD 1                                                                        DBDSP
287   1 CD 1      DBDSP PROCESSES THE DATA PRODUCED BY SCANNING FUNCTIONAL          DBDSP
288   1 CD 1      PROCESSORS. IT GENERATES A DIGITAL MATRIX DISPLAY CONTAINING      DBDSP
289   1 CD 1      THE VALUES OF ANY PARAMETER OR PAIR OF PARAMETERS OVER THE        DBDSP
290   1 CD 1      SCANNED SPACE                                                     DBDSP
291   1 CD 1                                                                        DBDSP
292   1 CD 1      BECAUSE OF THE SIZE OF THIS PROCESSOR, IT WAS DIVIDED INTO        DBDSP
293   1 CD 1      THREE OVERLAYS AS FOLLOW                                          DBDSP
294   1 CD 1                                                                        DBDSP
295   1 CD 1                  XZDIN - READS AND EDITS INTERFACE TABLE               DBDSP
296   1 CD 1                                                                        DBDSP
297   1 CD 1                  XZDP1 - READS DRDE A ; VALIDATES NAMES IN             DBDSP
298   1 CD 1                          INTERFACE TABLE AGAINST NAMES IN              DBDSP
299   1 CD 1                          DATABOX SCAN.  WITH SUBROUTINE XIDMK          DBDSP
300   1 CD 1                          IT DEVELOPS CONSTRAINT MASKS FOR              DBDSP
301   1 CD 1                          ARRAYS                                        DBDSP
302   1 CD 1                                                                        DBDSP
303   1 CD 1                  XZDP2 - PROMPTS USER IF REQUIRED AND OUTPUTS          DBDSP
304   1 CD 1                          REQUESTED PAGE ARRAYS OF UP TO TWO            DBDSP
305   1 CD 1                          VARIABLES TO THE SPECIFIED LU DEVISE          DBDSP
306   1 CD 1                          WITH SUBROUTINE XZDOT                         DBDSP
307   1 CD 1                                                                        DBDSP
308   1 CD 1      DBDSP MERELY CALLS THESE OVERLAYS IN THE PROPER ORDER AND         DBDSP
309   1 CD 1      EXITS                                                             DBDSP
310   1 CD 1                                                                        DBDSP
311   1 CD 1                                                                        DBDSP
312   1 CD *******                                                                  DBDSP
313   1 CD 2                                                                        DBDSP
314   1 CD 2      INPUTS TO DBDSP FROM INTERFACE TABLE                              DBDSP
315   1 CD 2                                                                        DBDSP
316   1 CD 2           DATBOX - DATA BOX FILE NAME                                  DBDSP
317   1 CD 2           NDVAR  - DISPLAY VARIABLE NAME LIST SET UP BY USER           DBDSP
318   1 CD 2           KEXP   - DISPLAY VARIABLE SCALE LIST SET UP BY USER          DBDSP
319   1 CD 2           VIODEF - CONSTRAINT VARIABLE DEFINITION LIST                 DBDSP
320   1 CD 2                                                                        DBDSP
321   1 CD 2      INPUTS TO DBDSP FROM DRDE FILE                                    DBDSP
322   1 CD 2                                                                        DBDSP
323   1 CD 2           RECORD 1                                                     DBDSP
324   1 CD 2               (1) - NAME OF FDS PROCESSOR CREATING FILE                DBDSP
325   1 CD 2               (4) - INTERFACE TABLE VARIABLE NAME FOR THIS FILE        DBDSP
326   1 CD 2               (7) - NAME OF FDS PROCESSOR UPDATING FILE                DBDSP
327   1 CD 2                     ( 3 ASCII WORDS OF BLANKS )                        DBDSP
328   1 CD 2              (10) - INTERFACE TABLE VARIABLE NAME FOR T&IS U-DATE       DBDSP
329   1 CD 2                     ( 3 ASCII WORDS OF BLANKS )                        DBDSP
330   1 CD 2                                                                        DBDSP
331   1 CD 2           RECORD 2                                                     DBDSP
332   1 CD 2                     (1N - NO OF ENTRIES IN SUMMARY TABLE               DBDSP
333   1 CD 2                     (2) - X SCAN VARIABLE ( 6 CHAR)                    DBDSP
334   1 CD 2                     (5) - X FIRST SUBSCRIPT (INT OR ZERO)              DBDSP
335   1 CD 2                     (6) - XSECOND SUBSCRIPT (INT OR ZERO)              DBDSP
336   1 CD 2                     (7) -X UNITS ( 6 CHAR)                             DBDSP
337   1 CD 2                     (10) - X CENTROID (REAL)                           DBDSP
```

```
1 CD 2                        (12) - X INCREMENT ( REAL )                                    DBDSP
1 CD 2                        (14) - X NUMBER OF STEPS (INTEGER 1-5)                         DBDSP
1 CD 2                        (15) - YSCAN VARIABLE ( 6 CHAR)                                DBDSP
1 CD 2                        (18) - Y FIRST SUBSCPT ( INT OR ZERO)                          DBDSP
1 CD 2                        (19) - Y SECOND SUBSCRIPT (INT OR ZERO)                        DBDSP
1 CD 2                        (20) - Y UNITS ( 6 CHAR )                                      DBDSP
1 CD 2                        (23) - Y CENTROID (REAL)                                       DBDSP
1 CD 2                        (25) -YX INCREMENT ( REAL )                                    DBDSP
1 CD 2                        (27) - Y NUMBER OF STEPS (INTEGER 1-5)                         DBDSP
1 CD 2                                                                                       DBDSP
1 CD 2                                                                                       DBDSP
1 CD 2                                                                                       DBDSP
1 CD 2          DEPENDANT VARIABLE NAME AND UNITS IN RECORDS 38 48 AND 5                     DBDSP
1 CD 2                                                                                       DBDSP
1 CD 2                                                                                       DBDSP
1 CD 2                                                                                       DBDSP
1 CD 2                                                                                       DBDSP
1 CD 2          SUMMARY TABLE RECORDS                                                        DBDSP
1 CD 2                                                                                       DBDSP
1 CD 2                - EACH SUMMARY TABLE CONTAIN VALUE FOR EACH                            DBDSP
1 CD 2                  DEPENDANT VARIABLE SCANNED (UP TO 32 VALUES                          DBDSP
1 CD 2                  INCLUDING ERROR FLAG WHICH IS FIRST VALUE                            DBDSP
1 CD 2                  IN SUMMARY TABLE)                                                    DBDSP
1 CD 2                                                                                       DBDSP
1 CD *******                                                                                 DBDSP
1 CD 3                                                                                       DBDSP
1 CD 3      OUTPUT FROM DBDSP                                                                DBDSP
1 CD 3                                                                                       DBDSP
1 CD 3         - DISPLAY IS CUTPUT TO LU IDENTIFIED IN THE INTERFACE TABLE                   DBDSP
1 CD 3                                                                                       DBDSP
1 CD *******                                                                                 DBDSP
1 CD 4  COMBUF - BUFFER TO RECEIVE ENCODED USER RESPONSE - XPRGM                             DBDSP
1 CD 4  DATBOX - NAME OF DATA BOX TO BE DISPLAYED BY DBDSP                                   DBDSP
1 CD 4  DATBUF - COMMON BUFFER FOR SCAN SUMMARY DESCRIPTORS                                  DBDSP
1 CD 4  IDV1   - POINTER TO DEP. DISPLAY VARIABLE NAME LIST FOR FIRST VAR                    DBDSP
1 CD 4  IDV2   - POINTER TO DEP. DISPLAY VARIABLE NAME LIST FOR 2ND VAR.                     DBDSP
1 CD 4  ISAVE  - LIST OF VIOLATED CONSTRAINTS BUILT BY XZMSK (MAX OF 8)                      DBDSP
1 CD 4  IXSCN1 - FIRST SUBSCRIPT FOR NAME OF X SCAN VARIABLE ( INT OR 0 )                    DBDSP
1 CD 4  IXSCN2 - SECOND SUBSCRIPT FOR NAME OF X SCAN VARIABLE (INT OR 0)                     DBDSP
1 CD 4  IFBUF  - NAME OF COMMON AREA USED FOR INTERFACE TABLE                                DBDSP
1 CD 4  IYSCN1 - FIRST SUBSCRIPT FOR NAME OF Y SCAN VARIABLE (INT OR 0)                      DBDSP
1 CD 4  IYSCN2 - SECOND SUBSCRIPT FOR NAME OF Y SCAN VARIABLE (INT OR 0)                     DBDSP
1 CD 4  LENBF  - LENGTH OF COMBUF IN 16-BIT WORDS - XPROM                                    DBDSP
1 CD 4  LENST  - LENGTH IN WDS OF CHAR STRING USED FOR USER PROMPT -XPRDM                    DBDSP
1 CD 4  MASK   - ARRAY CONTAINING MASKS FOR UP TO 8 CONSTR( 4 X 121 SIZE)                    DBDSP
1 CD 4  NAMVRL - NAME LIST FOR VARIABLES SCANNED BY SCAN/ENDSCN                              DBDSP
1 CD 4  NC     - NUMBER OF CONSTRAINTS INPUT BY USER  (INTEGER)                              DBDSP
1 CD 4  NCRELL - LIST OF CONSTRAINT RELATIONS INPUT BY USER                                  DBDSP
1 CD 4  NCVARL - LIST OF CONSTRAINT VARIABLE NAMES INPUT BY USER (32 MAX)                    DBDSP
1 CD 4  NDVARL - LIST OF DEP DISP VARIABLE PAIRS FOR PAGED OUTPUT(1-16PR)                    DBDSP
1 CD 4  NDVRCT - NUMBER OF DEP DISP VAR IN NDVARL LIST  (INTEGER)                            DBDSP
1 CD 4  NDVRUL - LIST OF DEP DISP VARIABLE PAIR SCALE FACTORS FOR O/P                        DBDSP
1 CD 4  MSKERR - SET OF INDICATORS FOR CONSTRAINTS VIOLATED=0/NOT   0                        DBDSP
1 CD 4  NXSTEP - NUMBER OF STEPS ON EITHER SIDE OF X CENTROID  (0 TO 5)                      DBDSP
1 CD 4  NYSTEP - NUMBER OF STEPS ON EITHER SIDE OF Y CENTROID  (0 TO 5)                      DBDSP
1 CD 4  RETC   - RTN CODE FM XPRDMY; 0=NORM,1=RTN TO EXEC,2=NULL OPR,3=ERR                   DBDSP
1 CD 4  STRING - CHAR STRING CONTAINING USER PROMPT MESSAGE                                  DBDSP
1 CD 4  XCORD  - LIST OF X VAR VALUES FOR X COORDINATES  (1 - 11 REAL)                       DBDSP
1 CD 4  XSCNNM - NAME OF X VAR SCANNED TO BE PLACED ON DISPLAY (6 CHAR)                      DBDSP
```

```
397    1 CC 4  XUNITS - NAME OF X VAR UNITS TO BE PLACED ON DISPLAY   (6 CHAR)                    DBDSP
398    1 CD 4  YCORD  - LIST OF X VAR VALUES FOR Y COORDINATES  (1 - 11 REAL)                    DBDSP
399    1 CD 4  YSCNNM - NAME OF Y VAR SCANNED TO BE PLACED ON DISPLAY (6 CHAR)                   DBDSP
400    1 CD 4  YUNITS - NAME OF Y VAR UNITS TO BE PLACED ON DISPLAY   (6 CHAR)                  DBDSP
401    1 CD 4  ZTABLE - TABLE IN COMMON FOR SUMTAB VARIABLE NAMES AND UNITS                      DBDSP
402    1 CD 4  NAMVUL - UNITS LIST FOR VARIABLES SCANNED BY SCAN/ENDSCN                          DBDSP
403    1 CD 4  SUMTAB - VALUES FOR SCAN VARIABLE(S) - 1 TO 32 VALUES/RECORD                      DBDSP
404    1 CD 4  PARMS  - COMMUNICATION BUFFER FOR RMPAR - LU, USER ID, FLAGS                      DBDSP
405    1 CD 4  LU     - LOGICAL UNIT # FOR XPRDM CALLING SEQUENCE - USER LOCATN                  DBDSP
406    1 CD 4  LUDSP  - DBDSP WILL OUTPUT DISPLAY TO THIS USER SUPPLIED LU                       DBDSP
407    1 CD 4  PROMPT - TABLE IN COMMON TO COMMUNICATE WITH XPRDM                               DBDSP
408    1 CD 4  DEBUG                                                                            DBDSP
409    1 CD 4  SELECT - SELECT =0 PROMPT ; SELECT NOT 0 RUN ALL DISPLAYS TO O/P                  DBDSP
410    1 CD 4           WITHOUT PROMPT                                                          DBDSP
411    1 CD 4  CARTRG - CARTRAGE USED TO LOCATE DATA BOX                                        DBDSP
412    1 CD 4  CVALUE -                                                                         DBDSP
413    1 CD 4  IDCB   -                                                                         DBDSP
414    1 CD *******                                                                             DBDSP
415    1 CD 5                                                                                   DBDSP
416    1 CD 5      USES ROUTINES                                                                DBDSP
417    1 CD 5                                                                                   DBDSP
418    1 CD 5         XPRDM, XELBS, XPXIT, RMPAR                                                DBDSP
419    1 CD 5                                                                                   DBDSP
420    1 CD 5                                                                                   DBDSP
421    1 CD *******                                                                             DBDSP
```

```
431   1 CD**************                                                    DEFIN
432   1 CDO                                                                DEFIN
433   1 CDO      DEFIN IS AN FDS PROCESSOR SCHEDULED BY THE MANAGER        DEFIN
434   1 CDO                                                                DEFIN
435   1 CD**************                                                    DEFIN
436   1 CD1                                                                DEFIN
437   1 CD1      DEFIN ALLOCATES DATA ELEMENTS IN THE AWA THAT WERE SPECIFIED DEFIN
438   1 CD1      BY THE PARAMETER KEYWORD DEFINE.  IF THE DATA ELEMENT ALREADY DEFIN
439   1 CD1      EXISTS, IT IS DELETED AND REALLOCATED.  DATA ELEMENTS ARE  DEFIN
440   1 CD1      INITIALIZED TO ZEROS (CHARACTER STRINGS TO BLANKS).       DEFIN
441   1 CD1                                                                DEFIN
442   1 CD**************                                                    DEFIN
443   1 CD2                                                                DEFIN
444   1 CD2      INPUTS FROM THE MANAGER:                                  DEFIN
445   1 CD2                                                                DEFIN
446   1 CD2           LU    - LOGICAL UNIT OF THE USER'S TERMINAL          DEFIN
447   1 CD2           DEBUG - FLAGS FOR DEBUG                            DEFIN
448   1 CD2                                                                DEFIN
449   1 CD2      INPUTS FROM THE INTERFACE TABLE:                          DEFIN
450   1 CD2                                                                DEFIN
451   1 CD2           DEFINE - SYMBOLIC STRING CONTAINING DATA ELEMENT NAME(S), DEFIN
452   1 CD2                    OPTIONAL I AND J DIMENSIONS AND A REQUIRED TYPE DEFIN
453   1 CD2                                                                DEFIN
454   1 CD**************                                                    DEFIN
455   1 CD3                                                                DEFIN
456   1 CD3      OUTPUTS TO THE AWA:                                       DEFIN
457   1 CD3                                                                DEFIN
458   1 CD3           SET OF DATA ELEMENT(S) REQUESTED                     DEFIN
459   1 CD3                                                                DEFIN
460   1 CD**************                                                    DEFIN
461   1 CD4                                                                DEFIN
462   1 CD4      INTERNAL VARIABLES:                                       DEFIN
463   1 CD4                                                                DEFIN
464   1 CD4           INTBUF - INTERFACE TABLE HEADER                      DEFIN
465   1 CD4           ISLENG - LENGTH OF SYMBOLIC STRING                   DEFIN
466   1 CD4           ITOKEN - POSITION WITHIN THE SYMBOLIC STRING         DEFIN
467   1 CD4           MAWARQ - BUFFER FOR VALID NAMES TO BE ALLOCATED      DEFIN
468   1 CD4           MRBUFF - BUFFER AREA FOR XPGET AND XPREQ USE         DEFIN
469   1 CD4           NENT   - NUMBER OF ENTRIES IN THIS AWA REQUEST       DEFIN
470   1 CD4           NNAME  - NUMBER OF NAMES IN SYMBOLIC STRING          DEFIN
471   1 CD4           NEXTNM - TOKEN POSITION FOR NEXT NAME                DEFIN
472   1 CD4           STRING - SYMBOLIC STRING INPUT TO DEFIN              DEFIN
473   1 CD4                                                                DEFIN
474   1 CD**************                                                    DEFIN
475   1 CD5                                                                DEFIN
476   1 CD5      EXTERNAL ROUTINES USED:                                   DEFIN
477   1 CD5                                                                DEFIN
478   1 CD5           EXEC,  IAND,  KCVT,  RMPAR, XPGET,                   DEFIN
479   1 CD5           XPREQ, XPXIT, XUDBG, XIDFT, XINSG                    DEFIN
480   1 CD5                                                                DEFIN
481   1 CD**************                                                    DEFIN
```

```
483    1 BEGIN DEFIN                                                                                                    DEFIN
484    2    CALL RMPAR TO GET LU AND DEBUG FLAGS                                                                        DEFIN
485    2    CALL XPGET TO RETRIEVE SYMBOLIC STRING                                                                      DEFIN
486    2    GET STRING LENGTH FROM INTERFACE TABLE HEADER                                                               DEFIN
487    2    STARTSEARCH WHILE TOKEN-POSITION .LT. STRING-LENGTH, OR                                                     DEFIN
488    3       WHILE CURRENT-TOKEN .NE. END-OF-STRING                                                                   DEFIN
489    3       PERFORM XZDNM TO FIND THE NEXT NAME IN THE SYMBOLIC STRING                                               DEFIN
490    3       PERFORM XZDPN TO PROCESS THE CURRENT NAME                                                                DEFIN
491    2    EXITIF THERE WAS AN ERROR RETURN FROM XZDPN                                                                 DEFIN
492    3       CALL XZMSG TO DISPLAY SYNTAX ERROR AND POSITION IN SYMBOLIC STRING                                       DEFIN
493    3       SET PROCESSOR RETURN CODE TO ABEND                                                                       DEFIN
494    2    ORELSE                                                                                                      DEFIN
495    3       INCREMENT TO NEXT ELEMENT IN THE SYMBOLIC STRING                                                         DEFIN
496    2    ENDLOOP                                                                                                     DEFIN
497    3       SET PROCESSOR RETURN FOR NORMAL EXIT                                                                     DEFIN
498    2    ENDSEARCH                                                                                                   DEFIN
499    2    SET OPTION SO XPREQ WILL DO A QUEUE REQUEST                                                                 DEFIN
500    2    DO FOR NUMBER OF REQUESTS IN REQUEST BUFFER (MAXARQ)                                                        DEFIN
501    3       IF THIS IS LAST REQUEST, THEN                                                                            DEFIN
502    4          SET OPTION TO CLOSE REQUEST BUFFER                                                                    DEFIN
503    3       ENDIF                                                                                                    DEFIN
504    3       COMPUTE INDEX TO THIS REQUEST                                                                            DEFIN
505    3       CALL XPREQ TO QUEUE THIS REQUEST                                                                         DEFIN
506    2    ENDDO                                                                                                       DEFIN
507    2    CALL XPXIT TO RETURN TO THE MANAGER                                                                         DEFIN
508    1 END DEFIN
```

```
510    1 BEGIN XZDNN                                                                                    DEFIN
511    2    SET NEXT NAME TO ZERO                                                                       DEFIN
512    2    DO WHILE NEXT NAME IS ZERO                                                                  DEFIN
513    3       CALL XZDFT TO FIND THE NEXT LEFT PAREN                                                   DEFIN
514    3       CALL XZDFT TO FIND THE NEXT COMMA                                                        DEFIN
515    3       IF THERE ARE NO SUBSCRIPTS (COMMA PRECEEDS LEFT PAREN) THEN                              DEFIN
516    4          SET NEXT NAME TO COMMA POSITION + 1                                                   DEFIN
 17    3       ELSE                                                                                     DEFIN
 18    4          CALL XZDFT TO FIND THE NEXT RIGHT PAREN                                               DEFIN
519    4          SET STARTING POSITION TO RIGHT PAREN POSITI   + 1                                     DEFIN
520    4          IF STARTING POSITION > END OF STRING THEN                                             DEFIN
521    5             SET STARTING POSITION = END OF STRING                                              DEFIN
522    4          ENDIF                                                                                 DEFIN
523    3       ENDIF                                                                                    DEFIN
524    2    ENDDO                                                                                       DEFIN
525    1 END XZDNN                                                                                      DEFIN
```

```
527   1 BEGIN XZDPN                                                                    DEFIN
528   2    SET IDIM AND JDIM TO 1                                                      DEFIN
529   2    IF TOKEN IS NOT A NAME THEN                                                 DEFIN
530   3       SET ERROR CODE                                                           DEFIN
531   2    ELSE                                                                        DEFIN
532   3       MOVE NAME INTO REQUEST                                                   DEFIN
533   3       INCREMENT TO NEXT TOKEN                                                  DEFIN
534   3       IF THERE ARE SUBSCRIPTS (TOKEN IS A LEFT PAREN) THEN                     DEFIN
535   4          INCREMENT TO NEXT TOKEN                                               DEFIN
536   4          IF TOKEN IS NOT AN INTEGER OR                                         DEFIN
537   5             TOKEN IS NOT ZERO THEN                                             DEFIN
538   5             CALL XZMSG TO DISPLAY ERROR "INVALID IDIM"                         DEFIN
539   4          EXIT TO :PNERR1:                                                      DEFIN
540   4          ENDIF                                                                 DEFIN
541   4          SET IDIM TO THIS TOKEN                                                DEFIN
542   4          INCREMENT TO NEXT TOKEN                                               DEFIN
543   4          IF THERE ARE TWO SUBSCRIPTS (TOKEN IS A COMMA) THEN                   DEFIN
544   5             INCREMENT TO NEXT TOKEN                                            DEFIN
545   5             IF TOKEN IS NOT AN INTEGER OR                                      DEFIN
546   6                TOKEN IS NOT ZERO THEN                                          DEFIN
547   6                CALL XZMSG TO DISPLAY ERROR "INVALID JDIM"                      DEFIN
548   5             EXIT TO :PNERR1:                                                   DEFIN
549   5             ENDIF                                                              DEFIN
550   5             SET JDIM TO THIS TOKEN                                             DEFIN
551   5             INCREMENT TO NEXT TOKEN                                            DEFIN
552   4          ENDIF                                                                 DEFIN
553   4          IF TOKEN IS NOT A RIGHT PAREN THEN                                    DEFIN
554   5             CALL XZMSG TO DISPLAY ERROR "INVALID SUBSCRIPT DELIMETER"          DEFIN
555   4          EXIT TO :PNERR1:                                                      DEFIN
556   4          ENDIF                                                                 DEFIN
557   4          INCREMENT TO NEXT TOKEN                                               DEFIN
558   3       ENDIF                                                                    DEFIN
559   3       IF TOKEN IS NOT A BEGIN TYPE FIELD SLASH THEN                            DEFIN
560   4          CALL XZMSG TO DISPLAY ERROR "INVALID OR MISSING TYPE FIELD"           DEFIN
561   3       EXIT TO :PNERR1:                                                         DEFIN
562   3       ENDIF                                                                    DEFIN
563   3       INCREMENT TO NEXT TOKEN                                                  DEFIN
564   3       IF TOKEN IS NOT A NAME THEN                                              DEFIN
565   4          CALL XZMSG TO DISPLAY ERROR "INVALID OR MISSING TYPE FIELD"           DEFIN
566   3       EXIT TO :PNERR1:                                                         DEFIN
567   3       ENDIF                                                                    DEFIN
568   3       INCREMENT TO NEXT TOKEN                                                  DEFIN
569   3       STARTSEARCH FOR ALL VALID TYPES                                          DEFIN
570   3       EXITIF TYPE MATCHES THE TYPE IN THE SYMBOLIC STRING                      DEFIN
571   4          SET TYPE AND CLASS IN REQUEST                                         DEFIN
572   4          COMPUTE SIZE AS IDIM * JDIM * LENGTH OF TYPE                          DEFIN
573   4          IF SIZE IS TOO LARGE (>1200 WORDS) THEN                               DEFIN
574   5             CALL XZMSG TO DISPLAY ERROR "DATA ELEMENT IS TOO LARGE"            DEFIN
575   4          EXIT TO :PNERR1:                                                      DEFIN
576   4          ENDIF                                                                 DEFIN
577   3       ENDLOOP                                                                  DEFIN
578   4          CALL XZMSG TO DISPLAY ERROR "INVALID OR MISSING TYPE FIELD"           DEFIN
579   3       EXIT TO :PNERR1:                                                         DEFIN
580   3       ENDSEARCH                                                                DEFIN
581   3       INCREMENT TO NEXT TOKEN                                                  DEFIN
582   3       IF TOKEN IS NOT END OF TYPE FIELD SLASH OR                               DEFIN
583   4          NEXT TOKEN IS NOT A COMMA THEN                                        DEFIN
584   4          CALL XZMSG TO DISPLAY WARNING "TYPE NOT TERMINATED BY A SLASH"        DEFIN
585   3       ENDIF                                                                    DEFIN
```

```
586   3        BUILD REQUEST TO DELETE/ VERIFY ABSENT THIS DATA ELEMENT                    DEFIN
587   3        BUILD REQUEST TO ALLOCATE THIS DATA ELEMENT                                 DEFIN
588   2     ENDIF                                                                          DEFIN
589   1 EXIT XZDPN                                                                         DEFIN

590   2     :PNEPR1:                                                                       DEFIN
591   2     CALL XIMSG TO DISPLAY SYNTAX ERROR AND POSITION IN SYMBOLIC STRING             DEFIN
592   1 END XZDPN                                                                          DEFIN
```

```
594   1 CD************                                                                ENDSC
595   1 CD0                                                                          ENDSC
596   1 CD0       ENDSC IS AN FDS PROCESSOR SCHEDULED BY THE MANAGER                 ENDSC
597   1 CD0                                                                          ENDSC
598   1 CD************                                                              ENDSC
599   1 CD1                                                                          ENDSC
600   1 CD1       ENDSC PROCESSOR IS USED IN CONJUNCTION WITH SCAN ONLY.  IT         ENDSC
601   1 CD1       MARKS THE END OF THE SERIES OF PROCESSORS TO BE SCANNED.           ENDSC
602   1 CD1       IT HAS NO INTERFACE TABLE.                                         ENDSC
603   1 CD1                                                                          ENDSC
604   1 CD************                                                              ENDSC
605   1 CD2                                                                          ENDSC
606   1 CD2       INPUTS FROM THE MANAGER:                                           ENDSC
607   1 CD2                                                                          ENDSC
608   1 CD2             LU    - LOGICAL UNIT OF THE USER'S TERMINAL                  ENDSC
609   1 CD2             USEID - USER ID CODE                                         ENDSC
610   1 CD2             FLAGS - FLAGS FOR DEBUG                                      ENDSC
611   1 CD2                                                                          ENDSC
612   1 CD2       INPUTS FROM AWA:                                                   ENDSC
613   1 CD2                                                                          ENDSC
614   1 CD2             &SCNTB - SCAN CONTROL TABLE CREATED BY SCAN                  ENDSC
615   1 CD2                                                                          ENDSC
616   1 CD************                                                              ENDSC
617   1 CD3                                                                          ENDSC
618   1 CD3       OUTPUTS TO MANAGER:                                                ENDSC
619   1 CD3                                                                          ENDSC
620   1 CD3             XZRET - RETURN CODE TO MANAGER                               ENDSC
621   1 CD3                                                                          ENDSC
622   1 CD3       OUTPUTS TO AWA:                                                    ENDSC
623   1 CD3                                                                          ENDSC
624   1 CD3             XSCAN - NEW VALUE FOR X SCAN VARIABLE                        ENDSC
625   1 CD3             YSCAN - NEW VALUE FOR Y SCAN VARIABLE                        ENDSC
626   1 CD3                                                                          ENDSC
627   1 CD************                                                              ENDSC
628   1 CD5                                                                          ENDSC
629   1 CD5       RTE ROUTINES USED:                                                 ENDSC
630   1 CD5                                                                          ENDSC
631   1 CD5             CLOSE, EXEC,  FLOAT, KCVT,  POSNT,                           ENDSC
632   1 CD5             READF, RMPAR, WRITF                                          ENDSC
633   1 CD5                                                                          ENDSC
634   1 CD5       FDS ROUTINES USED:                                                 ENDSC
635   1 CD5                                                                          ENDSC
636   1 CD5             XPXIT, XREXT, XRMOV, XUDBG, XVPAW,                           ENDSC
637   1 CD5             XZFCL, XZMSG                                                 ENDSC
638   1 CD5                                                                          ENDSC
639   1 CD************                                                              ENDSC
```

```
641   1 BEGIN ENDSC                                                                                                                    ENDSC
642   2    CALL RMPAR TO GET INPUTS FROM MANAGER                                                                                       ENDSC
643   2    SAVE REQUEST TO RETRIEVE &SCNTB FROM AWS                                                                                    ENDSC
644   2    CALL XVPAW TO REQUEST AWA MANAGEMENT                                                                                        ENDSC
645   2    ERREXIT IF &SCNTB NOT FOUND TO :ERR4:                                                                                       ENDSC
646   2    CALL EXEC TO READ IN &SCNTB                                                                                                 ENDSC
647   2    SAVE REQUEST TO RETRIEVE SUMTAB IN REQBUF                                                                                   ENDSC
648   2    CALL XVPAW TO REQUEST AWA MANAGEMENT                                                                                        ENDSC
649   2    ERREXIT IF NOT FOUND TO :ERR4:                                                                                              ENDSC
650   2    CALL EXEC TO READ IN SUMTAB                                                                                                 ENDSC
651   2    IF SUMMARY TABLE IS LARGER THAN 32 ENTRIES THEN                                                                             ENDSC
652   3       SET SIZE OF SUMTAB TO 32 ENTRIES                                                                                         ENDSC
653   2    ENDIF                                                                                                                       ENDSC
654   2    CALL WRITF TO WRITE SUMTAB TO DATBOX                                                                                        ENDSC
655   2    ERREXIT IF WRITF ERROR TO :ERR4                                                                                             ENDSC
656   2    IF THERE IS 1 VARIABLE AND XCUR IS CENTROID OR                                                                              ENDSC
657   3       THERE ARE 2 VARIABLES AND XCUR IS CENTROID AND YCUR IS CENTROID THEN                                                      ENDSC
658   3       CALL READF TO READ HEADER RECORD                                                                                         ENDSC
659   3       ERREXIT IF READF ERROR TO :ERR4.                                                                                         ENDSC
660   3       UPDATE NUMBER OF SUMMARY TABLE ENTRIES                                                                                   ENDSC
661   3       CALL WRITF TO WRITE UPDATED HEADER                                                                                       ENDSC
662   3       ERREXIT IF WRITF ERROR TO :ERR4:                                                                                         ENDSC
663   3       CALL CLOSE TO CLOSE DATBOX                                                                                               ENDSC
664   3       SAVE REQUEST TO DEL/VER ABS &SCNTB IN REQBUF                                                                             ENDSC
665   3       IF THERE ARE REMAINING SCANS IN &SCNTB THEN                                                                              ENDSC
666   4          CALL EXEC TO WRITE REMAINING &SCNTB                                                                                   ENDSC
667   4          SAVE REQUEST TO ALLOC AND STORE VALUES FOR NEW &SCNTB                                                                 ENDSC
668   3       ENDIF                                                                                                                    ENDSC
669   3       SET RETURN PARAMETER TO NORMAL RETURN                                                                                    ENDSC
670   2    ELSE                                                                                                                        ENDSC
671   3       PERFORM SETXY                                                                                                            ENDSC
672   2    ENDIF                                                                                                                       ENDSC
673   2    CALL XVPAW TO REQUEST AWA MANAGEMENT                                                                                        ENDSC
674   2    CALL XPXIT TO TERMINATE WITH RETURN PARAMETERS                                                                              ENDSC
675   1 EXIT ENDSC                                                                                                                     ENDSC

676   2    :ERR4:                                                                                                                      ENDSC
677   2    CALL XZMSG TO DISPLAY ERROR                                                                                                 ENDSC
678   2    CALL XPXIT TO ABEND PROCESSOR                                                                                               ENDSC
679   1 END ENDSC                                                                                                                      ENDSC
```

```
681   1 BEGIN SETXY                                                                                                                ENDSC
682   2    IF XCUR IS END STEP THEN                                                                                                ENDSC
683   3       IF THERE IS 1 VARIABLE THEN                                                                                          ENDSC
684   4          SET XCUR TO ZERO                                                                                                  ENDSC
685   4          SET X TO CENTROID                                                                                                 ENDSC
686   4          CALL POSNT TO POSITION FILE TO CENTROID RECORD                                                                    ENDSC
687   4          ERREXIT IF POSNT ERROR TO :ERR4:                                                                                  ENDSC
688   3       ELSE                                                                                                                 ENDSC
689   4          IF YCUR IS END STEP THEN                                                                                          ENDSC
690   5             SET XCUR TO ZERO                                                                                               ENDSC
691   5             SET X TO CENTROID                                                                                              ENDSC
692   5             SET YCUR TO ZERO                                                                                               ENDSC
693   5             SET Y TO CENTROID                                                                                              ENDSC
694   5             CALL POSNT TO POSITION TO CENTROID RECORD                                                                      ENDSC
695   5             ERREXIT IF POSNT ERROR TO :ERR4:                                                                               ENDSC
696   4          ELSE                                                                                                              ENDSC
697   5             SET XCUR TO (-XSTEP)                                                                                           ENDSC
698   5             INCREMENT YCUR BY 1                                                                                            ENDSC
699   5             IF XSTEP IS ZERO AND Y IS CENTROID THEN                                                                        ENDSC
700   6                INCREMENT YCUR BY 1                                                                                         ENDSC
701   6                CALL WRITF TO WRITE DUMMY AS CENTROID RECORD                                                                ENDSC
702   6                ERREXIT IF WRITF ERROR TO :ERR4:                                                                            ENDSC
703   5             ENDIF                                                                                                          ENDSC
704   5             COMPUTE X AS (XCENT + XINCR * FLOAT (XCUR))                                                                    ENDSC
705   5             COMPUTE Y AS (YCENT + YINCR * FLOAT (YCUR))                                                                    ENDSC
706   4          ENDIF                                                                                                             ENDSC
707   3       ENDIF                                                                                                                ENDSC
708   2    ELSE                                                                                                                    ENDSC
709   3       INCREMENT XCUR BY 1                                                                                                  ENDSC
710   3       IF THERE IS 1 VARIABLE AND X IS THE CENTROID OR                                                                      ENDSC
711   4          THERE ARE 2 VARIABLES AND X IS THE CENTROID AND Y IS THE CENTROID THEN                                            ENDSC
712   4          INCREMENT XCUR BY 1                                                                                               ENDSC
713   4          CALL WRITF TO WRITE DUMMY AS CENTROID RECORD                                                                      ENDSC
714   4          ERREXIT IF WRITE ERROR TO :ERR4:                                                                                  ENDSC
715   3       ENDIF                                                                                                                ENDSC
716   3       COMPUTE X AS (XCENT + XINCR * FLOAT(XCUR))                                                                           ENDSC
717   3       IF THERE ARE 2 VARIABLES THEN                                                                                        ENDSC
718   4          COMPUTE Y AS (YCENT + YINCR * FLOAT(YCUR))                                                                        ENDSC
719   3       ENDIF                                                                                                                ENDSC
720   2    ENDIF                                                                                                                   ENDSC
721   2    DO FOR # SCAN VARIABLES                                                                                                 ENDSC
722   3       CALL EXEC TO WRITE VARIABLE                                                                                          ENDSC
723   3       SAVE REQUEST TO STORE VALUES FOR VARIABLE IN REQBUF                                                                  ENDSC
724   2    ENDDO                                                                                                                   ENDSC
725   2    CALL EXEC TO WRITE &SCNTB                                                                                               ENDSC
726   2    SAVE REQUEST TO STORE NEW &SCNTB                                                                                        ENDSC
727   2    SET RETURN PARAMETERS TO RESET SEQUENCE NUMBER                                                                          ENDSC
728   1 END SETXY
```

```
730   1 CD************                                                                                    SCAN
731   1 CD0                                                                                              SCAN
732   1 CD0       SCAN IS AN FDS PROCESSOR SCHEDULED BY THE MANAGER                                      SCAN
733   1 CD0                                                                                              SCAN
734   1 CD************                                                                                    SCAN
735   1 CD1                                                                                              SCAN
736   1 CD1       SCAN PROCESSOR EXECUTES A SERIES OF PROCESSORS ITERATING ON XSCAN AND                   SCAN
737   1 CD1       YSCAN (IF ENTERED) VALUES COMPUTED USING THE CENTROID, THE INCREMENT,                  SCAN
738   1 CD1       AND THE CURRENT STEP NUMBER, CREATING A DATA BOX FILE.                                 SCAN
739   1 CD1                                                                                              SCAN
740   1 CD************                                                                                    SCAN
741   1 CD2                                                                                              SCAN
742   1 CD2       INPUTS FROM MANAGER:                                                                   SCAN
743   1 CD2                                                                                              SCAN
744   1 CD2            LU    -LOGICAL UNIT OF THE USER'S TERMINAL                                        SCAN
745   1 CD2            USEID -USER ID CODE                                                               SCAN
746   1 CD2            FLAGS -FLAGS FOR DEBUG                                                            SCAN
747   1 CD2            ENTSUB-DISPLACEMENT OF THIS SCAN ENTRY IN &SEQTB                                  SCAN
748   1 CD2                                                                                              SCAN
749   1 CD2       INPUTS FROM INTERFACE TABLE:                                                           SCAN
750   1 CD2                                                                                              SCAN
751   1 CD2            PROCON-CARTRIDGE # FOR DATA BOX FILE                                              SCAN
752   1 CD2            SUMTAB-SUMMARY TABLE                                                              SCAN
753   1 CD2           *DATBOX-NAME OF DATA BOX ENTERED                                                   SCAN
754   1 CD2            NOVAR -NUMBER OF SCAN VARIABLES                                                   SCAN
755   1 CD2           *XSCAN -NAME OF X SCAN VARIABLE                                                    SCAN
756   1 CD2            XUNIT -UNIT OF X VARIABLE                                                         SCAN
757   1 CD2            XCENTR-CENTROID OF X VARIABLE                                                     SCAN
758   1 CD2            XINCR -INCREMENT FOR X                                                            SCAN
759   1 CD2            XSTEPS-NUMBER OF STEPS FOR X                                                      SCAN
760   1 CD2           *YSCAN -NAME OF Y SCAN VARIABLE                                                    SCAN
761   1 CD2            YUNIT -UNIT OF Y VARIABLE                                                         SCAN
762   1 CD2            YCENTR-CENTROID OF Y VARIABLE                                                     SCAN
763   1 CD2            YINCR -INCREMENT FOR Y                                                            SCAN
764   1 CD2            YSTEPS-NUMBER OF STEPS FOR Y                                                      SCAN
765   1 CD2                                                                                              SCAN
766   1 CD2           * - ACTUALLY OUTPUTS, ONLY NAME ENTERED IS AN INPUT TO SCAN                        SCAN
767   1 CD2                                                                                              SCAN
768   1 CD************                                                                                    SCAN
769   1 CD3                                                                                              SCAN
770   1 CD3       OUTPUTS TO MANAGER                                                                     SCAN
771   1 CD3                                                                                              SCAN
772   1 CD3            XZRET -RETURN CODE TO MANAGER                                                     SCAN
773   1 CD3                                                                                              SCAN
774   1 CD3       OUTPUTS TO WORK AREA (AWA):                                                            SCAN
775   1 CD3                                                                                              SCAN
776   1 CD3            DATBOX-FILE WHERE SUMMARY TABLE IS WRITTEN                                        SCAN
777   1 CD3            XSCAN -X SCAN VARIABLE                                                            SCAN
778   1 CD3            YSCAN -Y SCAN VARIABLE                                                            SCAN
779   1 CD3            &SCNTB-SCAN CONTROL TABLE (173 WORDS PER ACTIVE SCAN)                             SCAN
780   1 CD3                   (1)  SUMTAB 6 CHAR NAME OF SUMMARY TABLE                                   SCAN
781   1 CD3                   (4)  DISPL  DISPLACEMENT FOR SUMMARY TABLE                                 SCAN
782   1 CD3                   (5)  DATBOX 6 CHAR QUALIFIED NAME OF DATA BOX FILE                         SCAN
783   1 CD3                   (8)  RESET  RESET SEQUENCE NUMBER                                          SCAN
784   1 CD3                   (9)  CENREC CENTROID RECORD NUMBER                                         SCAN
785   1 CD3                   (10) XSCAN  NAME OF X VARIABLE                                             SCAN
786   1 CD3                   (13) XDISPL DISPLACEMENT FOR X                                             SCAN
787   1 CD3                   (14) XCENT  CENTROID FOR X                                                 SCAN
788   1 CD3                   (15) XINCR  INCREMENT FOR X                                                SCAN
```

```
789    1 CD3                  (18) XSTEP  # STEPS OF X                                  SCAN
790    1 CD3                  (19) XCUR   CURRENT X STEP NUMBER                         SCAN
791    1 CD3                  (20) YSCAN  NAME OF Y VARIABLE                            SCAN
792    1 CD3                  (23) YDISPL DISPLACEMENT FOR Y                            SCAN
793    1 CD3                  (24) YCENT  CENTROID OF Y                                 SCAN
794    1 CD3                  (26) YINCR  INCREMENT FOR Y                               SCAN
795    1 CD3                  (28) YSTEP  # STEPS OF Y                                  SCAN
796    1 CD3                  (29) YCUR   CURRENT Y STEP NUMBER                         SCAN
797    1 CD3                  (30) IDCB   144 WORD DCB FOR DATBOX                       SCAN
798    1 CD************                                                                 SCAN
799    1 CD5                                                                            SCAN
800    1 CD5       RTE SUBROUTINES USED:                                                SCAN
801    1 CD5                                                                            SCAN
802    1 CD5            CLOSE, CREAT, EXEC,  FLOAT, IAND,                               SCAN
803    1 CD5            KCVT,  MOD,   POSNT, PURGE, RMPAR,                              SCAN
804    1 CD5            WRITF                                                           SCAN
805    1 CD5                                                                            SCAN
806    1 CD5       FDS SUBROUTINES USED:                                                SCAN
807    1 CD5                                                                            SCAN
808    1 CD5            XPATR, XPGET, XPPUT, XPXIT, XRCPR,                              SCAN
809    1 CD5            XREXT, XRMOV, XRQFN, XRUNQ, XUDBG,                              SCAN
810    1 CD5            XVPAW, XZMSG                                                    SCAN
811    1 CD5                                                                            SCAN
812    1 CD************                                                                 SCAN
```

```
814    1 BEGIN SCAN                                                                                                    SCAN
815    2    CALL RMPAR TO RECEIVE INPUTS FROM MANAGER (LU, FLAGS, ENTRY DISPLACEMENT)                                  SCAN
816    2    CALL XPGET TO GET PROCON AND # SCAN VARIABLES                                                              SCAN
817    2    ERREXIT IF # SCAN VARIABLES <1 OR >2 TO :ERR3:                                                             SCAN
818    2    GET SUMMARY TABLE NAME AND DISPLACEMENT FROM INTERFACE TABLE                                               SCAN
819    2    ERREXIT IF SUMTAB IS A LITERAL TO :ERR3:                                                                   SCAN
820    2    ERREXIT IF DISPLACEMENT IS NOT AN ELEMENT BOUNDARY TO :ERR3:                                               SCAN
821    2    GET DATA BOX NAME FROM INTERFACE TABLE                                                                     SCAN
822    2    DO FOR # SCAN VARIABLES                                                                                    SCAN
823    3       CALL XPATR TO GET NAME AND DISPLACEMENT                                                                 SCAN
824    3       COMPUTE SUBSCRIPTS FROM DISPLACEMENT AND IDIM                                                           SCAN
825    3       CALL XPGET TO GET UNITS, CENTROID, INCR, # STEPS                                                        SCAN
826    3       ERREXIT IT # STEPS <0 OR >5 TO :ERR3:                                                                  SCAN
827    2    ENDDO                                                                                                      SCAN
828    2    SAVE REQUEST TO RETRIEVE VALUES FOR &SEQTB AND &SCNTB                                                      SCAN
829    2    CALL XVPAW TO REQUEST AWA MANAGEMENT                                                                       SCAN
830    2    CALL EXEC TO READ IN &SEQTB                                                                                SCAN
831    2    IF &SCNTB NOT FOUND THEN                                                                                   SCAN
832    3       SET # SCANS TO ZERO                                                                                     SCAN
833    2    ELSE                                                                                                       SCAN
834    3       SET # SCANS TO (TOTAL SIZE OF &SCNTB / SIZE OF ONE SCAN ENTRY)                                          SCAN
835    3       CALL EXEC TO READ IN &SCNTB                                                                             SCAN
836    3       ERREXIT IF # SCANS = MAXIMUM ALLOWED (4) TO :ERR3:                                                      SCAN
837    3       ERREXIT IF THIS DATBOX NAME IS ALREADY IN USE TO :ERR3:                                                 SCAN
838    2    ENDIF                                                                                                      SCAN
839    2    ERREXIT IF THIS IS THE LAST ENTRY IN &SEQTB TO :ERR3:                                                      SCAN
840    2    GET THE SEQUENCE NUMBER OF THIS SCAN FROM &SEQTB                                                           SCAN
841    2    IF THE DISPLACEMENT OF THIS SCAN IS ZERO THEN                                                              SCAN
842    3       SEARCH &SEQTB FOR THE SEQUENCE NUMBER                                                                   SCAN
843    3       ERREXIT IF THIS SCAN IS THE LAST ENTRY IN &SEQTB TO :ERR3:                                              SCAN
844    3       IF THIS IS A SEMI OVERRIDE (2 PROCESSOR NAMES NOT EQUAL) THEN                                           SCAN
845    4          SET RESET NUMBER TO THIS ENTRY SEQUENCE NUMBER                                                       SCAN
846    3       ELSE                                                                                                    SCAN
847    4          SET RESET NUMBER TO NEXT ENTRY SEQUENCE NUMBER                                                       SCAN
848    3       ENDIF                                                                                                   SCAN
849    2    ELSE                                                                                                       SCAN
850    3       SET RESET NUMBER TO NEXT ENTRY SEQUENCE NUMBER                                                          SCAN
851    2    ENDIF                                                                                                      SCAN
852    2    COMPUTE SIZE OF DATBOX FILE = (2+((2 * XSTEPS + 1)+(2 * YSTEPS + 1)+1) /2)                                 SCAN
853    2    COMPUTE CENTROID RECORD NUMBER = SIZE + 3                                                                  SCAN
854    2    DO FOR # SCAN VARIABLES                                                                                    SCAN
855    3       COMPUTE BEGINNING VALUE = (CENT + INCR * FLOAT (CUR STEP))                                              SCAN
856    2    ENDDO                                                                                                      SCAN
857    2    CALL XPPUT TO STORE DATBOX AND SCAN VARIABLE(S)                                                            SCAN
858    2    CALL CREAT TO CREAT DATBOX FILE                                                                            SCAN
859    2    IF FILE ALREADY EXISTS THEN                                                                                SCAN
860    3       CALL PURGE TO PURGE FILE                                                                                SCAN
861    3       ERREXIT IF PURGE ERROR TO :ERR2:                                                                        SCAN
862    3       CALL CREAT TO CREATE FILE                                                                               SCAN
863    2    ENDIF                                                                                                      SCAN
864    2    ERREXIT IF CREAT ERROR TO :ERR2:                                                                           SCAN
865    2    CALL WRITF TO WRITE HEADER RECORDS TO DATBOX                                                               SCAN
866    2    ERREXIT IF WRITF ERROR TO :ERR2:                                                                           SCAN
867    2    CALL POSNT TO POSITION FILE TO FIRST DATA RECORD                                                           SCAN
868    2    ERREXIT IF POSNT ERROR TO :ERR2:                                                                           SCAN
869    2    CALL EXEC TO WRITE &SCNTB                                                                                  SCAN
870    2    SAVE REQUEST TO DELETE/VERIFY ABSENT &SCNTB IN REQBUF                                                      SCAN
871    2    SAVE REQUESTS TO ALLOC AND STORE VALUES FOR NEW &SCNTB                                                     SCAN
872    2    CALL XVPAW TO REQUEST AWA MANAGEMENT                                                                       SCAN
```

```
873   2    ERREXIT IF NO AWA SPACE TO :ERR1:                                    SCAN
874   2    CALL XPXIT TO EXIT NORMALLY                                          SCAN
875   1 EXIT SCAN                                                               SCAN

876   2    :ERR1:                                                               SCAN
877   2    IF THERE ARE MORE ACTIVE SCANS (# SCANS > 0) THEN                    SCAN
878   3       CALL EXEC TO READ IN NEW &SCNTB                                   SCAN
879   3       CALL EXEC TO WRITE OUT ORIGINAL &SCNTB                            SCAN
880   3       SAVE REQUESTS TO ALLOC AND STORE VALUES FOR ORIGINAL &SCNTB       SCAN
881   2    ENDIF                                                                SCAN

882   2    :ERR2:                                                               SCAN
883   2    CALL CLOSE TO CLOSE DATBOX                                           SCAN
884   2    CALL PURGE TO PURGE DATBOX                                           SCAN
885   2    SET VALUE FOR XPPUT                                                  SCAN
886   2    SAVE REQUEST TO DELETE DATBOX FROM AWA                               SCAN
887   2    CALL XVPAW TO REQUEST AWA MANAGEMENT                                 SCAN

888   2    :ERR3:                                                               SCAN
889   2    CALL XZMSG TO DISPLAY ERROR                                          SCAN
890   2    CALL XPXIT TO ABEND SCAN                                             SCAN
891   1 END SCAN                                                                SCAN
```

```
893    1 CD0
894    1 CD0       FORTRAN CALLING PROCEDURE:
895    1 CD0
896    1 CD0       CALL XZCHR
897    1 CD0
898    1 CD********************
899    1 CD1
900    1 CD1       XZCHR IS USED BY THE ASSGM ROUTINE XZPS2 TO PROCESS DATA ASSIGNMENTS
901    1 CD1       FOR CHARACTER-TYPE OBJECT DATA ELEMENTS
902    1 CD1
903    1 CD********************
904    1 CD2
905    1 CD2       INPUTS FROM ASGCOM
906    1 CD2
907    1 CD2       LU,SYMTAB,SSTRNG,DATTPS,RSLTPT,CLSTRN,MAPWDS,RESULT
908    1 CD2
909    1 CD********************
910    1 CD3
911    1 CD3       OUTPUTS TO ASGCOM
912    1 CD3
913    1 CD3       REQST,NUMWDS,RSLTPT,OFRNDS
914    1 CD3
915    1 CD********************
916    1 CD5
917    1 CD5       EXTERNAL REFERENCES
918    1 CD5
919    1 CD5       FDS - XPREQ,XPXIT,XRMOV,XZLSS,XZMSG
920    1 CD5
921    1 CD5       RTE - IAND
922    1 CD5
923    1 CD********************
```

XZCHR

```
925   1 BEGIN XZCHR                                                                                                    XZCHR
926   2    SET # WORDS TO BE STORED IN OBJECT = # WORDS/ELEMENT FOR OBJECT                                             XZCHR
927   2 *                                          -3         -2         -1         1                                 XZCHR
928   2    CASE (RESULT DATA TYPE )    :CHRSTR:, :CHRFRE:, :CHRFRE:, :FIXERR:,                                        XZCHR
929   3 *                                    2          3                                                           XZCHR
930   3 *                                 :FIXERR:, :FIXERR:                                                         XZCHR

931   3       :CHRSTR:                                                                                               XZCHR
932   3       DETERMINE # WORDS IN CHARACTER STRING                                                                  XZCHR
933   3       MOVE CHARACTER STRING TO RESULT LOCATION                                                               XZCHR
934   3       IF # WORDS IN STRING < # WORDS/ELEMENT FOR OBJECT, THEN                                                 XZCHR
935   4          BLANK FILL AFTER CHARACTER STRING                                                                   XZCHR
936   3       ENDIF                                                                                                  XZCHR

937   3       :CHRFRE:                                                                                               XZCHR
938   3       IF TOP ENTRY ON RESULT STACK IS A DISPLACEMENT (TYPE = -2), THEN                                        XZCHR
939   4          POP DISPLACEMENT FROM RESULT STACK                                                                  XZCHR
940   3       ELSE                                                                                                   XZCHR
941   4          SET DISPLACEMENT = 0                                                                                XZCHR
942   3       ENDIF                                                                                                  XZCHR
943   3       POP RESULT OPERAND FROM STACK                                                                          XZCHR
944   3       IF RESULT OPERAND IS FREE, THEN                                                                        XZCHR
945   4          SET #WORDS TO BE RETRIEVED TO # WORDS/ELEMENT FOR OBJECT                                            XZCHR
946   3       ELSE  CHARACTER = CHARACTER                                                                            XZCHR
947   4          SET # WORDS TO BE RETRIEVED TO # WORDS/ELEMENT FOR OBJECT                                           XZCHR
948   3       ENDIF                                                                                                  XZCHR
949   3       CALL XPREQ TO RETRIEVE DATA FROM RESULT OPERAND AT DISPLACEMENT DETERMINED                             XZCHR
950   3       IF # WORDS RETRIEVED IS < # WORDS TO BE STORED, THEN                                                   XZCHR
951   4          BLANK FILL REMAINING DATA                                                                           XZCHR
952   3       ENDIF                                                                                                  XZCHR
953   2    ENDCASE                                                                                                   XZCHR
954   1 EXIT XZCHR                                                                                                   XZCHR

955   2    :FIXERR:                                                                                                  XZCHR
956   2    SET MESSAGE TO BE OUTPUT TO "CHARACTER DATA ELEMENT CANNOT BE SET EQUAL TO                                XZCHR
957   2 *                              NUMERICAL DATA"                                                               XZCHR
958   2    CALL XZMSG TO OUTPUT MESSAGE TO USER                                                                      XZCHR
959   2    CALL XZLSS TO LIST SYMBOLIC STRING                                                                        XZCHR
960   2    CALL XPXIT TO EXIT PROCESSER                                                                              XZCHR
961   1 END XZCHR                                                                                                    XZCHR
```

```
963   1 CD*******                                                                                                    XZDIN
964   1 CD0                                                                                                          XZDIN
965   1 CD0       XZDIN - DBDSP INPUT PROCESSOR                                                                      XZDIN
966   1 CD0                                                                                                          XZDIN
967   1 CD*******                                                                                                    XZDIN
968   1 CD1                                                                                                          XZDIN
969   1 CD1       XZDIN IS CALLED TO INTERPRET THE VARIOUS INTERFACE TABLE INPUTS                                     XZDIN
970   1 CD1       (MOSTLY SYMBOLIC STRINGS) AND BUILD DATA ARRAYS FROM THEM.                                         XZDIN
971   1 CD1                                                                                                          XZDIN
972   1 CD*******                                                                                                    XZDIN
973   1 CD2                                                                                                          XZDIN
974   1 CD2       INPUT                                                                                              XZDIN
975   1 CD2                                                                                                          XZDIN
976   1 CD2       ALL INPUT COMES FROM THE 26 INTERFACE TABLE ARGUMENTS                                              XZDIN
977   1 CD2                                                                                                          XZDIN
978   1 CD*******                                                                                                    XZDIN
979   1 CD3                                                                                                          XZDIN
980   1 CD3       OUTPUT                                                                                             XZDIN
981   1 CD3                                                                                                          XZDIN
982   1 CD3       COMMON                                                                                             XZDIN
983   1 CD3       NCVARL, NCRELL, CVALUE, NDVARL,NDVRUL, NC                                                          XZDIN
984   1 CD3                                                                                                          XZDIN
985   1 CD*******                                                                                                    XZDIN
986   1 CD4                                                                                                          XZDIN
987   1 CD4       NOTES                                                                                              XZDIN
988   1 CD4                                                                                                          XZDIN
989   1 CD4    USES ROUTINES                                                                                         XZDIN
990   1 CD4                                                                                                          XZDIN
991   1 CD4       EXEC                                                                                               XZDIN
992   1 CD4       XPGET                                                                                              XZDIN
993   1 CD4       XPXIT                                                                                              XZDIN
994   1 CD4       XRMOV                                                                                              XZDIN
995   1 CD4       XZLSS                                                                                              XZDIN
996   1 CD4                                                                                                          XZDIN
997   1 CD*******                                                                                                    XZDIN
```

```
1053    1 CD************                                                        XZDFT
1054    1 CD0                                                                  XZDFT
1055    1 CD0       FORTRAN CALLING SEQUENCE:                                  XZDFT
1056    1 CD0                                                                  XZDFT
1057    1 CD0             CALL XZDFT (ARRAY,START,END,TOKEN,INDEX)             XZDFT
1058    1 CD0                                                                  XZDFT
1059    1 CD************                                                        XZDFT
1060    1 CD1                                                                  XZDFT
1061    1 CD1       XZDFT SEARCHES 'ARRAY' BEGINNING AT 'START' AND ENDING AT 'END'  XZDFT
1062    1 CD1       FOR A SPECIFIED 'TOKEN'. IT RETURNS EITHER THE POSITION OF  XZDFT
1063    1 CD1       THE 'TOKEN' OR THE 'END' VALUE IN 'INDEX'.                 XZDFT
1064    1 CD1                                                                  XZDFT
1065    1 CD************                                                        XZDFT
1066    1 CD2                                                                  XZDFT
1067    1 CD2       INPUTS FROM CALLING SEQUENCE:                              XZDFT
1068    1 CD2                                                                  XZDFT
1069    1 CD2             ARRAY  - SYMBOLIC STRING TO BE SEARCHED              XZDFT
1070    1 CD2             END    - ENDING INDEX INTO 'ARRAY'                   XZDFT
1071    1 CD2             START  - BEGINNING INDEX INTO 'ARRAY'                XZDFT
1072    1 CD2             TOKEN  - TOKEN TO BE SEARCHED FOR                    XZDFT
1073    1 CD2                                                                  XZDFT
1074    1 CD************                                                        XZDFT
1075    1 CD3                                                                  XZDFT
1076    1 CD3       OUTPUTS TO CALLING SEQUENCE:                               XZDFT
1077    1 CD3                                                                  XZDFT
1078    1 CD3             INDEX  - POSITION OF 'TOKEN' IN SYMBOLIC STRING IF FOUND,  XZDFT
1079    1 CD3                      OR 'END' IF NOT FOUND                       XZDFT
1080    1 CD3                                                                  XZDFT
1081    1 CD************                                                        XZDFT
```

```
1083    1 BEGIN    XZDFT                                                    XZDFT
1084    2    SET INDEX TO START                                             XZDFT
1085    2    DO UNTIL INDEX .GE. END                                        XZDFT
1086    3       IF ARRAY(INDEX) .EQ. TOKEN, THEN                            XZDFT
1087    4          SET END TO INDEX                                         XZDFT
1088    3       ELSE                                                        XZDFT
1089    4          CASE ARRAY(INDEX)+1 (                                    XZDFT
1090    5             :ADD1:,        0  END OF STRING                       XZDFT
1091    5             :ADD2:,        1  INTEGER                             XZDFT
1092    5             :ADD3:,        2  REAL                                XZDFT
1093    5             :ADD4:,        3  DOUBLE                              XZDFT
1094    5             :ADD1:,        4  UNKNOWN                             XZDFT
1095    5             :ADD4:,        5  NAME                                XZDFT
1096    5             :ADD1:,        6  UNKNOWN                             XZDFT
1097    5             :ADD1:,        7  UNKNOWN                             XZDFT
1098    5             :CALST:,       8  CHARACTER STRING                    XZDFT
1099    5             :ADD1:,        9  UNKNOWN                             XZDFT
1100    5             :ADD1:,       10 +                                    XZDFT
1101    5             :ADD1:,       11 -                                    XZDFT
1102    5             :ADD1:,       12 *                                    XZDFT
1103    5             :ADD1:,       13 /                                    XZDFT
1104    5             :ADD1:,       14 <                                    XZDFT
1105    5             :ADD1:,       15 >                                    XZDFT
1106    5             :ADD1:,       16 #                                    XZDFT
1107    5             :ADD1:,       17 @                                    XZDFT
1108    5             :ADD1:,       18 =                                    XZDFT
1109    5             :ADD1:,       19 &                                    XZDFT
1110    5             :ADD1:,       20 ?                                    XZDFT
1111    5             :ADD1:,       21 (                                    XZDFT
1112    5             :ADD1:,       22 )                                    XZDFT
1113    5             :ADD2:,       23 START OF SYMBOLIC STRING             XZDFT
1114    5             :ADD1:,       24 END OF STRING                        XZDFT
1115    5             :ADD1:,       25 %                                    XZDFT
```

```
1116  5            :ADD1:,        26 BACK SLASH                                  XZDFT
1117  5            :ADD1:,        27 $                                           XZDFT
1118  5            :ADD1:,        28 .                                           XZDFT
1119  5            :ADD1:,        29 OPEN BRACKET                                XZDFT
1120  5            :ADD1:,        30 CLOSE BRACKET                               XZDFT
1121  5            :ADD2:,        31 REPEAT                                      XZDFT
1122  5            :ADD1:,        32 ,                                           XZDFT
1123  5            :ADD1: )       33 ;                                           XZDFT
1124  5            :ADD1:                                                        XZDFT
1125  5            INDEX=INDEX+1                                                 XZDFT
1126  5            :ADD2:                                                        XZDFT
1127  5            INDEX=INDEX+2                                                 XZDFT
1128  5            :ADD3:                                                        XZDFT
1129  5            INDEX=INDEX+3                                                 XZDFT
1130  5            :ADD4:                                                        XZDFT
1131  5            INDEX=INDEX+4                                                 XZDFT
1132  5            :CALST:                                                       XZDFT
1133  5            INDEX=INDEX+2+(ARRAY(INDEX+1)+1)/2                            XZDFT
1134  4          ENDCASE                                                         XZDFT
1135  3        ENDIF                                                             XZDFT
1136  2      ENDDO                                                               XZDFT
1137  1 END    XZDFT                                                             XZDFT
```

```
1139    1 CD *******
1140    1 CD 0
1141    1 CD 0
1142    1 CD 0      XZDP1 - DATA BOX DISPLAY OVERLAY - PREPARES DATA FOR DISPLAY
1143    1 CD 0
1144    1 CD 0           - SCHEDULED BY DBDSP
1145    1 CD 0
1146    1 CD *******
1147    1 CD 1
1148    1 CD 1
1149    1 CD 1      XZDP1 READS IN THE SCAN CONTROL INFORMATION AND THE SUMMARY
1150    1 CD 1      TABLES PRODUCED BY SCAN/ENDSCAN FOR FURTHER PROCESSING.
1151    1 CD 1      IN ADDITION, XZDP1 VERIFIES THE DISPLAY DEPENDANT VARIABLE
1152    1 CD 1      NAMES AND THE CONSTRAINT VARIABLE NAMES ARE CONTAINED IN THE
1153    1 CD 1      DEPENDANT VARIABLE NAME LIST GENERATED DURING THE SCAN.
1154    1 CD 1      XZDP1 ALSO GENERATES CONSTRAINT MASKS WITH SUBROUTINE XZDMK
1155    1 CD 1      FOR SUBSEQUENT DISPLAY BY XZDP2 AND XZDOT.
1156    1 CD 1
1157    1 CD *******
1158    1 CD 2
1159    1 CD 2
1160    1 CD 2
1161    1 CD 2   INPUTS FROM THE DATA BOX
1162    1 CD 2
1163    1 CD 2          RECORD 1
1164    1 CD 2              (1) - NAME OF FDS PROCESSOR CREATING FILE
1165    1 CD 2              (4) - INTERFACE TABLE VARIABLE NAME FOR THIS FILE
1166    1 CD 2              (7) - NAME OF FDS PROCESSOR UPDATING FILE
1167    1 CD 2                      ( 3 ASCII WORDS OF BLANKS )
1168    1 CD 2             (10) - INTERFACE TABLE VARIABLE NAME FOR THIS U-DATE
1169    1 CD 2                      ( 3 ASCII WORDS OF BLANKS )
1170    1 CD 2
1171    1 CD 2          RECORD 2
1172    1 CD 2                      (1N - NO OF ENTRIES IN SUMMARY TABLE
1173    1 CD 2                      (2) - X SCAN VARIABLE ( 6 CHAR)
1174    1 CD 2                      (5) - X FIRST SUBSCRIPT (INT OR ZERO)
1175    1 CD 2                      (6) - XSECOND SUBSCRIPT (INT OR ZERO)
1176    1 CD 2                      (7) -X UNITS ( 6 CHAR)
1177    1 CD 2                      (10) - X CENTROID (REAL)
1178    1 CD 2                      (12) - X INCREMENT ( REAL )
1179    1 CD 2                      (14) - X NUMBER OF STEPS (INTEGER 1-5)
1180    1 CD 2                      (15) - YSCAN VARIABLE ( 6 CHAR)
1181    1 CD 2                      (18) - Y FIRST SUBSCPT ( INT OR ZERO)
1182    1 CD 2                      (19) - Y SECOND SUBSCRIPT (INT OR ZERO)
1183    1 CD 2                      (20) - Y UNITS ( 6 CHAR )
1184    1 CD 2                      (23) - Y CENTROID (REAL)
1185    1 CD 2                      (25) -YX INCREMENT ( REAL )
1186    1 CD 2                      (27) - Y NUMBER OF STEPS (INTEGER 1-5)
1187    1 CD 2
1188    1 CD 2
1189    1 CD 2
1190    1 CD 2          DEPENDENT VARIABLE NAMES AND UNITS IN RECORDS 3, 4, & 5
1191    1 CD 2
1192    1 CD 2
1193    1 CD 2
1194    1 CD 2
1195    1 CD 2          SUMMARY TABLE RECORDS
1196    1 CD 2
1197    1 CD 2                - EACH SUMMARY TABLE CONTAIN VALUE FOR EACH
```

XZDP1 (repeated)

```
1198    1 CD 2                  DEPENDANT VARIABLE SCANNED (UP TO 32 VALUES                    XZDP1
1199    1 CD 2                  INCLUDING ERROR FLAG WHICH IS FIRST VALUE                      XZDP1
1200    1 CD 2                  IN SUMMARY TABLE)                                              XZDP1
1201    1 CD 2                                                                                 XZDP1
1202    1 CD *******                                                                           XZDP1
1203    1 CD 3                                                                                 XZDP1
1204    1 CD 3      OUTPUT FROM XZDP1                                                          XZDP1
1205    1 CD 3                                                                                 XZDP1
1206    1 CD 3                                                                                 XZDP1
1207    1 CD 3            MASK TABLES CONTAINING THE CONSTRAINT MASKS FOR EACH                 XZDP1
1208    1 CD 3             CONSTRAINT WHICH WAS VIOLATED AND INDICATION OF                     XZDP1
1209    1 CD 3             WHETHER OR NOT ANY CONSTRAINT WAS VIOLATED FOR EACH                 XZDP1
1210    1 CD 3             ARRAY COORDINATE                                                    XZDP1
1211    1 CD 3                                                                                 XZDP1
1212    1 CD 3            LIST OF VALID DISPLAY DEPENDANT VARIABLES FOR EACH                   XZDP1
1213    1 CD 3             PAGE (UP TO 16 PAGES)                                               XZDP1
1214    1 CD 3            LIST OF VALID CONSTRAINT VARIABLES WHICH WERE VIOLATED               XZDP1
1215    1 CD 3              (UP TO 8 CONSTRAINTS)                                              XZDP1
1216    1 CD 3                                                                                 XZDP1
1217    1 CD 3            DATA BOX DISPLAY ARRAY VALUES FOR ALL VARIABLES                      XZDP1
1218    1 CD 3                                                                                 XZDP1
1219    1 CD *******                                                                           XZDP1
1220    1 CD 4  DATBOX - NAME OF DATA BOX TO BE DISPLAYED BY DBDSP                             XZDP1
1221    1 CD 4  DATBUF - COMMON BUFFER FOR SCAN SUMMARY DESCRIPTORS                            XZDP1
1222    1 CD 4  IDV1   - POINTER TO DEP. DISPLAY VARIABLE NAME LIST FOR FIRST VAR              XZDP1
1223    1 CD 4  IDV2   - POINTER TO DEP. DISPLAY VARIABLE NAME LIST FOR 2ND VAR.               XZDP1
1224    1 CD 4  ISAVE  - LIST OF VIOLATED CONSTRAINTS BUILT BY XZMSK (MAX OF 8)               XZDP1
1225    1 CD 4  IXSCN1 - FIRST SUBSCRIPT FOR NAME OF X SCAN VARIABLE ( INT OR 0 )             XZDP1
1226    1 CD 4  IXSCN2 - SECOND SUBSCRIPT FOR NAME OF X SCAN VARIABLE (INT OR 0)              XZDP1
1227    1 CD 4  IFBUF  - NAME OF COMMON AREA USED FOR INTERFACE TABLE                          XZDP1
1228    1 CD 4  IYSCN1 - FIRST SUBSCRIPT FOR NAME OF Y SCAN VARIABLE (INT OR 0)              XZDP1
1229    1 CD 4  IYSCN2 - SECOND SUBSCRIPT FOR NAME OF Y SCAN VARIABLE (INT OR 0)             XZDP1
1230    1 CD 4      MASK1 - ARRAY CONTAINING CONSTRAINTS A THRU D                           XZDP1
1231    1 CD 4      MASK2 - ARRAY CONTAINING CONSTRAINTS E THRU G                           XZDP1
1232    1 CD 4  NAMVRL - NAME LIST FOR VARIABLES SCANNED BY SCAN/ENDSCN                     XZDP1
1233    1 CD 4  NC     - NUMBER OF CONSTRAINTS INPUT BY USER  (INTEGER)                     XZDP1
1234    1 CD 4  NCRELL - LIST OF CONSTRAINT RELATIONS INPUT BY USER                         XZDP1
1235    1 CD 4  NCVARL - LIST OF CONSTRAINT VARIABLE NAMES INPUT BY USER (32 MAX)           XZDP1
1236    1 CD 4  NDVARL - LIST OF DEP DISP VARIABLE PAIRS FOR PAGED OUTPUT(1-16PR)           XZDP1
1237    1 CD 4  NDVRCT - NUMBER OF DEP DISP VAR IN NDVARL LIST  (INTEGER)                   XZDP1
1238    1 CD 4  NDVRUL - LIST OF DEP DISP VARIABLE PAIR SCALE FACTORS FOR O/P               XZDP1
1239    1 CD 4  MSKERR - SET OF INDICATORS FOR CONSTRAINTS VIOLATED=0/NOT   0               XZDP1
1240    1 CD 4  NXSTEP - NUMBER OF STEPS ON EITHER SIDE OF X CENTROID  (0 TO 5)             XZDP1
1241    1 CD 4  NYSTEP - NUMBER OF STEPS ON EITHER SIDE OF Y CENTROID  (0 TO 5)             XZDP1
1242    1 CD 4  XCORD  - LIST OF X VAR VALUES FOR X COORDINATES  (1 - 11 REAL)              XZDP1
1243    1 CD 4  XSCNNM - NAME OF X VAR SCANNED TO BE PLACED ON DISPLAY (6 CHAR)             XZDP1
1244    1 CD 4  XUNITS - NAME OF X VAR UNITS TO BE PLACED ON DISPLAY   (6 CHAR)             XZDP1
1245    1 CD 4  YCORD  - LIST OF X VAR VALUES FOR Y COORDINATES  (1 - 11 REAL)              XZDP1
1246    1 CD 4  YSCNNM - NAME OF Y VAR SCANNED TO BE PLACED ON DISPLAY (6 CHAR)             XZDP1
1247    1 CD 4  YUNITS - NAME OF Y VAR UNITS TO BE PLACED ON DISPLAY   (6 CHAR)             XZDP1
1248    1 CD 4  ZTABLE - TABLE IN COMMON FOR SUMTAB VARIABLE NAMES AND UNITS                XZDP1
1249    1 CD 4  NAMVUL - UNITS LIST FOR VARIABLES SCANNED BY SCAN/ENDSCN                    XZDP1
1250    1 CD 4  SUMTAB - VALUES FOR SCAN VARIABLE(S) - 1 TO 32 VALUES/RECORD                XZDP1
1251    1 CD 4  PARMS  - COMMUNICATION BUFFER FOR RMPAR - LU, USER ID, FLAGS                XZDP1
1252    1 CD 4  LU     - LOGICAL UNIT # FOR XPRDM CALLING SEQUENCE - USER LOCATN            XZDP1
1253    1 CD 4  LUDSP  - DBDSP WILL OUTPUT DISPLAY TO THIS USER SUPPLIED LU                 XZDP1
1254    1 CD 4  PROMPT - TABLE IN COMMON TO COMMUNICATE WITH XPRDM                          XZDP1
1255    1 CD 4  DEBUG                                                                       XZDP1
1256    1 CD 4  SELECT - SELECT =0 PROMPT ; SELECT NOT 0 RUN ALL DISPLAYS TO O/P            XZDP1
```

```
1257    1 CD 4            WITHOUT PROMPT                                                    XZDP1
1258    1 CD 4  CARTRG - CARTRAGE USED TO LOCATE DATA BOX                                   XZDP1
1259    1 CD 4       CVALUE - NUMERIC (REAL) VALUES FOR EACH CONSTRAINT (UP TO 8)           XZDP1
1260    1 CD 4       IDCB   - DCB BLOCK FOR DRDE READ                                       XZDP1
1261    1 CD *******                                                                         XZDP1
1262    1 CD 5                                                                               XZDP1
1263    1 CD 5       USES ROUTINES                                                           XZDP1
1264    1 CD 5                                                                               XZDP1
1265    1 CD 5       XPRDM, XZDMK, EXEC, OPEN, READF                                         XZDP1
1266    1 CD 5       XCMOV, XRCPR, XRI6                                                      XZDP1
1267    1 CD 5                                                                               XZDP1
1268    1 CD 5                                                                               XZDP1
1269    1 CD *******                                                                         XZDP1
```

```
1306    ; CD*******                                                                    XZDMK
1307    1 CD0                                                                          XZDMK
1308    1 CD0       XZDMK - CONSRAINT MASKING ROUTINE                                  XZDMK
1309    1 CD0                                                                          XZDMK
1310    1 CD*******                                                                    XZDMK
1311    1 CD1                                                                          XZDMK
1312    1 CD1       XZDMK IS CALLED ONCE BY XZDP1 TO BUILD THE CONSTRAINT              XZDMK
1313    1 CD1       VIOLATION MASKS FOR ALL VALUE POSITIONS OF THE DISPLAY GRID.       XZDMK
1314    1 CD1                                                                          XZDMK
1315    1 CD*******                                                                    XZDMK
1316    1 CD2                                                                          XZDMK
1317    1 CD2    INPUTS                                                                XZDMK
1318    1 CD2                                                                          XZDMK
1319    1 CD2       COMMON - ATABLE, NCVARL, NCRELL, CVALUE                            XZDMK
1320    1 CD2                                                                          XZDMK
1321    1 CD*******                                                                    XZDMK
1322    1 CD3                                                                          XZDMK
1323    1 CD3    OUTPUTS                                                               XZDMK
1324    1 CD3                                                                          XZDMK
1325    1 CD3       COMMON - MSKERR, MASK1, MASK2                                      XZDMK
1326    1 CD3                ISAVE                                                     XZDMK
1327    1 CD3                                                                          XZDMK
1328    1 CD*******                                                                    XZDMK
1329    1 CD4                                                                          XZDMK
1330    1 CD4       NOTES                                                              XZDMK
1331    1 CD4                                                                          XZDMK
1332    1 CD4    USES ROUTINES                                                         XZDMK
1333    1 CD4                                                                          XZDMK
1334    1 CD4       XRCFR                                                              XZDMK
1335    1 CD4       XRMOV                                                              XZDMK
1336    1 CD4       XRSET                                                              XZDMK
1337    1 CD4                                                                          XZDMK
1338    1 CD*******                                                                    XZDMK
```

```
1402    1 CD *******                                                                    XZDP2
1403    1 CD 0                                                                          XZDP2
1404    1 CD 0      XZDP2  - DATA BOX DISPLAY OVERLAY - PROMPTS USER FOR DISPLAY        XZDP2
1405    1 CD 0               DESIRED, THEN FORMATS AND DISPLAYS DATA ACCORDINGLY        XZDP2
1406    1 CD 0                                                                          XZDP2
1407    1 CD 0             - SCHEDULED BY DBDSP                                         XZDP2
1408    1 CD 0                                                                          XZDP2
1409    1 CD *******                                                                    XZDP2
1410    1 CD 1                                                                          XZDP2
1411    1 CD 1     XZDP2 ISSUES PROMPTS TO THE USER REQUESTING PAGE NUMBER,             XZDP2
1412    1 CD 1             (STARTING ROW NUMBER, NUMBER OF ROWS, AND DESIRED          XZDP2
1413    1 CD 1              LU.)  SELECT OPTION CAN BE SET  WHEN ENTERING              XZDP2
1414    1 CD 1              DBDSP IN THE INTERFACE TABLE TO OUTPUT ALL PAGES           XZDP2
1415    1 CD 1              TO NAMED LU.                                               XZDP2
1416    1 CD 1              XZDP2 UTILIZES SUBROUTINE XZDOT TO ACTUALLY FORMAT         XZDP2
1417    1 CD 1              AND DISPLAY THE SCAN RESULTS.                              XZDP2
1418    1 CD 1                                                                         XZDP2
1419    1 CD *******                                                                   XZDP2
1420    1 CD 2                                                                         XZDP2
1421    1 CD 2      INPUTS TO XZDP2 FROM XZDP1                                         XZDP2
1422    1 CD 2                                                                         XZDP2
1423    1 CD 2             MASK TABLES CONTAINING THE CONSTRAINT MASKS FOR EACH        XZDP2
1424    1 CD 2              CONSTRAINT WHICH WAS VIOLATED AND INDICATION OF            XZDP2
1425    1 CD 2              WHETHER OR NOT ANY CONSTRAINT WAS VIOLATED FOR EACH        XZDP2
1426    1 CD 2              ARRAY COORDINATE                                           XZDP2
1427    1 CD 2                                                                         XZDP2
1428    1 CD 2             LIST OF VALID DISPLAY DEPENDANT VARIABLES FOR EACH          XZDP2
1429    1 CD 2              PAGE (UP TO 16 PAGES)                                      XZDP2
1430    1 CD 2                                                                         XZDP2
1431    1 CD 2             DATA BOX DISPLAY ARRAY VALUES FOR ALL VARIABLES             XZDP2
1432    1 CD 2                                                                         XZDP2
1433    1 CD 2                                                                         XZDP2
1434    1 CD 2                                                                         XZDP2
1435    1 CD *******                                                                   XZDP2
1436    1 CD 3                                                                         XZDP2
1437    1 CD 3      OUTPUT FROM XZDP2                                                  XZDP2
1438    1 CD 3                                                                         XZDP2
1439    1 CD 3              DISPLAY FORMAT SHOWN IN DOCUMENTATION IS SENT TO           XZDP2
1440    1 CD 3              NAME LU DEVISE                                             XZDP2
1441    1 CD 5                                                                         XZDP2
1442    1 CD *******                                                                   XZDP2
1443    1 CD 4  COMBUF - BUFFER TO RECEIVE ENCODED USER RESPONSE - XPROM               XZDP2
1444    1 CD 4  DATBOX - NAME OF DATA BOX TO BE DISPLAYED BY DBDSP                     XZDP2
1445    1 CD 4  DATBUF - COMMON BUFFER FOR SCAN SUMMARY DESCRIPTORS                    XZDP2
1446    1 CD 4  IDV1   - POINTER TO DEP. DISPLAY VARIABLE NAME LIST FOR FIRST VAR      XZDP2
1447    1 CD 4  IDV2   - POINTER TO DEP. DISPLAY VARIABLE NAME LIST FOR 2ND VAR.       XZDP2
1448    1 CD 4  ISAVE  - LIST OF VIOLATED CONSTRAINTS BUILT BY XZMSK (MAX OF 8)        XZDP2
1449    1 CD 4  IXSCN1 - FIRST SUBSCRIPT FOR NAME OF X SCAN VARIABLE ( INT OR 0 )      XZDP2
1450    1 CD 4  IXSCN2 - SECOND SUBSCRIPT FOR NAME OF X SCAN VARIABLE (INT OR 0)       XZDP2
1451    1 CD 4  IFBUF  - NAME OF COMMON AREA USED FOR INTERFACE TABLE                  XZDP2
1452    1 CD 4  IYSCN1 - FIRST SUBSCRIPT FOR NAME OF Y SCAN VARIABLE (INT OR 0)        XZDP2
1453    1 CD 4  IYSCN2 - SECOND SUBSCRIPT FOR NAME OF Y SCAN VARIABLE (INT OR 0)       XZDP2
1454    1 CD 4  LENBF  - LENGTH OF COMBUF IN 16-BIT WORDS - XPRDM                      XZDP2
1455    1 CD 4  LENST  - LENGTH IN WDS OF CHAR STRING USED FOR USER PROMPT -XPRDM      XZDP2
1456    1 CD 4  MASK   - ARRAY CONTAINING MASKS FOR UP TO 8 CONSTR( 4 X 121 SIZE)     XZDP2
1457    1 CD 4  NAMVRL - NAME LIST FOR VARIABLES SCANNED BY SCAN/ENDSCN                XZDP2
1458    1 CD 4  NC     - NUMBER OF CONSTRAINTS INPUT BY USER  (INTEGER)                XZDP2
1459    1 CD 4  NCRELL - LIST OF CONSTRAINT RELATIONS INPUT BY USER                    XZDP2
1460    1 CD 4  NCVARL - LIST OF CONSTRAINT VARIABLE NAMES INPUT BY USER (32 MAX)      XZDP2
```

```
1461    1 CD 4  NDVARL - LIST OF DEP DISP VARIABLE PAIRS FOR PAGED OUTPUT(1-16PR)                    XZDP2
1462    1 CD 4  NDVRCT - NUMBER OF DEP DISP VAR IN NDVARL LIST  (INTEGER)                           XZDP2
1463    1 CD 4  NDVRUL - LIST OF DEP DISP VARIABLE PAIR SCALE FACTORS FOR O/P                       XZDP2
1464    1 CD 4  MSKERR - SET OF INDICATORS FOR CONSTRAINTS VIOLATED=0/NOT   0                       XZDP2
1465    1 CD 4  NXSTEP - NUMBER OF STEPS ON EITHER SIDE OF X CENTROID  (0 TO 5)                     XZDP2
1466    1 CD 4  NYSTEP - NUMBER OF STEPS ON EITHER SIDE OF Y CENTROID  (0 TO 5)                     XZDP2
1467    1 CD 4  RETC   - RTN CODE FM XPRDMX; 0=NRML,1=RTN TO EXEC,2=NULL BFR,3=ERR                  XZDP2
1468    1 CD 4  STRING - CHAR STRING CONTAINING USER PROMPT MESSAGE                                 XZDP2
1469    1 CD 4  XCORD  - LIST OF X VAR VALUES FOR X COORDINATES  (1 - 11 REAL)                      XZDP2
1470    1 CD 4  XSCNNM - NAME OF X VAR SCANNED TO BE PLACED ON DISPLAY (6 CHAR)                     XZDP2
1471    1 CD 4  XUNITS - NAME OF X VAR UNITS TO BE PLACED ON DISPLAY   (6 CHAR)                    XZDP2
1472    1 CD 4  YCORD  - LIST OF X VAR VALUES FOR Y COORDINATES  (1 - 11 REAL)                      XZDP2
1473    1 CD 4  YSCNNM - NAME OF Y VAR SCANNED TO BE PLACED ON DISPLAY (6 CHAR)                     XZDP2
1474    1 CD 4  YUNITS - NAME OF Y VAR UNITS TO BE PLACED ON DISPLAY   (6 CHAR)                    XZDP2
1475    1 CD 4  ZTABLE - TABLE IN COMMON FOR SUMTAB VARIABLE NAMES AND UNITS                        XZDP2
1476    1 CD 4  NAMVUL - UNITS LIST FOR VARIABLES SCANNED BY SCAN/ENDSCN                            XZDP2
1477    1 CD 4  SUMTAB - VALUES FOR SCAN VARIABLE(S) - 1 TO 32 VALUES/RECORD                        XZDP2
1478    1 CD 4  PARMS  - COMMUNICATION BUFFER FOR RMPAR - LU, USER ID, FLAGS                        XZDP2
1479    1 CD 4  LU     - LOGICAL UNIT # FOR XPRDM CALLING SEQUENCE - USER LOCATN                    XZDP2
1480    1 CD 4  LUDSP  - DBDSP WILL OUTPUT DISPLAY TO THIS USER SUPPLIED LU                         XZDP2
1481    1 CD 4  PROMPT - TABLE IN COMMON TO COMMUNICATE WITH XPRDM                                  XZDP2
1482    1 CD 4  SELECT - SELECT =0 PROMPT ; SELECT NOT 0 RUN ALL DISPLAYS TO O/P                    XZDP2
1483    1 CD 4           WITHOUT PROMPT                                                             XZDP2
1484    1 CD 4  CARTRG - CARTRAGE USED TO LOCATE DATA BOX                                           XZDP2
1485    1 CD *******                                                                                XZDP2
1486    1 CD 5                                                                                      XZDP2
1487    1 CD 5      USES ROUTINES                                                                   XZDP2
1488    1 CD 5                                                                                      XZDP2
1489    1 CD 5        XPRDM, EXEC, XZDOT                                                            XZDP2
1490    1 CD 5                                                                                      XZDP2
1491    1 CD 5                                                                                      XZDP2
1492    1 CD *******                                                                                XZDP2
```

```
1531    1 CD*******
1532    1 CD0
1533    1 CD0       XZDOT - DISPLAY OUTPUT ROUTINE
1534    1 CD0
1535    1 CD*******
1536    1 CD1
1537    1 CD1       XZDOT IS CALLED ONCE BY XZDP2 TO OUTPUT THE DISPLAY PAGE
1538    1 CD1
1539    1 CD*******
1540    1 CD2
1541    1 CD2    INPUTS
1542    1 CD2
1543    1 CD2       COMMON - NCVARL, NDVARL, NCRELL, CVALUE, K1, K2, ZTABLE
1544    1 CD2                ATABLE, NDVURL, NPAGE, DTBX, XSCNNM, YSCNNM
1545    1 CD2                XUNITS, YUNITS, IXSCN1, IXSCN2, IYSCN1, IYSCN2
1546    1 CD2
1547    1 CD*******
1548    1 CD3
1549    1 CD3    OUTPUTS
1550    1 CD3
1551    1 CD3       NONE
1552    1 CD3
1553    1 CD*******
1554    1 CD4
1555    1 CD4    NOTES
1556    1 CD4
1557    1 CD4       USES ROUTINES
1558    1 CD4
1559    1 CD4         XRMOV
1560    1 CD4         EXEC
1561    1 CD4         XZ1SP
1562    1 CD4         XRPCK
1563    1 CD4         XRUPK
1564    1 CD4         XRI6
1565    1 CD4         XRE16
1566    1 CD4         IABS
1567    1 CD4
1568    1 CD*******
```

```
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
XZDOT
```

```
1591    1 CD*******                                                                    XZ1SP
1592    1 CDO                                                                          XZ1SP
1593    1 CDO       FORTRAN CALLING PROCEEDURE                                         XZ1SP
1594    1 CDO    CALL XZ1SP (STRING, LEN)                                              XZ1SP
1595    1 CDO                                                                          XZ1SP
1596    1 CD*******                                                                    XZ1SP
1597    1 CD1                                                                          XZ1SP
1598    1 CD1       XZ1SP REMOVES DUPLICATE (I.E. CONSECUTIVE) BLANKS FROM             XZ1SP
1599    1 CD1       A CHARACTER STRING AND FILLS THE VACATED TRAILING WORDS            XZ1SP
1600    1 CD1       WITH BLANKS                                                        XZ1SP
1601    1 CD1                                                                          XZ1SP
1602    1 CD*******                                                                    XZ1SP
1603    1 CD2                                                                          XZ1SP
1604    1 CD2       INPUT                                                              XZ1SP
1605    1 CD2                                                                          XZ1SP
1606    1 CD2       CALLING SEQUENCE                                                   XZ1SP
1607    1 CD2                                                                          XZ1SP
1608    1 CD2         STRING - INPUT CHARACTER STRING                                  XZ1SP
1609    1 CD2         LEN    - NUMBER OF WORDS IN STRING                               XZ1SP
1610    1 CD2                                                                          XZ1SP
1611    1 CD*******                                                                    XZ1SP
1612    1 CD3                                                                          XZ1SP
1613    1 CD3       OUTPUT                                                             XZ1SP
1614    1 CD3                                                                          XZ1SP
1615    1 CD3       CALLING SEQUENCE                                                   XZ1SP
1616    1 CD3                                                                          XZ1SP
1617    1 CD3       STRING   - CHARACTER STRING WITH ALL FIELDS OF CONSECUTIVE         XZ1SP
1618    1 CD3                  BLANKS REDUCED TO 1 BLANK AND TRAILING BLANK FILL      XZ1SP
1619    1 CD3       LEN      - NO. OF WORDS IN STRING PRIOR TO TRAILING BLANK FILL     XZ1SP
1620    1 CD3                                                                          XZ1SP
1621    1 CD*******                                                                    XZ1SP
1622    1 CD4                                                                          XZ1SP
1623    1 CD4       NOTES                                                              XZ1SP
1624    1 CD4                                                                          XZ1SP
1625    1 CD4    USES ROUTINES                                                         XZ1SP
1626    1 CD4                                                                          XZ1SP
1627    1 CD4       XR1SP                                                              XZ1SP
1628    1 CD4                                                                          XZ1SP
1629    1 CD*******                                                                    XZ1SP
1630    1 BEGIN XZ1SP                                                                  XZ1SP
1631    2    CALL XR1SP TO REMOVE DUPLICATE BLANKS FROM STRING                         XZ1SP
1632    2    DO WHILE THERE ARE TRAILING WORDS IN STRING                               XZ1SP
1633    3       SET THIS TRAILING WORD TO BLANKS                                       XZ1SP
1634    2    ENDDO                                                                     XZ1SP
1635    1 END XZ1SP                                                                    XZ1SP
```

```
1637    1 CD*************                                                    XZFCL
1638    1 CD0                                                               XZFCL
1639    1 CD0       FORTRAN CALLING SEQUENCE:                               XZFCL
1640    1 CD0                                                               XZFCL
1641    1 CD0             CALL XZFCL (LU)                                   XZFCL
1642    1 CD0                                                               XZFCL
1643    1 CD*************                                                   XZFCL
1644    1 CD1                                                               XZFCL
1645    1 CD1       XZFCL SEARCHES XVSTB LOOKING FOR LU SO THAT THE         XZFCL
1646    1 CD1       PROCESSOR'S CLASS NUMBER CAN BE EXTRACTED. IT           XZFCL
1647    1 CD1       SAVES THE CLASS NUMBER IN EXTERNAL XPCLS.               XZFCL
1648    1 CD1                                                               XZFCL
1649    1 CD*************                                                   XZFCL
1650    1 CD2                                                               XZFCL
1651    1 CD2       INPUTS FROM CALLING SEQUENCE:                           XZFCL
1652    1 CD2                                                               XZFCL
1653    1 CD2             LU - LOGICAL UNIT OF TERMINAL BEING USED.         XZFCL
1654    1 CD2                                                               XZFCL
1655    1 CD*************                                                   XZFCL
1656    1 CD5                                                               XZFCL
1657    1 CD5       NOTE: CALLING PROGRAM MUST HAVE XPCLS DEFINED           XZFCL
1658    1 CD5             AS AN EXTERNAL REFERENCE.                         XZFCL
1659    1 CD5                                                               XZFCL
1660    1 CD*************                                                   XZFCL
1661    1 *                                                                 XZFCL
1662    1 *                                                                 XZFCL
1663    1 *                                                                 XZFCL
1664    1 *                                                                 XZFCL
1665    1 BEGIN XZFCL                                                       XZFCL
1666    2    CALL .ENTR TO RESOLVE PARAMETER ADDRESS                        XZFCL
1667    2    STARTSEARCH FOR ALL XVSTB ENTRIES                              XZFCL
1668    2    EXIT IF THIS ENTRY'S LU IS MINE                                XZFCL
1669    2    ENDSEARCH                                                      XZFCL
1670    2    GET CLASS NUMBER FROM XVSTB                                    XZFCL
1671    2    SAVE CLASS NUMBER IN XPCLS                                     XZFCL
1672    1 END XZFCL                                                         XZFCL
```

```
1674    1 CD0                                                                                                                  XZFNC
1675    1 CD0       FORTRAN CALLING PROCEDURE:                                                                                 XZFNC
1676    1 CD0                                                                                                                  XZFNC
1677    1 CD0       CALL XZFNC(ENTRY)                                                                                          XZFNC
1678    1 CD0                                                                                                                  XZFNC
1679    1 CD********************                                                                                               XZFNC
1680    1 CD1                                                                                                                  XZFNC
1681    1 CD1       XZFNC IS USED BY THE ASSGN ROUTINE XZPS2 TO EVALUATE FUNCTION                                              XZFNC
1682    1 CD1       OPERATIONS                                                                                                 XZFNC
1683    1 CD1                                                                                                                  XZFNC
1684    1 CD********************                                                                                               XZFNC
1685    1 CD2                                                                                                                  XZFNC
1686    1 CD2       INPUTS                                                                                                     XZFNC
1687    1 CD2                                                                                                                  XZFNC
1688    1 CD2       ENTRY - FUNCTION TOKEN CURRENTLY BEING PROCESSED                                                           XZFNC
1689    1 CD2                                                                                                                  XZFNC
1690    1 CD2       FROM ASGCOM - LU,SSTRNG,OPRNDS,OPINFO                                                                      XZFNC
1691    1 CD2                                                                                                                  XZFNC
1692    1 CD********************                                                                                               XZFNC
1693    1 CD3                                                                                                                  XZFNC
1694    1 CD3       OUTPUTS TO ASGCOM                                                                                          XZFNC
1695    1 CD3                                                                                                                  XZFNC
1696    1 CD3       RESULT,RSLTPT,OPRNDS                                                                                       XZFNC
1697    1 CD3                                                                                                                  XZFNC
1698    1 CD********************                                                                                               XZFNC
1699    1 CD5                                                                                                                  XZFNC
1700    1 CD5       EXTERNAL REFERENCES                                                                                        XZFNC
1701    1 CD5                                                                                                                  XZFNC
1702    1 CD5       FDS - XPXIT, XRMOV, XZLSS, XZMSG                                                                           XZFNC
1703    1 CD5                                                                                                                  XZFNC
1704    1 CD5       RTE - ABS, AINT, ALOG, ALOGT, AMOD, ATAN, ATAN2, COS, DABS, DATAN,                                         XZFNC
1705    1 CD5       DATN2, DBLE, DCOS, DDINT, DEXP, DLOG, DLOGT, DMOD, DSIGN, DSIN, DSQRT,                                     XZFNC
1706    1 CD5       EXP, FLOAT, IABS, IDINT, IFIX, ISIGN, MOD, OVF, SIGN, SIN, SNGLE, SQRT,                                    XZFNC
1707    1 CD5       TAN, TANH                                                                                                  XZFNC
1708    1 CD5                                                                                                                  XZFNC
1709    1 CD********************                                                                                               XZFNC
```

```
1711  1 BEGIN XIFNC                                                    XIFNC
1712  2   PERFORM FUNCTION INDICATED BY ENTRY                          XIFNC
1713  2.2   ERREXIT TO :OVER: IF OVERFLOW OR UNDERFLOW IS INDICATED   XIFNC
1714  2   PUSH RESULT AND TYPE ONTO RESULT STACK                       XIFNC
1715  1 EXIT XIFNC                                                     XIFNC

1716  2   :OVER:                                                       XIFNC
1717  2   SET MESSAGE TO BE OUTPUT TO "OVERFLOW OR UNDERFLOW DETECTED" XIFNC
1718  2   CALL XZMSG TO OUTPUT MESSAGE TO USER                         XIFNC
1719  2   CALL XZLSS TO LIST SYMBOLIC STRING                           XIFNC
1720  2   CALL XPXIT TO EXIT PROCESSER                                 XIFNC
1721  1 END XIFNC                                                      XIFNC
```

```
1723    1 CDO                                                                   XZFRE
1724    1 CDO       FORTRAN CALLING PROCEDURE:                                  XZFRE
1725    1 CDO                                                                   XZFRE
1726    1 CDO       CALL XZFRE                                                  XZFRE
1727    1 CDO                                                                   XZFRE
1728    1 CD********************                                               XZFRE
1729    1 CD1                                                                   XZFRE
1730    1 CD1       XZFRE IS USED BY THE ASSGN ROUTINE XZPS2 TO PROCESS DATA ASSIGNMENTS  XZFRE
1731    1 CD1       FOR FREE-TYPE OBJECT DATA ELEMENTS                          XZFRE
1732    1 CD1                                                                   XZFRE
1733    1 CD********************                                               XZFRE
1734    1 CD2                                                                   XZFRE
1735    1 CD2       INPUTS FROM ASGCOM                                          XZFRE
1736    1 CD2                                                                   XZFRE
1737    1 CD2       SYMTAB,SSTRNS,RESULT,RSLTPT,CLSTRN,MAPWDS                   XZFRE
1738    1 CD2                                                                   XZFRE
1739    1 CD********************                                               XZFRE
1740    1 CD3                                                                   XZFRE
1741    1 CD3       OUTPUTS TO ASGCOM                                           XZFRE
1742    1 CD3                                                                   XZFRE
1743    1 CD3       CPRNDS,REQST,NUMWDS,RSLTPT                                  XZFRE
1744    1 CD3                                                                   XZFRE
1745    1 CD********************                                               XZFRE
1746    1 CD5                                                                   XZFRE
1747    1 CD5       EXTERNAL REFERENCES                                         XZFRE
1748    1 CD5                                                                   XZFRE
1749    1 CD5       FDS - XPREQ,XRMOV,XZPCS                                     XZFRE
1750    1 CD5                                                                   XZFRE
1751    1 CD5       RTE - IAND                                                  XZFRE
1752    1 CD5                                                                   XZFRE
1753    1 CD********************                                               XZFRE
```

```
1 BEGIN XZFRE                                                                XZFRE
2 *                                     -3         -2         -1         1   XZFRE
2    CASE  (RESULT DATA TYPE )  :FRESTR:, :FREFRE:, :FREFRE:, :FREFIX:,       XZFRE
3 *                                      2          3                         XZFRE
3 *                             :FREFIX:, :FREFIX:,                           XZFRE

3       :FRESTR:                                                             XZFRE
3       SET # WORDS TO BE STORED = LENGTH OF CHARACTER STRING IN WORDS       XZFRE
3       MOVE CHARACTER STRING TO RESULT LOCATION                             XZFRE

3       :FREFRE:                                                             XZFRE
3       IF TOP ENTRY OF RESULT STACK IS A DISPLACEMENT (TYPE = -2), THEN     XZFRE
4          POP DISPLACEMENT FROM RESULT STACK                                XZFRE
3       ELSE  FREE OR CHARACTER ELEMENT HAS NOT BEEN SUBSCRIPTED             XZFRE
4          SET DISPLACEMENT = 0                                              XZFRE
3       ENDIF                                                                XZFRE
3       POP RESULT OPERAND FROM RESULT STACK (SYMBOL TABLE INDEX)            XZFRE
3       GET DATA TYPE FOR RESULT OPERAND FROM SYMBOL TABLE                   XZFRE
3       IF RESULT OPERAND IS FREE, THEN                                      XZFRE
4          CALL XPREQ TO RETRIEVE 1 WORD FROM RESULT OPERAND AT DISPLACEMENT DETERMIND   XZFRE
4          SET # WORDS TO BE STORED IN OBJECT = 1                            XZFRE
3       ELSE  FREE = CHARACTER DATA ELEMENT                                  XZFRE
4          CALL XPREQ TO RETRIEVE LOGICAL ELEMENT OF CHARACTER DATA FROM RESULT OPERAND   XZFRE
4          AT DISPLACEMENT DETERMINED                                        XZFRE
4          SET # WORDS TO BE STORED IN OBJECT = # WORDS/ELEMENT FOR RESULT OPERAND   XZFRE
3       ENDIF                                                                XZFRE

3       :FREFIX:                                                             XZFRE
3       CALL XZPCS TO POP RESULT OPERAND, CONVERT IF NECESSARY, AND SET UP FOR STORE   XZFRE
3       SET # WORDS TO BE STORED = RESULT DATA TYPE                          XZFRE
2    END CASE                                                                XZFRE
1 END XZFRE                                                                  XZFRE
```


C-5

```
1785    1 CD0                                                                   XZFXD
1786    1 CD0       FORTRAN CALLING PROCEDURE:                                  XZFXD
1787    1 CD0                                                                   XZFXD
1788    1 CD0       CALL XZFXD                                                  XZFXD
1789    1 CD0                                                                   XZFXD
1790    1 CD********************                                               XZFXD
1791    1 CD1                                                                   XZFXD
1792    1 CD1       XZFXD IS USED BY THE ASSGN ROUTINE XZPS2 TO PROCESS DATA ASSIGNMENTS  XZFXD
1793    1 CD1       FOR FIXED-TYPE OBJECT DATA ELEMENTS                         XZFXD
1794    1 CD1                                                                   XZFXD
1795    1 CD********************                                               XZFXD
1796    1 CD2                                                                   XZFXD
1797    1 CD2       INPUTS FROM ASGCOM                                          XZFXD
1798    1 CD2                                                                   XZFXD
1799    1 CD2       LU,SYMTAB,SSTRNG,RESULT,RSLTPT,DATYPS,CLSTRN                XZFXD
1800    1 CD2                                                                   XZFXD
1801    1 CD********************                                               XZFXD
1802    1 CD3                                                                   XZFXD
1803    1 CD3       OUTPUTS TO ASGCOM                                           XZFXD
1804    1 CD3                                                                   XZFXD
1805    1 CD3       RSLTPT,REQST,NUMWDS,OPRNDS                                  XZFXD
1806    1 CD3                                                                   XZFXD
1807    1 CD********************                                               XZFXD
1808    1 CD5                                                                   XZFXD
1809    1 CD5       FDS - XPREQ,XPXIT,XRMOV,XZLSS,XZMSG,XZPCS                   XZFXD
1810    1 CD5                                                                   XZFXD
1811    1 CD5       RTE - IAND                                                  XZFXD
1812    1 CD5                                                                   XZFXD
1813    1 CD********************                                               XZFXD
```

```
1815    1 BEGIN XZFXD                                                                                                  XZFXD
1816    2    SET # WORDS TO BE STORED IN OBJECT = OBJECT DATA TYPE                                                     XZFXD
1817    2 *                                        -1       -2       -1       1        2                            XZFXD
1818    2    CASE  (RESULT DATA TYPE )    :NOCHAR:,:FIXFRE:,:FIXFRE:,:FIXFIX:,:FIXFIX:                             XZFXD
1819    3 *                                          3                                                               XZFXD
1820    3 *                                 ,:FIXFIX:                                                              XZFXD

1821    3       :FIXFRE:                                                                                           XZFXD
1822    3       ERREXIT TO :NOCHAR: IF TYPE IN SYMBOL TABLE FOR RESULT OPERAND IS NOT FREE                         XZFXD
1823    3       IF TOP ENTRY ON RESULT STACK IS A DISPLACEMENT (TYPE = -2), THEN                                   XZFXD
1824    4          POP DISPLACEMENT FROM RESULT STACK                                                              XZFXD
1825    3       ELSE  FREE ELEMENT HAS NOT BEEN SUBSCRIPTED                                                        XZFXD
1826    4          SET DISPLACEMENT = 0                                                                            XZFXD
1827    3       ENDIF                                                                                              XZFXD
1828    3       POP RESULT OPERAND FROM RESULT STACK  (SYMBOL TABLE INDEX)                                         XZFXD
1829    3       CALL XPREQ TO RETRIEVE DATA FROM RESULT OPERAND AT DISPLACEMENT DETERMINED                         XZFXD
1830    3       (# WORDS RETRIEVED = OBJECT DATA TYPE)                                                             XZFXD

1831    3       :FIXFIX:                                                                                           XZFXD
1832    3       SET TARGET TYPE TO OBJECT DATA TYPE                                                                XZFXD
1833    3       CALL XZPCS TO POP RESULT VALUE,CONVERT IF NECESSARY, AND SET UP FOR STORE                          XZFXD
1834    2    ENDCASE                                                                                               XZFXD
1835    1 EXIT XZFXD                                                                                               XZFXD

1836    2    :NOCHAR:                                                                                              XZFXD
1837    2    SET MESSAGE TO BE OUTPUT TO "NUMERICAL DATA ELEMENT CANNOT BE SET EQUAL TO                            XZFXD
1838    2 *                              CHARACTER DATA"                                                           XZFXD
1839    2    CALL XZMSG TO OUTPUT MESSAGE TO USER                                                                  XZFXD
1840    2    CALL XZLSS TO LIST SYMBOLIC STRING                                                                    XZFXD
1841    2    CALL XPXIT TO EXIT PROCESSER                                                                          XZFXD
1842    1 END XZFXD                                                                                                XZFXD
```


```
1844    1 CD0                                                                                                  XZLSS
1845    1 CD0       FORTRAN CALLING PROCEEDURE                                                                 XZLSS
1846    1 CD0                                                                                                  XZLSS
1847    1 CD0    CALL XZLSS (LU, STRING, INDEX)                                                                XZLSS
1848    1 CD0                                                                                                  XZLSS
1849    1 C********                                                                                            XZLSS
1850    1 CD1                                                                                                  XZLSS
1851    1 CD1    XZLSS IS CALLED TO LIST A SYMBOLIC STRING AND AN INDICATOR TO  A                               XZLSS
1852    1 CD1    PARTICULAR TOKEN IN THAT STRING                                                               XZLSS
1853    1 CD1                                                                                                  XZLSS
1854    1 C********                                                                                            XZLSS
1855    1 CD2                                                                                                  XZLSS
1856    1 CD2       INPUT                                                                                      XZLSS
1857    1 CD2                                                                                                  XZLSS
1858    1 CD2    LU     - LOGICAL UNIT NO. FOR OUTPUT OF STRING                                                XZLSS
1859    1 CD2    STRING - SYMBOLIC STRING TO BE LISTED                                                         XZLSS
1860    1 CD2    INDEX  - SUBSCRIPT INTO STRING OF THE TOKEN TO BE INDICATED                                   XZLSS
1861    1 CD2                                                                                                  XZLSS
1862    1 C********                                                                                            XZLSS
1863    1 CD3                                                                                                  XZLSS
1864    1 CD3       OUTPUT                                                                                     XZLSS
1865    1 CD3                                                                                                  XZLSS
1866    1 CD3    THE SYMBOLIC STRING IS OUTPUT TO THE LU FOLLOWED BY A LINE CONTAINING                         XZLSS
1867    1 CD3    AN INDICATOR (UP ARROW) TO THE DESIGNATED TOKEN.                                              XZLSS
1868    1 CD3                                                                                                  XZLSS
1869    1 C********                                                                                            XZLSS
```

```
1871   1 CD4                                                                                                                XZLSS
1872   1 CD4      INTERNAL VARIABLES                                                                                        XZLSS
1873   1 CD4                                                                                                                XZLSS
1874   1 CD4    CONTRL = CONTROL TABLE DESCRIBING DISPOSITION AND PROCESSING                                                XZLSS
1875   1 CD4             FOR EACH OF THE TOKEN VALUES 1-32. EACH CONTROL TABLE                                          XZLSS
1876   1 CD4             ENTRY IS 3 WORDS :                                                                             XZLSS
1877   1 CD4                   WORD 1 (SIZE) = NO. OF WORDS IN PRINT BUFFER                                             XZLSS
1878   1 CD4                   WORD 2 (FIELD)= CONTENTS TO GO INTO PRINT BUFFER                                         XZLSS
1879   1 CD4                                   OR FLAG DESCRIBING HOW TO COMPUTE                                        XZLSS
1880   1 CD4                                   THEM                                                                     XZLSS
1881   1 CD4                   WORD 3 (TOKSIZ) = NO. OF WORDS IN SYMBOLIC STRING                                        XZLSS
1882   1 CD4                                                                                                            XZLSS
1883   1 CD4                                                                                                            XZLSS
1884   1 CD4  ENTRY  TOKEN       WORD 1          WORD 2              WORD 3                                             XZLSS
1885   1 CD4   NO.               (SIZE)          (FIELD)             (TOKSIZ)                                           XZLSS
1886   1 CD4     1   INTEGER       3           -1=>CALL XRI6           2                                                XZLSS
1887   1 CD4     2   REAL          7           -2=>CALL XRE14          3                                                XZLSS
1888   1 CD4     3   DOUBLE        9           -3=>CALL XRD13          4                                                XZLSS
1889   1 CD4     4   ----          0            0=>ERROR (INVALID)     0                                                XZLSS
1890   1 CD4     5   NAME          3           -4=>USE 3 WORDS         4                                                XZLSS
1891   1 CD4                                   FOLLOWING TOKEN                                                          XZLSS
1892   1 CD4     6   ----          0            0=>ERROR (INVALID)     0                                                XZLSS
1893   1 CD4     7   ----          0            0=>ERROR (INVALID)     0                                                XZLSS
1894   1 CD4     8   CHAR. STR. -1=>USE VALUE  -5=>USE SIZE WORDS  -1=>USE SIZE+2 WORDS                                 XZLSS
1895   1 CD4                    WORD FOLLOWING   BEGINNING 2 PAST                                                       XZLSS
1896   1 CD4                    TOKEN            TOKEN                                                                  XZLSS
1897   1 CD4     9   ----          0            0=>ERROR (INVALID)     0                                                XZLSS
1898   1 CD4    10   +             1            +                      1                                                XZLSS
1899   1 CD4    11   -             1            -                      1                                                XZLSS
1900   1 CD4    12   *             1            *                      1                                                XZLSS
1901   1 CD4    13   /             1            /                      1                                                XZLSS
1902   1 CD4    14   <             1            <                      1                                                XZLSS
1903   1 CD4    15   >             1            >                      1                                                XZLSS
1904   1 CD4    16   #             1            #                      1                                                XZLSS
1905   1 CD4    17   @             1            @                      1                                                XZLSS
1906   1 CD4    18   =             1            =                      1                                                XZLSS
1907   1 CD4    19   &             1            &                      1                                                XZLSS
1908   1 CD4    20   ?             1            ?                      1                                                XZLSS
1909   1 CD4    21   (             1            (                      1                                                XZLSS
1910   1 CD4    22   )             1            )                      1                                                XZLSS
1911   1 CD4    23   ----          0            0=>ERROR (INVALID)     0                                                XZLSS
1912   1 CD4    24   END SYM.STR.  0           -7=>END SYMB. STR.      0                                                XZLSS
1913   1 CD4    25   X             1            X                      1                                                XZLSS
1914   1 CD4    26  BACKSLASH      1            BACKSLASH              1                                                XZLSS
1915   1 CD4    27   $             1            $                      1                                                XZLSS
1916   1 CD4    28   .             1            .                      1                                                XZLSS
1917   1 CD4    29 LEFT BRACKET    1           LEFT BRACKET            1                                                XZLSS
1918   1 CD4    30 RIGHT BRACKET   1           RIGHT BRACKET           1                                                XZLSS
1919   1 CD4    31 REPEAT          4           -6=>CALL XRI6 AND       2                                                XZLSS
1920   1 CD4                                   APPEND "R "                                                              XZLSS
1921   1 CD4    32   ,             1            ,                      1                                                XZLSS
1922   1 CD4                                                                                                            XZLSS
```

```
1924   1 C********                                                                                                    XZLSS
1925   1 CDS                                                                                                          XZLSS
1926   1 CDS       NOTES                                                                                              XZLSS
1927   1 CDS                                                                                                          XZLSS
1928   1 CDS    USES ROUTINES                                                                                         XZLSS
1929   1 CDS                                                                                                          XZLSS
1930   1 CDS       EXEC                                                                                               XZLSS
1931   1 CDS       XRI6                                                                                               XZLSS
1932   1 CDS       XRE14                                                                                              XZLSS
1933   1 CDS       XRD18                                                                                              XZLSS
1934   1 CDS       XRMSG                                                                                              XZLSS
1935   1 CDS       XRMOV                                                                                              XZLSS
1936   1 CDS                                                                                                          XZLSS
1937   1 C********                                                                                                    XZLSS
1938   1 BEGIN XZLSS                                                                                                  XZLSS
1939   2    MOVE A ' CHARACTER INTO PRINT BUFFER AND INCREMENT BUFFER POINTER                                          XZLSS
1940   2    DO UNTIL ALL TOKENS OF SYMBOLIC STRING HAVE BEEN PROCESSED                                                XZLSS
1941   2    EXIT TO ERROR 1 IF TOKEN VALUE IS < 1 OR > 32                                                             XZLSS
1942   3       USE TOKEN VALUE TO RETRIVE 3 CONTROL WORDS (SIZE, FIELD, TOKSIZ)                                       XZLSS
1943   2    EXIT TO ERROR 2 IF FIELD = 0                                                                              XZLSS
1944   3       IF SIZE < 0, THEN                                                                                      XZLSS
1945   4          SET SIZE TO VALUE IN WORD FOLLOWING THIS TOKEN IN THE SYMB. STRING                                   XZLSS
1946   3       ENDIF                                                                                                  XZLSS
1947   3       IF TOKSIZ < 0, THEN                                                                                    XZLSS
1948   4          SET TOKSIZ TO SIZE + 2                                                                              XZLSS
1949   3       ENDIF                                                                                                  XZLSS
1950   3       IF THERE IS NOT ROOM IN PRINT BUFFER FOR SIZE WORDS, THEN                                              XZLSS
1951   4          CALL EXEC TO WRITE PRINT BUFFER TO INDICATED DEVICE                                                 XZLSS
1952   4          SET BUFFER POINTER TO 1ST POSITION FOR DATA                                                         XZLSS
1953   4          IF INDICATED TOKEN PROCESSED, THEN                                                                 XZLSS
1954   5             OUTPUT LINE WITH INDICATOR                                                                      XZLSS
1955   4          ENDIF                                                                                              XZLSS
1956   3       ENDIF                                                                                                  XZLSS
1957   3       IF FIELD > 0, THEN                                                                                     XZLSS
1958   4          MOVE FIELD INTO CURRENT PRINT BUFFER POSITION                                                       XZLSS
1959   3       ELSE                                                                                                   XZLSS
1960   4          CASE (ONE, TWO, THREE, FOUR, FIVE, SIX, EXIT), -FIELD                                               XZLSS

1961   5             :ONE: CALL XRI6 WITH VALUE IN NEXT WORD OF SYMB. STRING                                          XZLSS
1962   5             AND PUT RESULTS INTO PRINT BUFFER                                                                XZLSS

1963   5             :TWO: CALL XRE14 WITH VALUE IN NEXT 2 WORDS OF SYMB. STRING                                      XZLSS
1964   5             AND PUT RESULTS INTO PRINT BUFFER                                                                XZLSS

1965   5             :THREE: CALL XRD18 WITH VALUE IN NEXT 3 WORDS OF SYMB. STRING                                    XZLSS
1966   5             AND PUT RESULTS INTO PRINT BUFFER                                                                XZLSS

1967   5             :FOUR: MOVE THE NEXT 3 WORDS OF SYMB. STRING INTO PRINT BUFFER                                   XZLSS

1968   5             :FIVE: MOVE SIZE WORDS FROM 2ND WORD PAST CURRENT TOKEN THE SYMB. STRING                         XZLSS

1969   5             :SIX: CALL XRI6 WITH VALUE IN NEXT WORD OF SYMB. STRING AND PUT RESULTS                          XZLSS
1970   5             INTO PRINT BUFFER FOLLOWED BY AN "R"                                                             XZLSS

1971   5             :EXIT: PUT A ' CHARACTER INTO THE PRINT BUFFER, INCREMENT THE BUFFER                              XZLSS
1972   5             INDEX BY 1                                                                                       XZLSS
1973   5             CALL EXEC TO WRITE THE PRINT BUFFER TO INDICATED DEVICE                                          XZLSS
1974   5             IF INDICATED TOKEN HAS BEEN PROCESSED, AND                                                       XZLSS
1975   6                INDICATOR LINE NOT YET OUTPUT, THEN                                                           XZLSS
```

```
                                                                    XZLSS
                                                                    XZLSS
                                                                    XZLSS
                                                                    XZLSS
                                                                    XZLSS
                                                                    XZLSS
                                                                    XZLSS
                                                                    XZLSS
1976  6        OUTPUT LINE WITH INDICATOR
1977  5           ENDIF
1978  4        EXIT XZLSS
1979  4           ENDCASE
1980  3        ENDIF
1981  3        INCREMENT PRINT BUFFER INDEX BY SIZE
1982  3        INCREMENT THE SYMBOLIC STRING INDEX BY TOKSIZ
1983  2     ENDDO
1984  1  END XZLSS
```

```
1986    1 CD0       FORTRAN CALLING PROCEDURE                                                    XZMSG
1987    1 CD0                                                                                    XZMSG
1988    1 CD0    CALL XZMSG (CONT, NUMBER, LOCATE, LENGTH, SOURCE)                               XZMSG
1989    1 CD0                                                                                    XZMSG
1990    1 C*********                                                                             XZMSG
1991    1 CD1                                                                                    XZMSG
1992    1 CD1       INSERT 'LENGTH' WORDS OF TEXT FROM 'SOURCE' INTO MESSAGE                     XZMSG
1993    1 CD1       'NUMBER' BEGINNING AFTER 'LOCATE' WORDS, CONCATENATE TO PREFIX,              XZMSG
1994    1 CD1       TRUNCATE TO EIGHTY CHARACTERS AND OUTPUT TO USER TERMINAL                    XZMSG
1995    1 CD1                                                                                    XZMSG
1996    1 C*********                                                                             XZMSG
1997    1 CD2       INPUT                                                                        XZMSG
1998    1 CD2    CONT   - THREE CONTROL WORDS CONTAINING THE TERMINAL LU AND THE                 XZMSG
1999    1 CD2             DEBUG CONTROL FLAG.  IF CONT(3) BIT 12 IS ON, XUDBG WILL             XZMSG
2000    1 CD2             BE CALLED AFTER THE MESSAGE IS OUTPUT                                XZMSG
2001    1 CD2    NUMBER - INTEGER MESSAGE NUMBER OF THE FORM 'ANN' WHERE                        XZMSG
2002    1 CD2             A  - AREA INDICATOR AS FOLLOWS                                       XZMSG
2003    1 CD2                  1 - AS                                                          XZMSG
2004    1 CD2                  2 - XD                                                          XZMSG
2005    1 CD2                  3 - XE                                                          XZMSG
2006    1 CD2                  4 - XI                                                          XZMSG
2007    1 CD2                  5 - XS                                                          XZMSG
2008    1 CD2                  6 - XT                                                          XZMSG
2009    1 CD2                  7 - XX                                                          XZMSG
2010    1 CD2                  8 - XL                                                          XZMSG
2011    1 CD2                  9 - DF                                                          XZMSG
2012    1 CD2                 10 - SC                                                          XZMSG
2013    1 CD2             NN - MESSAGE NUMBER OR ZERO WHICH INDICATES ONLY 'LENGTH'            XZMSG
2014    1 CD2                  WORDS OF 'SOURCE' TO BE OUTPUT                                  XZMSG
2015    1 CD2    LOCATE - INTEGER NUMBER OF WORDS OF MESSAGE TO PRECEED                         XZMSG
2016    1 CD2             'SOURCE' (NOT USED IF 'NN' OF 'NUMBER' IS ZERO)                      XZMSG
2017    1 CD2    LENGTH - INTEGER NUMBER OF WORDS OF 'SOURCE' TO BE INSERTED INTO             XZMSG
2018    1 CD2             MESSAGE.  ZERO INDICATES NO INSERTION                                XZMSG
2019    1 CD2    SOURCE - ARRAY OF CHARACTERS TO BE INSERTED INTO MESSAGE (NOT USED           XZMSG
2020    1 CD2             IF 'LENGTH' IS ZERO)                                                 XZMSG
2021    1 CD2                                                                                  XZMSG
2022    1 C*********                                                                           XZMSG
2023    1 CD3       OUTPUT                                                                     XZMSG
2024    1 CD3    UP TO EIGHTY CHARACTER LINE OF TEXT TO UNIT 'LU' OF THE FORM                  XZMSG
2025    1 CD3                                                                                  XZMSG
2026    1 CD5    *AANN*    MESSAGE(1-LOCATE) SOURCE(1-LENGTH) REMAINDER OF MESSAGE            XZMSG
2027    1 CD3                                                                                  XZMSG
2028    1 C*********                                                                           XZMSG
2029    1 CD5       NOTES                                                                      XZMSG
2030    1 CD5    USES FDS SYSTEM MESSAGE FILE )XRMSG                                           XZMSG
2031    1 CD5    USES CLOSE, EXEC, IAND, KCVT, OPEN, READF, XRMOV, XUDBG                       XZMSG
2032    1 CD5                                                                                  XZMSG
2033    1 C*********                                                                           XZMSG
```

```
2035    1 BEGIN XZMSG                                                                                                    XZMSG
2036    2    SEPARATE NUMBER INTO AREA AND MESSAGE NUMBER                                                                XZMSG
2037    2    SET NUMBER IN PREFIX                                                                                        XZMSG
2038    2    READ MESSAGE DIRECTORY RECORD                                                                               XZMSG
2039    2    IF AREA VALID                                                                                               XZMSG
2040    2    THEN                                                                                                        XZMSG
2041    3       SET AREA CODE IN PREFIX                                                                                  XZMSG
2042    3       IF MESSAGE NUMBER > 0                                                                                    XZMSG
2043    3       THEN                                                                                                     XZMSG
2044    4          IF VALID MESSAGE NUMBER                                                                               XZMSG
2045    4          THEN                                                                                                  XZMSG
2046    5             COMPUTE MESSAGE RECORD NUMBER                                                                      XZMSG
2047    5             READ RECORD                                                                                        XZMSG
2048    5             CALL XRMOV TO MOVE LOCATE WORDS FROM RECORD INTO BUFFER                                            XZMSG
2049    4          ELSE                                                                                                  XZMSG
2050    4          EXIT TO :ERROR:                                                                                       XZMSG
2051    4          ENDIF                                                                                                 XZMSG
2052    3       ENDIF                                                                                                    XZMSG
2053    3       CALL XRMOV TO MOVE LENGTH WORDS FROM SOURCE INTO BUFFER (MAX OF 40 TOTAL WORDS)                          XZMSG
2054    3       IF MESSAGE NUMBER > 0                                                                                    XZMSG
2055    3       THEN                                                                                                     XZMSG
2056    4          CALL XRMOV TO MOVE REMAINING RECORD INTO BUFFER (MAX OF 40 TOTAL WORDS)                               XZMSG
2057    3       ENDIF                                                                                                    XZMSG
2058    2    ELSE                                                                                                        XZMSG
2059    3       SET AREA IN PREFIX                                                                                       XZMSG
2060    3       DERROR:  CALL XRMOV TO MOVE 'XZMSG ERROR' INTO BUFFER                                                    XZMSG
2061    3       CALL XRMOV TO MOVE LENGTH WORDS OF SOURCE INTO BUFFER (MAX OF 40 TOTAL WORDS)                            XZMSG
2062    2    ENDIF                                                                                                       XZMSG
2063    2    OUTPUT BUFFER TO USER'S TERMINAL                                                                            XZMSG
2064    2    IF DEBUG IS REQUESTED                                                                                       XZMSG
2065    2    THEN                                                                                                        XZMSG
2066    3       CALL XUDBG                                                                                               XZMSG
2067    2    ENDIF                                                                                                       XZMSG
2068    1 END XZMSG                                                                                                      XZMSG
```

```
2070    1 CD0
2071    1 CD0       FORTRAN CALLING PROCEDURE:                                                                    XZOPR
2072    1 CD0                                                                                                     XZOPR
2073    1 CD0       CALL XZOPR(ENTRY)                                                                             XZOPR
2074    1 CD0                                                                                                     XZOPR
2075    1 CD********************                                                                                  XZOPR
2076    1 CD1                                                                                                     XZOPR
2077    1 CD1       XZOPR IS USED BY THE ASSGN ROUTINE XZPS2 TO EVALUATE MATH OPERATIONS                           XZOPR
2078    1 CD1                                                                                                     XZOPR
2079    1 CD********************                                                                                  XZOPR
2080    1 CD2                                                                                                     XZOPR
2081    1 CD2       INPUTS                                                                                        XZOPR
2082    1 CD2                                                                                                     XZOPR
2083    1 CD2       ENTRY - OPERATOR TOKEN CURRENTLY BEING PROCESSED                                              XZOPR
2084    1 CD2                                                                                                     XZOPR
2085    1 CD2       FROM ASGCOM - LU,SSTRNG,SYMTAB,DATYPS,RSLTPT,CLSTRN,MAPWDS,RESULT                             XZOPR
2086    1 CD2                                                                                                     XZOPR
2087    1 CD********************                                                                                  XZOPR
2088    1 CD3                                                                                                     XZOPR
2089    1 CD3       OUTPUTS TO ASGCOM                                                                             XZOPR
2090    1 CD3                                                                                                     XZOPR
2091    1 CD3       RESULT,OPRNDS,REQST,RSLTPT                                                                    XZOPR
2092    1 CD3                                                                                                     XZOPR
2093    1 CD********************                                                                                  XZOPR
2094    1 CD4                                                                                                     XZOPR
2095    1 CD4       INTERNAL VARIABLES                                                                            XZOPR
2096    1 CD4                                                                                                     XZOPR
2097    1 CD4       MAPOP - MAPS OPERATOR TOKENS FOR EXECUTION                                                    XZOPR
2098    1 CD4                                                                                                     XZOPR
2099    1 CD********************                                                                                  XZOPR
2100    1 CD5                                                                                                     XZOPR
2101    1 CD5       EXTERNAL REFERENCES                                                                           XZOPR
2102    1 CD5                                                                                                     XZOPR
2103    1 CD5       FDS - XPREQ,XPXIT,XRMOV,XZLSS,XZMSG                                                           XZOPR
2104    1 CD5                                                                                                     XZOPR
2105    1 CD5       RTE - IAND,OVF                                                                                XZOPR
2106    1 CD5                                                                                                     XZOPR
2107    1 CD********************                                                                                  XZOPR
```

```
1 BEGIN XZOPR
                          +       -        *         /          (,)         ()
2 *
2    CASE  OPERATOR  :ADD:, :SUBTR:, :MULT:, :DIVIDE:, :INDEX:, :SUBSCR:,
3 *                         **      UNARY -
3 *                      :EXPO:, :UMINUS:

3        :ADD:
3        PERFORM ADDITION
3        SET RESULT'S DATA TYPE AS APPROPRIATE

3        :SUBTR:
3        PERFORM SUBTRACTION
3        SET RESULT'S DATA TYPE AS APPROPRIATE

3        :MULT:
3        PERFORM MULTIPLICATION
3        SET RESULT'S DATA TYPE AS APPROPRIATE

3        :DIVIDE:
3        PERFORM DIVISION
3        SET RESULT'S DATA TYPE AS APPROPRIATE

3        :INDEX:
3        IF RESULT STACK IS NOT EMPTY, THEN
4           GET I-DIM FOR TOP OPERAND IN RESULT STACK
4           SET # WORDS PER ELEMENT BASED ON TYPE OF TOP ENTRY OF RESULT STACK
3        ELSE  DOUBLE SUBSCRIPT FOR OBJECT IS BEING EVALUATED
4           GET I-DIM FOR FIRST ENTRY IN SYMBOL TABLE
4           SET # WORDS PER ELEMENT BASED ON TYPE OF FIRST ENTRY IN SYMBOL TABLE
3        ENDIF
3        CALCULATE RESULT (I.E., INDEX = IDIM*(SECOND OPERAND-1)+FIRST OPERAND)

3        :SUBSCR:
3        IF RESULT STACK IS NOT EMPTY, THEN
4           SET # WORDS/ELEMENT BASED ON TOP ENTRY OF RESULT STACK ENTRY
3        ELSE  OBJECT IS BEING SUBSCRIPTED
4           SET # WORDS/ELEMENT BASED ON TYPE OF FIRST ENTRY IN SYMBOL TABLE
3        ENDIF
3        CALCULATE RESULT (I.E.,DISPLACEMENT = (OPERAND-1)* #WORDS PER ENTRY)
3        IF RESULT STACK IS NOT EMPTY AND TOP ENTRY IS NOT FREE OR CHARACTER, THEN
4           CALL XPREQ TO RETRIEVE A LOGICAL VALUE FOR TOP OPERAND AT DISPLACEMENT
4           DECREMENT RESULT STACK POINTER TO DISCARD ARRAY NAME
3        ELSE  OBJECT IS BEING SUBSCRIPTED
4           SET TYPE = -2
3        ENDIF

3        :EXPO:
3        PERFORM EXPONENTIATION
3        SET RESULT'S DATA TYPE AS APPROPRIATE

3        :UMINUS:
3        CHANGE SIGN OF OPERAND FOR RESULT
2     ENDCASE
2     ERREXIT IF OVERFLOW OR UNDERFLOW IS INDICATED TO :OVER:
2     PUSH RESULT AND TYPE ONTO RESULT STACK
1 EXIT XZOPR

2     :OVER:
2     SET MESSAGE TO BE OUTPUT TO "OVERFLOW OR UNDERFLOW DETECTED"
```

```
2159    2    CALL XZMSG TO OUTPUT MESSAGE TO USER                    XZOPR
2160    2    CALL XZLSS TO LIST SYMBOLIC STRING                      XZOPR
2161    2    CALL XPXIT TO EXIT PROCESSER                            XZOPR
2162    1 END XZOPR                                                  XZOPR
                                                                     XZOPR
```

```
2164    1 CD0
2165    1 CD0       FORTRAN CALLING PROCEDURE:                                                                  XZPCS
2166    1 CD0                                                                                                   XZPCS
2167    1 CD0       CALL XZPCS(TARGET,OPNUM)                                                                    XZPCS
2168    1 CD0                                                                                                   XZPCS
2169    1 CD********************                                                                               XZPCS
2170    1 CD1                                                                                                   XZPCS
2171    1 CD1       XZPCS IS USED BY ASSGN TO POP AN OPERAND FROM THE RESULT STACK, CONVERT                     XZPCS
2172    1 CD1       IT TO A TARGET TYPE, AND STORE IT FOR USE IN A MATH OR FUNCTION                             XZPCS
2173    1 CD1       OPERATION                                                                                   XZPCS
2174    1 CD1                                                                                                   XZPCS
2175    1 CD********************                                                                               XZPCS
2176    1 CD2                                                                                                   XZPCS
2177    1 CD2       INPUTS                                                                                      XZPCS
2178    1 CD2                                                                                                   XZPCS
2179    1 CD2       TARGET - DESIRED FDS FIXED DATA TYPE                                                        XZPCS
2180    1 CD2       OPNUM - OPERAND NUMBER FOR ENTRY CURRENTLY BEING SET UP                                     XZPCS
2181    1 CD2                                                                                                   XZPCS
2182    1 CD2       FROM ASGCOM - LU,SSTRNG,RESULT,RSLTPT,DATYPS                                                XZPCS
2183    1 CD2                                                                                                   XZPCS
2184    1 CD********************                                                                               XZPCS
2185    1 CD3                                                                                                   XZPCS
2186    1 CD3       OUTPUTS TO ASGCOM                                                                           XZPCS
2187    1 CD3                                                                                                   XZPCS
2188    1 CD3       RSLTPT,OPRNDS                                                                               XZPCS
2189    1 CD3                                                                                                   XZPCS
2190    1 CD********************                                                                               XZPCS
2191    1 CD5                                                                                                   XZPCS
2192    1 CD5       EXTERNAL REFERENCES                                                                         XZPCS
2193    1 CD5                                                                                                   XZPCS
2194    1 CD5       FDS - XPXIT,XRMOV,XZLSS,XZMSG                                                               XZPCS
2195    1 CD5                                                                                                   XZPCS
2196    1 CD5       RTE - CBLE,FLOST,IFIX,OVF,SNGL                                                              XZPCS
2197    1 CD5                                                                                                   XZPCS
2198    1 CD********************                                                                               XZPCS
                                                                                                                XZPCS
```

```
2200    1 BEGIN XZPCS                                                                                                  XZPCS
2201    2    POP OPERAND AND DATA TYPE FROM TOP ENTRY OF RESULT STACK                                                  XZPCS
2202    2    IF TARGET TYPE AND DATA TYPE ARE NOT EQUAL, THEN                                                          XZPCS
2203    3       CASE  (TARGET TYPE)  :INTG:, :REAL:, :DBLE:                                                           XZPCS

2204    4          :INTG:                                                                                             XZPCS
2205    4          CONVERT OPERAND TO INTEGER                                                                         XZPCS

2206    4          :REAL:                                                                                             XZPCS
2207    4          CONVERT OPERAND TO SINGLE PRECISION REAL                                                           XZPCS

2208    4          :DBLE:                                                                                             XZPCS
2209    4          CONVERT OPERAND TO DOUBLE PRECISION REAL                                                           XZPCS
2210    3       ENDCASE                                                                                               XZPCS
2211    2    ENDIF                                                                                                     XZPCS
2212    2    ERREXIT IF OVERFLOW OR UNDERFLOW IS INDICATED  TO :OVER:                                                  XZPCS
2213    2    STORE OPERAND AND CURRENT DATA TYPE FOR USE BY FUNCTION OP ARITHMETIC OPER.                                XZPCS
2214    1 EXIT XZPCS                                                                                                   XZPCS

2215    2    :OVER:                                                                                                    XZPCS
2216    2    SET MESSAGE TO BE OUTPUT TO "OVERFLOW OR UNDERFLOW DETECTED"                                              XZPCS
2217    2    CALL XZMSG TO OUTPUT MESSAGE TO USER                                                                      XZPCS
2218    2    CALL XZLSS TO LIST SYMBOLIC STRING                                                                        XZPCS
2219    2    CALL XPXIT TO EXIT PROCESSER                                                                              XZPCS
2220    1 END XZPCS                                                                                                    XZPCS
```

```
2222    1 CD0                                                                   XZPS1
2223    1 CD0       FORTRAN CALLING PROCEDURE:                                  XZPS1
2224    1 CD0                                                                   XZPS1
2225    1 CD0       CALL XZPS1                                                  XZPS1
2226    1 CD0                                                                   XZPS1
2227    1 CD*********************                                               XZPS1
2228    1 CD1                                                                   XZPS1
2229    1 CD1       XZPS1 IS USED BY ASSGN TO TRANSLATE THE SYMBOLIC STRING EXPRESSION  XZPS1
2230    1 CD1       TO A POST-FIX NOTATION STRING                               XZPS1
2231    1 CD1                                                                   XZPS1
2232    1 CD*********************                                               XZPS1
2233    1 CD2                                                                   XZPS1
2234    1 CD2       INPUTS FROM ASGCOM                                          XZPS1
2235    1 CD2                                                                   XZPS1
2236    1 CD2       LU, TOKENS, SSTRNG, EXPTR, SYNTAX, FNCTBL                   XZPS1
2237    1 CD2                                                                   XZPS1
2238    1 CD*********************                                               XZPS1
2239    1 CD3                                                                   XZPS1
2240    1 CD3       OUTPUTS TO ASGCOM                                           XZPS1
2241    1 CD3                                                                   XZPS1
2242    1 CD3       EXPTR, RNGSTK, POLISH, SYNTAX                               XZPS1
2243    1 CD3                                                                   XZPS1
2244    1 CD*********************                                               XZPS1
2245    1 CD4                                                                   XZPS1
2246    1 CD4       INTERNAL VARIABLES                                          XZPS1
2247    1 CD4       GRPSTK - 2X120 ARRAY USED TO TRACK FUNCTIONS, SUBSCRIPTS, AND  XZPS1
2248    1 CD4                PARENTHETICAL GROUPINGS                            XZPS1
2249    1 CD4       OPSTK - 2X120 ARRAY; EACH ENTRY CONTAINS OPERATOR TOKEN AND ITS OUTPUT  XZPS1
2250    1 CD4                PRIORITY                                           XZPS1
2251    1 CD4       PRCDNT - PRECEDNT TYPE FOR PRECEDING CHARACTER ; USED FOR SYNTAX CHECK  XZPS1
2252    1 CD4       TKNPTR - POINTER TO TOKEN BEING PROCESSED IN SYMBOLIC STRING  XZPS1
2253    1 CD4       TOKEN - TOKEN CURRENTLY BEING PROCESSED                     XZPS1
2254    1 CD4                                                                   XZPS1
2255    1 CD*********************                                               XZPS1
2256    1 CD5                                                                   XZPS1
2257    1 CD5       EXTERNAL REFERENCES                                         XZPS1
2258    1 CD5                                                                   XZPS1
2259    1 CD5       RTE - IAND, MINO                                            XZPS1
2260    1 CD5                                                                   XZPS1
2261    1 CD5       FDS - XPXIT, XZLSS, XZMSG, XZSYM                            XZPS1
2262    1 CD5                                                                   XZPS1
2263    1 CD*********************                                               XZPS1
```

```
2265    1 BEGIN XZPS1                                                                  XZPS1
2266    2    DO FOR EACH TOKEN UNTIL ; OR ESS IS REACHED                               XZPS1
2267    3       PERFORM SETUP FOR SPECIAL HANDLING FOR NUMBERS, NAMES, +, *, (, AND "  XZPS1
2268    3       PERFORM STRING FOR SYNTAX CHECKING AND POLISH STRING BUILD              XZPS1
2269    2    END DO                                                                    XZPS1
2270    2    PERFORM RANGE TO PROCESS RANGE DEFINITION, IF NECESSARY                   XZPS1
2271    1 END XZPS1                                                                    XZPS1
```

```
2273    1 BEGIN SETUP                                                                                                    XZPS1
2274    2 *                    EOS       INTEGER    REAL     DOUBLE                                                     XZPS1
2275    2    CASE  TOKEN     :INVLD:,  :SYMBOL:, :SYMBOL:, :SYMBOL:,                                                    XZPS1
2276    3 *                    NAME    CHARACTER    +          -          *                                            XZPS1
2277    3 *                   :SYMBOL:, :CHAR:,  :PLUS:,    :MINUS:,  :ASTER:,                                       XZPS1
2278    3 *                      /         <          >          $          @                                          XZPS1
2279    3 *                   :ENDCAS:,  :INVLD:,  :INVLD:,  :INVLD:,  :INVLD:,                                       XZPS1
2280    3 *                      =         &          ?          (          )                                          XZPS1
2281    3 *                   :ENDCAS:,  :INVLD:,  :INVLD:,  :LPAREN:, :ENDCAS:,                                      XZPS1
2282    3 *                     BSS        ESS        X          BK/        $                                          XZPS1
2283    3 *                   :INVLD:,  :ENDCAS:,  :INVLD:,  :INVLD:,  :INVLD:,                                       XZPS1
2284    3 *                      .       LFT BRKT   RT BRKT     RPT        ,                                          XZPS1
2285    3 *                   :INVLD:,  :ENDCAS:,  :ENDCAS:,  :INVLD:,  :ENDCAS:                                      XZPS1
2286    3 *                      ;                                                                                    XZPS1
2287    3 *                   :ENDCAS:                                                                                XZPS1

2288    3       :SYMBOL:                                                                                             XZPS1
2289    3       IF TOKEN IS A NAME AND NEXT TOKEN IS A LFT BRKT, THEN                                                XZPS1
2290    4          STARTSEARCH UNTIL FUNCTION TABLE IS CHECKED                                                       XZPS1
2291    4          EXITIF TOKEN NAME MATCHES FUNCTION NAME                                                           XZPS1
2292    5             CHANGE TOKEN TO FUNCTION'S INDEX IN TABLE + 128                                                XZPS1
2293    5             PUSH COUNT=-BINARY/UNARY FLAG AND LIMIT=0 ONTO GROUPING STACK                                  XZPS1
2294    4          ENDLOOP                                                                                           XZPS1
2295    5             ERREXIT TO :BADFUN:IF NO MATCH IS FOUND                                                        XZPS1
2296    4          ENDSEARCH                                                                                         XZPS1
2297    3       ELSE  TOKEN IS AN OPERAND                                                                            XZPS1
2298    4          IF TOKEN IS A NAME AND NEXT TOKEN IS (, THEN                                                      XZPS1
2299    5             SET SUBSCRIPT FLAG                                                                             XZPS1
2300    4          ENDIF                                                                                             XZPS1
2301    4          CALL XZSYM TO STORE SYMBOL AND SUBSCRIPT FLAG                                                     XZPS1
2302    4          PUSH SYMBOL INDEX ONTO EXPRESSION STACK                                                           XZPS1
2303    3       ENDIF                                                                                                XZPS1

2304    3       :CHAR:                                                                                               XZPS1
2305    3       PUSH NEGATIVE POINTER TO CHARACTER STRING ONTO EXPRESSION STACK                                      XZPS1

2306    3       :PLUS:                                                                                               XZPS1
2307    3       IF PRECEDING TOKEN WAS LEFT BRACKET OR ( OR , OR =, THEN                                             XZPS1
2308    4          INCREMENT TO NEXT TOKEN AND EXIT TO :ENDO:                                                        XZPS1
2309    3       ENDIF                                                                                                XZPS1

2310    3       :MINUS:                                                                                              XZPS1
2311    3       IF PRECEDING TOKEN WAS LEFT BRACKET OR ( OR , OR =, THEN                                             XZPS1
2312    4          CHANGE TOKEN TO UNARY MINUS                                                                       XZPS1
2313    3       ENDIF                                                                                                XZPS1

2314    3       :ASTER:                                                                                              XZPS1
2315    3       IF NEXT TOKEN IS *, THEN                                                                             XZPS1
2316    4          CHANGE TOKEN TO EXPONENTIATION                                                                    XZPS1
2317    3       ENDIF                                                                                                XZPS1

2318    3       :LPAREN:                                                                                             XZPS1
2319    3       IF PRECEDING TOKEN WAS A VARIABLE (DATA ELEMENT NAME), THEN                                          XZPS1
2320    4          PUSH COUNT AND LIMIT ALLOWING DOUBLE SUBSCRIPTING ONTO GROUPING STACK                             XZPS1
2321    3       ELSE  EXPRESSION IS BEING PROCESSED                                                                  XZPS1
2322    4          PUSH COUNT AND LIMIT ALLOWING NO SUBSCRIPTING ONTO GROUPING STACK                                 XZPS1
2323    3       ENDIF                                                                                                XZPS1
2324    2    ENDCASE                                                                                                 XZPS1
```

```
2325    2     :ENDCAS:                                                                                    XZPS1
2326    1 EXIT SETUP                                                                                      XZPS1

2327    2     :INVLD:
2328    2     SET MESSAGE TO BE OUTPUT TO "INVALID CHARACTER"                                             XZPS1
                                                                                                          XZPS1

2329    2     :BADFUN:
2330    2     SET MESSAGE TO BE OUTPUT TO "FUNCTION NOT SUPPORTED BY THIS PROCESSOR"                      XZPS1
2331    2     CALL XZMSG TO OUTPUT MESSAGE TO USER                                                        XZPS1
2332    2     CALL XZLSS TO LIST SYMBOLIC STRING                                                          XZPS1
2333    2     CALL XPXIT TO EXIT PROCESSOR                                                                XZPS1
2334    1 END SETUP                                                                                       XZPS1
```

```
2336    1 BEGIN STRING                                                                                                  XZPS1
2337    2    SET INDEX INTO SYNTAX TABLE TO MIN(TOKEN,40)                                                               XZPS1
2338    2    ERREXIT TO :SYNTX1: IF THIS TOKEN IS NOT VALID ACCORDING TO SYNTAX TABLE                                   XZPS1
2339    2    IF TOKEN IS NOT AN OPERAND, THEN                                                                           XZPS1
2340    3       DO UNTIL TOKEN IS PUSHED ONTO OPERATOR STACK OR DISCARDED                                               XZPS1
2341    4          IF INPUT PRIORITY OF THIS TOKEN > OUTPUT PRIORITY OF TOP ENTRY IN OPERATOR                           XZPS1
2342    5             STACK, THEN                                                                                      XZPS1
2343    5             IF TOKEN IS ,  THEN                                                                              XZPS1
2344    6                INCREMENT COUNT FOR TOP ENTRY IN GROUPING STACK                                               XZPS1
2345    6                ERREXIT TO :COMERR: IF COUNT > COMMA LIMIT FOR TOP ENTRY IN GRPING STK                        XZPS1
2346    6                IF COMMA LIMIT SHOWS SUBSCRIPTING IN PROGRESS (LIMIT > 0) , THEN                              XZPS1
2347    ?                   PUSH TOKEN AND OUTPUT PRIORITY ONTO OPERATOR STACK                                          XZPS1
2348    6                ELSE                                                                                          XZPS1
2349    7                   DISCARD , (FUNCTION LIST IS BEING PROCESSED                                                 XZPS1
2350    6                ENDIF                                                                                         XZPS1
2351    5             ELSE  OPERATOR IS NOT ,                                                                          XZPS1
2352    6                PUSH TOKEN AND ITS OUTPUT PRIORITY ONTO OPERATOR STACK                                        XZPS1
2353    5             ENDIF                                                                                            XZPS1
2354    4          ELSE  INPUT PRIORITY IS < OR = OUTPUT PRIORITY                                                      XZPS1
2355    5             IF INPUT PRIORITY < OUTPUT PRIORITY OF TOP ENTRY IN OPERATOR STACK OR                            XZPS1
2356    6                INPUT PRIORITY = OUTPUT PRIORITY NOT = 2, THEN                                               XZPS1
2357    6                ERREXIT TO :SYNTX1: IF INPUT PRIORITY = 0 AND OPERATOR STACK IS EMPTY                        XZPS1
2358    6                IF TOP ENTRY OF OPERATOR STACK IS =, THEN                                                    XZPS1
2359    7                   ERREXIT IF THIS IS NOT LAST ENTRY ON OPERATOR STACK TO :BADEQL:                           XZPS1
2360    7                   ERREXIT IF GROUPING STACK IS NOT EMPTY TO :SYNTX2:                                        XZPS1
2361    7                   DISCARD TOKEN (; OR ESS)                                                                  XZPS1
2362    6                ENDIF                                                                                        XZPS1
2363    6                POP OPERATOR STACK                                                                           XZPS1
2364    6                PUSH OPERATOR ONTO EXPRESSION STACK                                                          XZPS1
2365    5             ELSE  BRACKETS OR PARENTHESES HVE BEEN MATCHED                                                  XZPS1
2366    6                IF TOP ENTRY OF OPERATOR STACK IS (, THEN                                                    XZPS1
2367    7                   ERREXIT IF CURRENT TOKEN IS NOT )  TO :SYNTX2:                                            XZPS1
2368    7                   IF TOP ENTRY OF GROUPING STACK INDICATES SUBSCRIPTING(LIMIT>0),THEN                       XZPS1
2369    8                      PUSH SUBSCRIPTING OPERATOR ONTO EXPRESSIUN ARRAY                                       XZPS1
2370    7                   ENDIF                                                                                     XZPS1
2371    6                ELSE  TOP OPERATOR ENTRY IS OPEN BRACKET (FUNCTION LIST)                                     XZPS1
2372    7                   ERREXIT IF CURRENT TOKEN IS NOT CLOSE BRACKET  TO :SYNTX2:                                XZPS1
2373    7                   ERREXIT TO :FLSTER: IF FUNCTION LIST IS NOT COMPLETE (TOP OF GRPING STK COUNT=0)          XZPS1
2374    6                ENDIF                                                                                        XZPS1
2375    6                POP OPERATOR STACK                                                                           XZPS1
2376    6                POP GROUPING STACK                                                                           XZPS1
2377    6                DISCARD CURRENT TOKEN                                                                        XZPS1
2378    5             ENDIF                                                                                           XZPS1
2379    4          ENDIF                                                                                              XZPS1
2380    3       END DO                                                                                                XZPS1
2381    2    ENDIF                                                                                                    XZPS1

2382    2    :ENDO:                                                                                                   XZPS1
2383    2    INCREMENT TO NEXT TOKEN USING TOKEN LENGTH FROM SYNTAX TABLE                                             XZPS1
2384    1 EXIT STRING                                                                                                 XZPS1

2385    2    :SYNTX1:                                                                                                 XZPS1
2386    2    SET MESSAGE TO BE OUTPUT TO "INVALID SEQUENCE OF CHARACTERS"                                             XZPS1

2387    2    :SYNTX2:                                                                                                 XZPS1
2388    2    SET MESSAGE TO BE OUTPUT TO "PARENTHESES OR BRACKETS DO NOT MATCH PROPERLY"                              XZPS1

2389    2    :COMERR:                                                                                                 XZPS1
2390    2    SET MESSAGE TO BE OUTPUT TO "INVALID COMMA OR TOO MANY COMMAS IN LIST"                                   XZPS1
```

```
2391    2     :FLSTER:                                                                    XZPS1
2392    2     SET MESSAGE TO BE OUTPUT TO "INCOMPLETE FUNCTION LIST"                      XZPS1

2393    2     :BADEQL:                                                                    XZPS1
2394    2     SET MESSAGE TO BE OUTPUT TO "INVALID SYNTAX TO LEFT OF = "                  XZPS1
2395    2     CALL XZMSG TO OUTPUT MESSAGE TO USER                                        XZPS1
2396    2     CALL XZLSS TO LIST SYMBOLIC STRING                                          XZPS1
2397    2     CALL XPXIT TO EXIT PROCESSOR                                                XZPS1
2398    1 END STRING                                                                      XZPS1
```

```
2400    1 BEGIN RANGE                                                                                    XZPS1
2401    2    INITIALIZE RANGE TABLE FOR 4 RANGES STARTING AT 1, ENDING AT 1, INCREMENT = 1               XZPS1
2402    2    IF LAST TOKEN INPUT WAS ;, THEN                                                             XZPS1
2403    3       DO UNTIL ESS IS REACHED OR FOUR RANGES ARE PROCESSED                                     XZPS1
2404    4          ERREXIT IF NEXT TOKEN IS NOT NAME TO :RNGSYN:                                         XZPS1
2405    4          CALL XZSYM TO INSERT NAME IN SYMBOL TABLE,IF NECESSARY, FLAG ENTRY AS                 XZPS1
2406    4          AN INDEX, AND RETURN SYMBOL TABLE INDEX                                               XZPS1
2407    4          ERREXIT IF NEXT TOKENS ARE NOT "=INTEGER,INTEGER"  TO :RNGSYN:                        XZPS1
2408    4          IF SECOND INTEGER VALUE < FIRST INTEGER VALUE, THEN                                   XZPS1
2409    5             SET INCREMENT TO -1                                                                XZPS1
2410    4          ENDIF                                                                                 XZPS1
2411    4          PUSH START, END, AND INCREMENT VALUES AND SYMBOL TABLE INDEX ONTO STACK                XZPS1
2412    3       END DO                                                                                   XZPS1
2413    3       ERREXIT TO :RNGSYN: IF ESS HAS NOT BEEN REACHED                                          XZPS1
2414    2    ENDIF                                                                                       XZPS1
2415    1 EXIT RANGE                                                                                     XZPS1

2416    2    :RNGSYN:                                                                                    XZPS1
2417    2    SET MESSAGE TO BE OUTPUT TO "RANGE SPECIFICATION DOES NOT FOLLOW: ";NAME=                    XZPS1
2418    2 *                              INTEGER,INTEGER"                                                XZPS1
2419    2    CALL XZMSG TO OUTPUT MESSAGE TO USER                                                        XZPS1
2420    2    CALL XZLSS TO LIST SYMBOLIC STRING                                                          XZPS1
2421    2    CALL XPXIT TO EXIT PROCESSOR                                                                XZPS1
2422    1 END RANGE                                                                                      XZPS1
```

```
2424    1 CD0                                                                   XZPS2
2425    1 CD0       FORTRAN CALLING PROCEDURE:                                  XZPS2
2426    1 CD0                                                                   XZPS2
2427    1 CD0       CALL XZPS2                                                  XZPS2
2428    1 CD0                                                                   XZPS2
2429    1 CD********************                                               XZPS2
2430    1 CD1                                                                   XZPS2
2431    1 CD1       XZPS2 IS USED BY ASSGN TO EVALUATE THE POST-FIX STRING GENERATED  XZPS2
2432    1 CD1       BY XZPS1 AND TO STORE THE RESULTS INTO THE OBJECT DATA ELEMENT  XZPS2
2433    1 CD1                                                                   XZPS2
2434    1 CD********************                                               XZPS2
2435    1 CD2                                                                   XZPS2
2436    1 CD2       INPUTS FROM ASGCOM                                          XZPS2
2437    1 CD2                                                                   XZPS2
2438    1 CD2       LU, TOKENS, LASTY, SYMTAB, SSTRNG, RESULT, OPRNDS           XZPS2
2439    1 CD2       OPINFO, NUMWDS, DATYPS, RSLTPT, EXPTR, CLSREQ, CLSTRN, DECLAS,  XZPS2
2440    1 CD2       RNGSTK, POLISH, SYNTAX, FNCTBL                              XZPS2
2441    1 CD2                                                                   XZPS2
2442    1 CD********************                                               XZPS2
2443    1 CD3                                                                   XZPS2
2444    1 CD3       OUTPUTS TO ASGCOM                                           XZPS2
2445    1 CD3                                                                   XZPS2
2446    1 CD3       SYMTAB, RESULT, REQST, RSLTPT, OPINFO                       XZPS2
2447    1 CD3                                                                   XZPS2
2448    1 CD********************                                               XZPS2
2449    1 CD4                                                                   XZPS2
2450    1 CD4       INTERNAL VARIABLES                                          XZPS2
2451    1 CD4                                                                   XZPS2
2452    1 CD4       ENTRY - ENTRY FROM POST-FIX STRING WHICH IS CURRENTLY BEING PROCESSED  XZPS2
2453    1 CD4                                                                   XZPS2
2454    1 CD********************                                               XZPS2
2455    1 CD5                                                                   XZPS2
2456    1 CD5       EXTERNAL REFERENCES                                         XZPS2
2457    1 CD5                                                                   XZPS2
2458    1 CD5       FDS - XPREQ, XPXIT, XRMOV, XZCHR, XZFNC, XZFRE, XZFXD, XZLSS, XZMSG,  XZPS2
2459    1 CD5       XZOPR, XZPCS, XZRET                                         XZPS2
2460    1 CD5                                                                   XZPS2
2461    1 CD5       RTE - IABS, IAND, MAX0                                      XZPS2
2462    1 CD5                                                                   XZPS2
2463    1 CD********************                                               XZPS2
```

```
2465    1 BEGIN XZPS2                                                                                                    XZPS2
2466    2    PERFORM TOC TO RETRIEVE TOC ENTRIES FOR ALL DATA ELEMENTS IN SYMBOL TABLE                                   XZPS2
2467    2    PERFORM DATA1 TO RETRIEVE DATA FOR NON-SUBSCRIPTED FIXED-TYPE DATA ELEMENTS                                XZPS2
2468    2    INITIALIZE RANGE VALUES FOR ITERATION                                                                      XZPS2
2469    2    DO UNTIL ALL RANGES ARE FINISHED                                                                           XZPS2
2470    3       DO UNTIL POLISH STRING IS EVALUATED (STARTING WITH SECOND ENTRY OF STRING)                              XZPS2
2471    4          POP ENTRY FROM POLISH STRING                                                                        XZPS2
2472    4          IF ENTRY IS AN OPERAND, THEN                                                                        XZPS2
2473    5             PERFORM RPUSH TO RETRIEVE DATA IF AVAILABLE AND PUSH ONTO RESULT STACK                           XZPS2
2474    4          ELSE  ENTRY IS AN OPERATOR                                                                          XZPS2
2475    5             IF OPERATOR IS NOT =, THEN                                                                       XZPS2
2476    6                PERFORM EXEVAL TO EVALUATE POLISH STRING                                                      XZPS2
2477    5             ELSE OPERATOR IS =                                                                               XZPS2
2478    6                PERFORM REPLAC TO STORE VALUE INTO OBJECT DATA ELEMENT                                        XZPS2
2479    5             ENDIF                                                                                            XZPS2
2480    4          ENDIF                                                                                               XZPS2
2481    3       END DO                                                                                                 XZPS2
2482    3       PERFORM RNGSET TO DETERMINE CORRECT RANGE VALUES                                                       XZPS2
2483    2    END DO                                                                                                    XZPS2
2484    1 END XZPS2                                                                                                    XZPS2
```

```
2486    1 BEGIN TOC                                                                                    XZPS2
2487    2    DO UNTIL ALL ENTRIES IN SYMBOL TABLE ARE PROCESSED                                        XZPS2
2488    3       IF SYMBOL NOT = 0 (SYMBOL IS DE OR INDEX), THEN                                        XZPS2
2489    4          IF ENTRY IS A DATA ELEMENT (SYMBOL'S FLAG WORD NOT= 1), THEN                        XZPS2
2490    5             CALL XPREQ TO RETRIEVE TOC ENTRY                                                XZPS2
2491    4          ELSE SYMBOL IS A RANGE INDEX                                                        XZPS2
2492    5             SET DATA TYPE TO INTEGER                                                        XZPS2
2493    4          ENDIF                                                                               XZPS2
2494    3       ENDIF                                                                                  XZPS2
2495    2    END DO                                                                                    XZPS2
2496    2    CALL XPREQ WITH A CLOSE BUFFER REQUEST                                                    XZPS2
2497    1 END TOC                                                                                      XZPS2
```

```
2499  1 BEGIN DATA1                                                    XIPS2
2500  2   DO UNTIL ALL ENTRIES IN SYMBOL TABLE ARE PROCESSED           XIPS2
2501  3     IF SYMBOL IS A NON-SUBSCRIPTED FIXED-TYPE DATA ELEMENT, THEN XIPS2
2502  4       CALL XPREQ TO QUEUE REQUEST FOR DATA RETRIEVAL           XIPS2
2503  3     ENDIF                                                      XIPS2
2504  2   END DO                                                       XIPS2
2505  2   CALL XPREQ WITH A CLOSE BUFFER REQUEST                       XIPS2
2506  1 END DATA1                                                      XIPS2
```

```
2508    1 BEGIN RPUSH                                                                   XZPS2
2509    2    IF OPERAND IS A CHARACTER STRING POINTER (OPERAND < 0), THEN                XZPS2
2510    3       PUSH ABSOLUTE VALUE OF OPERAND AND -3 DATA TYPE ONTO RESULT STACK       XZPS2
2511    2    ELSE  OPERAND IS A SYMBOL TABLE INDEX                                      XZPS2
2512    3       CALL XZRET TO RETRIEVE DATA AND DATA TYPE FOR OPERAND                   XZPS2
2513    3       PUSH RETURNED VALUE AND DATA TYPE ONTO RESULT STACK                     XZPS2
2514    2    ENDIF                                                                      XZPS2
2515    1 END RPUSH                                                                     XZPS2
```

```
2517    1 BEGIN EXEVAL                                                                                  XZPS2
2518    2    IF OPERATOR IS A FUNCTION, THEN                                                            XZPS2
2519    3       DETERMINE # OPERANDS FROM FUNCTION TABLE                                                XZPS2
2520    2    ELSE                                                                                       XZPS2
2521    3       DETERMINE # OPERANDS FROM SYNTAX TABLE                                                  XZPS2
2522    2    ENDIF                                                                                      XZPS2
2523    2    DO FOR EACH OPERAND                                                                        XZPS2
2524    3       IF RESULT STACK CONTAINS A SYMBOL INDEX, THEN                                           XZPS2
2525    4          ERREXIT IF DATA TYPE IS FREE OR CHARACTER TO :MIXERR:                                XZPS2
2526    4          CALL XPREQ TO RETRIEVE DATA AND TYPE FOR FIRST ELEMENT OF ARRAY                      XZPS2
2527    4          STORE DATA AND TYPE IN RESULT STACK AT THIS ENTRY                                    XZPS2
2528    3       ELSE                                                                                    XZPS2
2529    4          ERREXIT IF DATA TYPE IS FREE OR CHARACTER TO :MIXERR:                                XZPS2
2530    3       ENDIF                                                                                   XZPS2
2531    2    END DO                                                                                     XZPS2
2532    2    IF THE OPERATOR IS A FUNCTION, THEN                                                        XZPS2
2533    3       DO FOR EACH FUNCTION OPERAND                                                            XZPS2
2534    4          SET TARGET TYPE FROM FUNCTION TABLE                                                  XZPS2
2535    4          CALL XZPCS TO POP OPERAND, CONVERT IF NECESSARY, AND SET UP                          XZPS2
2536    3       END DO                                                                                  XZPS2
2537    3       CALL XZFNC TO PERFORM FUNCTION FOR RESULTS                                              XZPS2
2538    3       SET DATA TYPE FOR RESULTS FROM FUNCTION TABLE                                           XZPS2
2539    2    ELSE  THIS IS AN OPERATOR OTHER THAN A FUNCTION OR "="                                     XZPS2
2540    3       DETERMINE TYPE REQUIREMENTS FOR THIS OPERATOR FROM SYNTAX TABLE                         XZPS2
2541    3       CASE (TYPE REQUIREMENT+1) :SAME:, :INTGR:                                               XZPS2

2542    4          :SAME:                                                                               XZPS2
2543    4          IF # OPERANDS > 1, THEN                                                              XZPS2
2544    5             SET TARGET TYPE TO MAX OF TWO DATA TYPES                                          XZPS2
2545    4          ELSE  # OPERANDS = 1                                                                 XZPS2
2546    5             SET TARGET TYPE TO OPERAND'S DATA TYPE                                            XZPS2
2547    4          ENDIF                                                                                XZPS2

2548    4          :INTGR:                                                                              XZPS2
2549    4          SET TARGET TYPE TO INTEGER                                                           XZPS2
2550    3       ENDCASE                                                                                 XZPS2
2551    3       DO FOR EACH OPERAND                                                                     XZPS2
2552    4          CALL XZPCS TO POP OPERAND,CONVERT IT TO TARGET TYPE, AND SET UP                       XZPS2
2553    3       END DO                                                                                  XZPS2
2554    3       CALL XZOPR TO PERFORM ARITHMETIC OPERATION AND PUSH RESULT AND TYPE ONTO STACK           XZPS2
2555    2    ENDIF                                                                                      XZPS2
2556    1 EXIT EXEVAL                                                                                   XZPS2

2557    2    :MIXERR:                                                                                   XZPS2
2558    2    SET MESSAGE TO BE OUTPUT TO "FREE OR CHARACTER DATA ELEMENT FOUND IN AN                    XZPS2
2559    2 *                              EXPRESSION"                                                    XZPS2
2560    2    CALL XZPSG TO OUTPUT MESSAGE TO USER                                                       XZPS2
2561    2    CALL XZLSS TO LIST SYMBOLIC STRING                                                         XZPS2
2562    2    CALL XPXIT TO EXIT PROCESSER                                                               XZPS2
2563    1 END EXEVAL                                                                                    XZPS2
```

```
2565    1 BEGIN REPLAC                                                                                          XZPS2
2566    2    IF THE TOP ENTRY OF THE RESULT STACK IS A SYMBOL TABLE INDEX, THEN                                  XZPS2
2567    3       IF THE DATA TYPE IN THE SYMBOL TABLE IS FIXED, THEN                                             XZPS2
2568    4          CALL XPREQ TO RETRIEVE FIRST ELEMENT OF THE ARRAY                                            XZPS2
2569    4          STORE DATA AND TYPE IN TOP RESULT STACK ENTRY                                                XZPS2
2570    3       ENDIF                                                                                           XZPS2
2571    2    ENDIF                                                                                              XZPS2
2572    2 *                                  0     1     2     3     4     5                                    XZPS2
2573    2    CASE (OBJECT DATA TYPE )  :FREE:,:FIXED:,:FIXED:,:FIXED:,:CHAR:,:CHAR:,                           XZPS2
2574    3 *                                     6     7     8                                                  XZPS2
2575    3 *                                 :CHAR:,:CHAR:,:CHAR:                                               XZPS2

2576    3       :FREE:                                                                                          XZPS2
2577    3       CALL XZFRE TO RETRIEVE DATA AND SET UP FOR STORE                                                XZPS2

2578    3       :FIXED:                                                                                         XZPS2
2579    3       CALL FXDOBJ TO RETRIEVE DATA,CONVERT IF NECESSARY, AND SET UP FOR STORE                         XZPS2

2580    3       :CHAR:                                                                                          XZPS2
2581    3       CALL CHROBJ TO RETRIEVE DATA AND SET UP FOR STORE                                               XZPS2
2582    2    END CASE                                                                                           XZPS2
2583    2    IF TOP ENTRY OF RESULT STACK IS A DISPLACEMENT, THEN                                               XZPS2
2584    3       POP OBJECT'S DISPLACEMENT FROM RESULT STACK                                                     XZPS2
2585    2    ELSE  OBJECT HAS NOT BEEN SUBSCRIPTED                                                              XZPS2
2586    3       SET OBJECT'S DISPLACEMENT TO 0                                                                  XZPS2
2587    2    ENDIF                                                                                              XZPS2
2588    2    BUILD XPREQ REQUEST TO STORE # WORDS CALCULATED INTO OBJECT AT OBJECT'S                            XZPS2
2589    2    DISPLACEMENT                                                                                       XZPS2
2590    2    CALL XPREQ TO STORE DATA IN OBJECT                                                                 XZPS2
2591    1 END REPLAC                                                                                            XZPS2
```

```
2593  1 BEGIN RNGSET                                                    XZPS2
2594  2   DO FOR EACH RANGE UNTIL AN INDEX IS SUCCESSFULLY INCREMENTED OR ALL DEFINED  XZPS2
2595  3 *   RANGES ARE PROCESSED                                        XZPS2
2596  3       IF THE CURRENT VALUE FOR RANGE INDEX IS NOT = TO END LIMIT, THEN  XZPS2
2597  4         INCREMENT RANGE VALUE                                   XZPS2
2598  3       ELSE                                                      XZPS2
2599  4         SET RANGE INDEX VALUE  TO BEGIN VALUE                   XZPS2
2600  3       ENDIF                                                     XZPS2
2601  2   END DO                                                        XZPS2
2602  1 END RNGSET                                                      XZPS2
```

```
2604    1 CD0
2605    1 CD0       FORTRAN CALLING PROCEDURE:                                                                          XZRET
2606    1 CD0                                                                                                           XZRET
2607    1 CD0       CALL XZRET(ENTRY)                                                                                   XZRET
2608    1 CD0                                                                                                           XZRET
2609    1 CD********************                                                                                        XZRET
2610    1 CD1                                                                                                           XZRET
2611    1 CD1       XZRET IS USED BY THE ASSGN ROUTINE XZPS2 TO RETRIEVE THE DATA VALUE                                 XZRET
2612    1 CD1       FOR A GIVEN SYMBOL INDEX FROM THE SYMBOL TABLE AND PUSH IT ONTO THE                                 XZRET
2613    1 CD1       RESULT STACK ALONG WITH ITS DATA TYPE                                                               XZRET
2614    1 CD1                                                                                                           XZRET
2615    1 CD********************                                                                                        XZRET
2616    1 CD2                                                                                                           XZRET
2617    1 CD2       INPUTS                                                                                              XZRET
2618    1 CD2                                                                                                           XZRET
2619    1 CD2       ENTRY - SYMBOL TABLE INDEX CURRENTLY BEING PROCESSED                                                XZRET
2620    1 CD2                                                                                                           XZRET
2621    1 CD2       FROM ASGCOM - SYMTAB,RSLTPT                                                                         XZRET
2622    1 CD2                                                                                                           XZRET
2623    1 CD********************                                                                                        XZRET
2624    1 CD3                                                                                                           XZRET
2625    1 CD3       OUTPUTS TO ASGCOM                                                                                   XZRET
2626    1 CD3                                                                                                           XZRET
2627    1 CD3       RESULT,RSLTPT                                                                                       XZRET
2628    1 CD3                                                                                                           XZRET
2629    1 CD********************                                                                                        XZRET
2630    1 CD5                                                                                                           XZRET
2631    1 CD5       EXTERNAL REFERENCES                                                                                 XZRET
2632    1 CD5                                                                                                           XZRET
2633    1 CD5       FDS - XRMOV                                                                                         XZRET
2634    1 CD5                                                                                                           XZRET
2635    1 CD5       RTE - IAND                                                                                          XZRET
2636    1 CD5                                                                                                           XZRET
2637    1 CD********************                                                                                        XZRET
```

```
2639    1 BEGIN XZRET                                                                                                   XZRET
2640    2    IF SYMBOL ENTRY IS FOR NON-SUBSCRIPTED FIXED-TYPE DATA (DATA ELEMENT OR                                    XZRET
2641    3       RANGE INDEX), THEN                                                                                      XZRET
2642    3       SET RETURN VALUE = VALUE IN SYMBOL TABLE FOR THIS ENTRY                                                 XZRET
2643    3       SET RETURN DATA TYPE = DATA TYPE IN SYMBOL TABLE FOR THIS ENTRY                                         XZRET
2644    2    ELSE  SYMBOL IS CHARACTER OR FREE DATA OR SUBSCRIPTED FIXED DATA                                           XZRET
2645    3       SET RETURN VALUE = SYMBOL TABLE INDEX                                                                   XZRET
2646    3       SET RETURN DATA TYPE = -1                                                                               XZRET
2647    2    ENDIF                                                                                                      XZRET
2648    1 END XZRET                                                                                                     XZRET
```

```
2650   1 CD0      FORTRAN CALLING PROCEDURE                                                    XZSYM
2651   1 CD0                                                                                   XZSYM
2652   1 CD0   CALL XZSYM (TKNPTR, FLAG, SYMIND)                                              XZSYM
2653   1 CD0                                                                                   XZSYM
2654   1 CD********                                                                            XZSYM
2655   1 CD1                                                                                   XZSYM
2656   1 CD1      XZSYM PROVIDES SYMBOL TABLE BUILDING AND ACCESS FOR ASSGN.  IT               XZSYM
2657   1 CD1      USES XZSYT TO PERFORM GENERAL SYMBOL TABLE FUNCTIONS THEN                    XZSYM
2658   1 CD1      ACCOMPLISHES RANGE AND SUBSCRIPT PROCESSING DIRECTLY                         XZSYM
2659   1 CD1                                                                                   XZSYM
2660   1 CD********                                                                            XZSYM
2661   1 CD2      INPUT                                                                        XZSYM
2662   1 CD2   TKNPTR - INDEX INTO INPUT SYMBOLIC STRING (SSTRNG) OF TOKEN TO BE              XZSYM
2663   1 CD2            ENTERED OR LOCATED IN SYMBOL TABLE (SYMTAB)                           XZSYM
2664   1 CD2   FLAG   - SPECIAL PROCESSING FLAG                                               XZSYM
2665   1 CD2            0, NORMAL PROCESSING                                                  XZSYM
2666   1 CD2            1, SYMBOL IS A RANGE                                                  XZSYM
2667   1 CD2            2, SYMBOL IS SUBSCRIPTED                                              XZSYM
2668   1 CD2   ASGCOM COMMON - LU, SSTRNG, STWIDE, SYMTAB, TOKENS,                            XZSYM
2669   1 CD2                                                                                   XZSYM
2670   1 CD********                                                                            XZSYM
2671   1 CD3      OUTPUT                                                                       XZSYM
2672   1 CD3   SYMIND - INDEX INTO SYMBOL TABLE (SYMTAB) WHERE SYMBOL IS LOCATED.             XZSYM
2673   1 CD3            VALUE OF SYMIND INDICATES ENTRY NUMBER, I.E., 1, 2, ...               XZSYM
2674   1 CD3   ASGCOM COMMON - SYMTAB                                                         XZSYM
2675   1 CD3                                                                                   XZSYM
2676   1 CD********                                                                            XZSYM
2677   1 CD4      INTERNAL                                                                     XZSYM
2678   1 CD4   SYMPTR - INDEX TO FLAG WORD OF ENTRY IN SYMTAB. VALUE IS EQUAL TO              XZSYM
2679   1 CD4            SYMIND * STWIDE                                                       XZSYM
2680   1 CD4                                                                                   XZSYM
2681   1 CD********                                                                            XZSYM
2682   1 CD5      EXTERNAL REFERENCES                                                          XZSYM
2683   1 CD5   XPXIT, XZLSS, XZMSG, XZSYT                                                     XZSYM
2684   1 CD5                                                                                   XZSYM
2685   1 CD********                                                                            XZSYM
```

```
2687    1 BEGIN XZSYM                                                                                                   XZSYM
2688    2    CALL XZSYT TO ENTER TOKEN INTO TABLE OR RETURN INDEX TO EXISTING ENTRY                                     XZSYM
2689    2    IF TOKEN IS A NAME, THEN                                                                                   XZSYM
2690    3       IF SPECIAL PROCESSING FLAG IS SET (1=RANGE INDEX, 2=SUBSCRIPTED), THEN                                  XZSYM
2691    4          IF FLAG INDICATES RANGE INDEX, THEN                                                                  XZSYM
2692    5             ERREXIT TO :BADRNG: WITH ERROR AS01 IF OBJECT (FIRST ENTRY IN TABLE)                              XZSYM
2693    5             ERREXIT TO :BADRNG: WITH ERROR AS02 ENTRY IS ALREADY SUBSCRIPTED                                  XZSYM
2694    5             ERREXIT TO :BADRNG: WITH ERROR AS03 ENTRY IS ALREADY A DEFINED RANGE                              XZSYM
2695    4          ENDIF                                                                                                XZSYM
2696    4          SET ENTRY FLAG WORD TO FLAG VALUE                                                                    XZSYM
2697    3       ENDIF                                                                                                   XZSYM
2698    2    ENDIF                                                                                                      XZSYM
2699    2    ADD BIAS OF 256 TO SYMBOL INDEX                                                                            XZSYM
2700    1 EXIT XZSYM                                                                                                    XZSYM

2701    2    :BADRNG:                                                                                                   XZSYM
2702    2    CALL XZMSG TO OUTPUT ERROR DESCRIPTION                                                                     XZSYM
2703    2    CALL XZLSS TO DISPLAY SYMBOLIC STRING AND POINT TO ERROR                                                   XZSYM
2704    2    CALL XPXIT TO TERMINATE PROCESSOR                                                                          XZSYM
2705    1 END XZSYM                                                                                                     XZSYM
```

```
2707    1 CD0      FORTRAN CALLING PROCEDURE                                                    XZSYT
2708    1 CD0                                                                                   XZSYT
2709    1 CD0   CALL XZSYT (TKNPTR, SYMIND)                                                    XZSYT
2710    1 CD0                                                                                   XZSYT
2711    1 CD********                                                                            XZSYT
2712    1 CD1                                                                                   XZSYT
2713    1 CD1   PROVIDED KEY VALUES ARE SUITABLY LOCATED IN COMMON, XZSYT PROVIDES              XZSYT
2714    1 CD1   A GENERAL CAPABILITY FOR SYMBOL TABLE ACCESS FOR TOKENS INPUT VIA             XZSYT
2715    1 CD1   A SYMBOLIC STRING                                                            XZSYT
2716    1 CD1                                                                                   XZSYT
2717    1 CD********                                                                            XZSYT
2718    1 CD2      INPUT                                                                        XZSYT
2719    1 CD2   TKNPTR - INDEX INTO INPUT SYMBOLIC STRING (SSTRNG) OF TOKEN TO BE              XZSYT
2720    1 CD2            ENTERED OR LOCATED IN SYMBOL TABLE (SYMTAB)                         XZSYT
2721    1 CD2   ASGCOM COMMON - LASTSY, SSTRNG, STWIDE, SYMTAB, TOKENS                         XZSYT
2722    1 CD2                                                                                   XZSYT
2723    1 CD********                                                                            XZSYT
2724    1 CD3      OUTPUT                                                                       XZSYT
2725    1 CD3   SYMIND - INDEX INTO SYMBOL TABLE (SYMTAB) WHERE SYMBOL IS LOCATED.             XZSYT
2726    1 CD3            VALUE OF SYMIND INDICATES ENTRY NUMBER , I.E., 1, 2, ...           XZSYT
2727    1 CD3   ASGCOM COMMON - LASTSY, SYMTAB                                               XZSYT
2728    1 CD3                                                                                   XZSYT
2729    1 CD********                                                                            XZSYT
2730    1 CD4      INTERNAL                                                                     XZSYT
2731    1 CD4   DISP   - DISPLACEMENT INTO SYMBOL TABLE ENTRY FOR FIELD TO BE USED             XZSYT
2732    1 CD4            AS KEY.  NAMES USE A VALUE OF 1, CONSTANTS A VALUE OF 8.           XZSYT
2733    1 CD4   EDLOOP - INDEX OF LAST ALLOCATED WORD IN SYMTAB.  VALUE IS EQUAL            XZSYT
2734    1 CD4            TO LASTSY*STWIDE                                                   XZSYT
2735    1 CD4   IBUF   - INTERNAL BUFFER FOR SYMBOL VALUE USED TO ASSURE LAST               XZSYT
2736    1 CD4            WORDS OF INTEGER AND REAL CONSTANTS ARE ZERO.                      XZSYT
2737    1 CD4                                                                                   XZSYT
2738    1 CD********                                                                            XZSYT
2739    1 CD5      EXTERNAL REFERENCES                                                          XZSYT
2740    1 CD5   XRCPR, XRMOV                                                                    XZSYT
2741    1 CD5                                                                                   XZSYT
2742    1 CD********                                                                            XZSYT
2743    1 CD6      SPECIAL REMARKS                                                              XZSYT
2744    1 CD6   THE REQUIRED FORMAT OF EACH SYMBOL TABLE ENTRY IS                               XZSYT
2745    1 CD6                                                                                   XZSYT
2746    1 CD6   ---------------------------------------------------------------------------     XZSYT
2747    1 CD6   I TYPE (1 WD)  I NAME    I UNUSED (4 WORDS)       I VALUE   I OTHER   I      XZSYT
2748    1 CD6   I = TOKEN CODE I (3 WDS) I RESERVED FOR TOC ENTRY I (3 WDS) I (N WDS) I      XZSYT
2749    1 CD6   ---------------------------------------------------------------------------     XZSYT
2750    1 CD6                                                                                   XZSYT
2751    1 CD********                                                                            XZSYT
```

```
2753    1 BEGIN XZSYT                                                                                                    XZSYT
2754    2    CLEAR BUFFER TO BE USED IN MOVING TOKEN                                                                     XZSYT
2755    2    IF TOKEN IS A NAME, THEN                                                                                    XZSYT
2756    3       SET COMPARISON DISPLACEMENT IN TABLE TO 1 (NAME FIELD)                                                   XZSYT
2757    2    ELSE                                                                                                        XZSYT
2758    3       SET COMPARISON DISPLACEMENT IN TABLE TO 8 (VALUE FIELD)                                                  XZSYT
2759    2    ENDIF                                                                                                       XZSYT
2760    2    MOVE TOKEN INTO BUFFER                                                                                      XZSYT
2761    2    START SEARCH UNTIL ALL ALLOCATED SYMBOL TABLE ENTRIES EXAMINED                                              XZSYT
2762    2    EXIT IF ENTRY MATCHES BUFFER CONTENTS AND TYPE FIELD MATCHES TOKEN CODE                                     XZSYT
2763    2    END LOOP                                                                                                    XZSYT
2764    3       STORE TOKEN CODE IN TYPE FIELD OF NEXT ENTRY                                                             XZSYT
2765    3       STORE BUFFER CONTENTS INTO APPROPRIATE FIELD OF ENTRY (NAME OR VALUE)                                    XZSYT
2766    3       INCREMENT NUMBER OF ALLOCATED ENTRIES                                                                    XZSYT
2767    2    END SEARCH                                                                                                  XZSYT
2768    2    SET SYMIND TO ENTRY NUMBER                                                                                  XZSYT
2769    1 END XZSYT                                                                                                      XZSYT
                                                                                                                         XZSYT
```

SYMBOL DEFINITION TABLE

```
:ADD    :  2114
:ADD1   :  1106
:ADD1   :  1105
:ADD1   :  1103
:ADD1   :  1124
:ADD1   :  1112
:ADD1   :  1123
:ADD1   :  1122
:ADD1   :  1102
:ADD1   :  1120
:ADD1   :  1119
:ADD1   :  1118
:ADD1   :  1117
:ADD1   :  1116
:ADD1   :  1115
:ADD1   :  1114
:ADD1   :  1101
:ADD1   :  1100
:ADD1   :  1111
:ADD1   :  1110
:ADD1   :  1109
:ADD1   :  1108
:ADD1   :  1090
:ADD1   :  1094
:ADD1   :  1099
:ADD1   :  1104
:ADD1   :  1107
:ADD1   :  1097
:ADD1   :  1096
:ADD2   :  1126
:ADD2   :  1091
:ADD2   :  1113
:ADD2   :  1121
:ADD3   :  1092
:ADD3   :  1128
:ADD4   :  1095
:ADD4   :  1093
:ADD4   :  1130
 ASSGN     272
:ASTER  :  2314
:BADEQL :  2393
:BADFUN :  2329
:BADRNG :  2701
:CALST  :  1098
:CALST  :  1132
:CHAR   :  2304
:CHAR   :  2580
:CHRFRE :   937
:CHRSTR :   931
:COMERR :  2389
 DATA1    2499
 DBOSP     423
:DBLE   :  2208
 DEFIN     483
:DIVIDE :  2123
:ENDCAS :  2325
:ENDO   :  2382
 ENDSC     641
```

```
:EQ          1364
:ERR1         876
:ERR10       1049
:ERR2         882
:ERR3         885
:ERR4         676
:ERR8        1023
:ERR9        1044
 EXEVAL      2517
:EXIT        1971
:EXPO        2148
:FIVE        1968
:FIXED       2578
:FIXERR       955
:FIXFIX      1831
:FIXFRE      1821
:FLSTER      2391
:FOUR        1967
:FREE        2576
:FREFIX      1779
:FREFRE      1763
:FRESTR      1760
:GE          1372
:GT          1380
:INDEX       2126
:INTG        2204
:INTGR       2548
:INVLD       2327
:LE          1356
:LPAREN      2318
:LT          1348
:MINUS       2310
:MIXERR      2557
:MULT        2120
:NOCHAR      1836
:ONE         1961
:OVER        2215
:OVER        2157
:OVER        1716
:PLUS        2306
:PNERR1       590
 RANGE       2400
:REAL        2206
 REPLAC      2565
 RNGSET      2593
:RNGSYM      2416
 RPUSH       2508
:SAME        2542
 SCAN         814
 SETUP       2273
 SETXY        681
:SIX         1969
 STRING      2336
:SUBSCR      2135
:SUBTR       2117
:SYMBOL      2268
:SYNTX1      2365
:SYNTX2      2387
:THREE       1965
```

```
 TOC       2486
:TWO   :   1963
:UMINUS:   2151
 XZCHR      925
 XZDFT     1083
 XZDIN      999
 XZDMK     1340
 XZDMN      510
 XZDOT     1570
 XZDPN      527
 XZDP1     1271
 XZDP2     1494
 XZFCL     1665
 XZFNC     1711
 XZFRE     1755
 XZFXD     1815
 XZLSS     1938
 XZMSG     2035
 XZOPR     2109
 XZPCS     2200
 XZPS1     2265
 XZPS2     2465
 XZRET     2639
 XZS[illegible]4   2687
 XZSYT     2753
 XZ1SP     1630

@XQT F.PDLIST
```

```
1 CD********
1 CD0
1 CD0    DO - CONDITIONAL ITERATION (LOOPING) PROCESSOR
1 CD0
1 CD0        SCHEDULED BY FDS
1 CD0
1 CD********
1 CD1
1 CD1    THE DO AND ENDDO UTILITY PROCESSOR PAIR PROVIDE FDS USERS WITH THE
1 CD1    CAPABILITY OF LOOPING THROUGH A BLOCK OF SEQUENCE TABLE ENTRIES
1 CD1    EITHER UNTIL A GIVEN CONDITION IS SATISFIED OR WHILE THE CONDITION
1 CD1    IS TRUE.
1 CD1
1 CD********
1 CD2
1 CD2       INPUT FROM MANAGER VIA SCHEDULING PARAMETERS
1 CD2    LU      - LOGICAL UNIT OF USER'S TERMINAL
1 CD2    FLAGS   - DEBUG FLAGS FROM USER SIGN-ON
1 CD2    SEQDSP  - INDEX OF CURRENTLY EXECUTING &SEQTB ENTRY
1 CD2
1 CD2       INPUT FROM INTERFACE TABLE
1 CD2    DOTYPE - 6CH CHARACTER STRING SPECIFYING LOOP CONDITION TYPE
1 CD2             UNTIL  - LOOP UNTIL RELATION BETWEEN OPRND1 AND OPRND2 IS
1 CD2                      TRUE.  TEST IS MADE AT THE END OF EACH LOOP
1 CD2                      EXECUTION
1 CD2             WHILE  - LOOP WHILE RELATION BETWEEN OPRND1 AND OPRND2 IS
1 CD2                      TRUE.  TEST IS MADE PRIOR TO STARTING EACH LOOP
1 CD2                      EXECUTION.
1 CD2    OPRND1 - FIRST REAL QUANTITY TO COMPARE(VALUE NOT RETRIEVED,
1 CD2             CHARACTERISTICS PASSED TO ENDDO)
1 CD2    RELATN - 2CH CHARACTER STRING CONTAINING RELATIONAL SYMBOL
1 CD2             #       - NOT EQUAL
1 CD2             <       - LESS THAN
1 CD2             <= OR =< - LESS THAN OR EQUAL
1 CD2             =       - EQUAL
1 CD2             >= OR => - GREATER THAN OR EQUAL
1 CD2             >       - GREATER THAN
1 CD2    OPRND2 - SECOND REAL QUANTITY TO COMPARE (SEE OPRND1)
1 CD2
1 CD2       INPUT FROM AWA ON REQUEST TO MANAGER
1 CD2    INTBUF - INTERFACE TABLE  (LESS LITERAL AREA)
1 CD2    LITERL - NINE WORD BUFFER FOR HOLDING ORIGINAL INTERFACE TABLE
1 CD2             LITERAL AREA
1 CD2    &DOSTK - SEE OUTPUT DEFINITION
1 CD2    &SEQTB - EXEC COPY OF CURRENTLY EXECUTING SEQUENCE TABLE
1 CD2
1 CD********
1 CD3
1 CD3       OUTPUT TO THE MANAGER VIA RETURN PARAMETERS
1 CD3    RETURN - RETURN CODE TO MANAGER
1 CD3                0 - NORMAL TERMINATION.  CONTINUE SEQUENTIAL
1 CD3                    EXECUTION
1 CD3                3 - NORMAL TERMINATION.  SKIP TO SPECIFIED SEQUENCE
1 CD3                    NUMBER FOR CONTINUED EXECUTION
1 CD3                8 - ABNORMAL TERMINATION.  ABORT SEQUENCE EXECUTION
1 CD3    RESET  - SEQUENCE RESET NUMBER OF ENDDO IF RETURN = 3
1 CD3
1 CD3       INPUT/OUTPUT FROM/TO AWA
1 CD3    &DOSTK - CONTROL INFORMATION FOR DO/ENDDO PROCESSORS.  DIMENSIONED
```

```
 61    1 CD3           (27,N) WHERE N IS THE CURRENT NUMBER OF ACTIVE LOOPS.
 62    1 CD3           EACH ENTRY HAS THE FOLLOWING FORM.
 63    1 CD3           WORDS  1- 7 - FIRST SEVEN WORDS OF DO INTERFACE TABLE
 64    1 CD3                         WITH THE FOLLOWING CHANGES (SEE FDS SD9,
 65    1 CD3                         VOLUME IV, FIGURE 1.2-17)
 66    1 CD3           WORDS  8-14 - INTERFACE TABLE ENTRY FOR OPRND1 (SAME AS
 67    1 CD3                         IN ORIGINAL TABLE EXCEPT FOR POSSIBLE NEW
 68    1 CD3                         VALUES POINTING INTO LITERAL AREA)
 69    1 CD3           WORDS 15-21 - INTERFACE TABLE ENTRY FOR OPRND2 (SEE
 70    1 CD3                         ABOVE)
 71    1 CD3           WORDS 22-25 - LITERAL AREA FOR VALUE(S)/SUBSCRIPT(S) OF
 72    1 CD3                         OPRND1 & 2
 73    1 CD3           WORD     26 - RELATION ID CODE AS FOLLOWS
 74    1 CD3                         0 - #
 75    1 CD3                         1 - >
 76    1 CD3                         2 - >= OR =>
 77    1 CD3                         3 - =
 78    1 CD3                         4 - <= OR =<
 79    1 CD3                         5 - <
 80    1 CD3           WORD     27 - SEQUENCE NUMBER OF TOP OF LOOP
 81    1 CD3
 82    1 CD********
 83    1 CD4
 84    1 CD4              INTERNAL VARIABLES
 85    1 CD4           CODE   - ARRAY OF EIGHT ACCEPTABLE RELATION MNEMONICS AND
 86    1 CD4                    CORRESPONDING INTERNAL CODES
 87    1 CD4           DOENT  - INDEX INTO &DOSTK FOR NEW 27 WORD ENTRY
 88    1 CD4           MRBUFF - 64 WORD MANAGER COMMUNICATIONS BUFFER.  EIGHT WORD
 89    1 CD4                    ENTRIES ARE OF THE FORM
 90    1 CD4                    REQST  - AWA MANAGEMENT REQUEST CODE
 91    1 CD4                    CLASS  - CLASS AND TYPE OF DATA
 92    1 CD4                    NAME   - SIX CHARACTER DATA NAME
 93    1 CD4                    SIZE   - SIZE OF DATA
 94    1 CD4                    DISP   - DISPLACEMENT INTO DATA OF TRANSACTION ORIGIN
 95    1 CD4                    CLASNO - CLASS I/O NUMBER THROUGH WHICH DATA IS
 96    1 CD4                             TRANSMITTED
 97    1 CD4           TOP    - INDEX INTO &SEQTB OF TOP OF LOOP
 98    1 CD4           XPCLS  - CLASS I/O NUMBER FOR MANAGER COMMUNICATIONS (SET BY
 99    1 CD4                    XPGET)
100    1 CD4
101    1 CD********
102    1 CD5
103    1 CD5              REFERENCED ROUTINES
104    1 CD5           EXEC, IAND, MOD, RMPAR, XPGET, XPREQ(XPGET), XPXIT, XRCPR, XRMOV,
105    1 CD5           XRSET, XUDBG, XVPAW, XZMSG
106    1 CD5
107    1 CD5              NOTES
108    1 CD5           DO AND ENDDO MUST BE USED IN PAIRS
109    1 CD5
110    1 CD5           &DOSTK IS ALLOCATED IN THE AWA FOR INTERPROCESSOR COMMUNICATIONS
111    1 CD5
112    1 CD5           XESCN CLEANS UP ANY RESIDUAL &DOSTK AFTER EXECUTION CONTROLER
113    1 CD5           TERMINATION
114    1 CD5
115    1 CD5           THE MAXIMUM NUMBER OF NESTED LOOPS IS 4
116    1 CD5
117    1 CD5           ALL NESTING (STRUCTURE) ERRORS ARE LEFT FOR EXECUTION TIME
118    1 CD5           DETECTION
119    1 CD5
```

222

```
120    1 CDS   MANUAL MODE EXECUTION IS NOT SUPPORTED
121    1 CDS
122    1 CD********
```

```
124   1 BEGIN DO                                                                                                    DO
125   2    CALL XPGET TO INITIALIZE ACCESS TO XPREQ AND TO RETRIEVE DOTYPE AND RELATN                               DO
126   2    CALL XVPAW TO RETRIEVE &DOSTK INTO BUFFER                                                               DO
127   2    IF RETRIEVAL FAILED, THEN                                                                               DO
128   3       INITIALIZE BUFFER FOR BUILDING FIRST &DOSTK ENTRY                                                    DO
129   2    ENDIF                                                                                                   DO
130   2    IF &DOSTK IS NOT FULL (NOT MAXIMUM NUMBER OF NESTS), THEN                                               DO
131   3       IF XPGET INTERFACE TABLE BUFFER INDICATES LITERAL DATA EXIST, THEN                                    DO
132   4          CALL XPREQ TO RETRIEVE LITERALS                                                                   DO
133   3       ENDIF                                                                                                DO
134   3       IF RELATN IS A VALID RELATION OPERATOR, THEN                                                         DO
135   4          SET RELATION CODE IN NEW ENTRY IN BUFFER                                                          DO
136   4          SET INTERFACE TABLE HEADER WITH NAME OF &INTAB AND NUMBER OF PARAMETERS OF 2                      DO
137   4          DO FOR EACH OPRND                                                                                 DO
138   5             MOVE OPRND ENTRY INTO NEW INTERFACE TABLE BUFFER                                               DO
139   5             IF OPRND HAS LITERAL VALUE OR DOUBLE SUBSCRIPTS, THEN                                          DO
140   6                MOVE LITEPAL DATA                                                                           DO
141   6                ADJUST LITERAL POINTERS                                                                     DO
142   5             ENDIF                                                                                          DO
143   4          ENDDO                                                                                             DO
144   4          CALL XPREQ TO RETRIEVE &SEQTB (EXECUTING SEQUENCE TABLE)                                          DO
145   4          IF &SEQTB DISPLACEMENT (SEQDSP) > 0, THEN                                                         DO
146   5             SET TOP OF LOOP TO NEXT SEQUENCE NUMBER IN TABLE (0 IF END OF TABLE)                           DO
147   4          ELSE INSERTED COMMAND                                                                             DO
148   4          EXIT TO :ERR02: IF SEQUENCE NUMBER IS ZERO (MANUAL)                                               DO
149   5             LOCATE ORIGINAL SEQUENCE ENTRY                                                                 DO
150   5             IF ORIGINAL ENTRY WAS ALSO A DO (OVERRIDE CONDITION), THEN                                     DO
151   6                SET TOP OF LOOP TO NEXT SEQUENCE NUMBER IN TABLE (OR ZERO)                                  DO
152   5             ELSE (INSERT)                                                                                  DO
153   6                SET TOP OF LOOP TO CURRENT NUMBER                                                           DO
154   5             ENDIF                                                                                          DO
155   4          ENDIF                                                                                             DO
156   4          CASE (:WHILE:, :UNTIL:, :OTHER:) DOTYPE                                                           DO

157   5             :WHILE:                                                                                        DO
158   5             INVERT RELATION CODE                                                                           DO
159   5             INITIALIZE NEST COUNTER TO 1                                                                   DO
160   5             START SEARCH FROM TOP OF LOOP ENTRY UNTIL ALL ENTRIES HAVE BEEN EXAMINED                       DO
161   6                IF COMMAND IS ENDDO, THEN                                                                   DO
162   7                   DECREMENT NEST COUNTER                                                                   DO
163   6                ELSE                                                                                        DO
164   7                   IF COMMAND IS ANOTHER DO, THEN                                                           DO
165   8                      INCREMENT  NEST COUNTER                                                               DO
166   7                   ENDIF                                                                                    DO
167   6                ENDIF                                                                                       DO
168   5             EXIT IF NEST COUNTER IS ZERO                                                                   DO
169   6                SET RESET NUMBER TO CURRENT SEQUENCE NUMBER (ENDDO JUST FOUND)                              DO
170   5             END LOOP                                                                                       DO
171   5             EXIT TO :ERR04: FOR NO MATCHING ENDDO                                                          DO
172   5             END SEARCH                                                                                     DO

173   5             :UNTIL:                                                                                        DO
174   5             CLEAR RESET NUMBER (CONTINUE SEQUENTIAL EXECUTION)                                             DO

175   5             :OTHER:                                                                                        DO
176   5             TERMINATE WITH ERR07 FOR UNRECOGNIZED DOTYPE                                                   DO
177   4          END CASE                                                                                          DO
178   4          CALL XPREQ TO OUTPUT NEW EXPANDED &DOSTK                                                          DO
179   3       ELSE INVALID RELATION                                                                                DO
```

```
180  4        TERMINATE WITH ERR05 FOR INVALID RELATION
181  3     ENDIF
182  2  ELSE &DOSTK OVERFLOW
183  3     TERMINATE WITH ERR01 FOR &DOSTK FULL
184  2  ENDIF
185  1  EXIT DO

186  2  :ERR02: TERMINATE FOR EXECUTING IN MANUAL MODE

187  2  :ERR04: TERMINATE FOR NO MATCHING ENDDO FOUND DURING WHILE PROCESSING
188  1  END DO
```

```
190   1 CD********                                                                  ELSE
191   1 CD0                                                                         ELSE
192   1 CD0       FORTRAN MAIN PROGRAM - STANDARD FDS SCHEDULING PARAMETERS         ELSE
193   1 CD0                                                                         ELSE
194   1 CD********                                                                  ELSE
195   1 CD1                                                                         ELSE
196   1 CD1       THE ELSE UTILITY PROCESSOR LOCATES THE ENDIF CORRESPONDING TO     ELSE
197   1 CD1       THIS ELSE AND SETS THE SEQUENCE RESET NUMBER (VIA XPXIT) SO       ELSE
198   1 CD1       THE PROCESSOR FOLLOWING ENDIF IS EXECUTED NEXT IN THE SEQUENCE.   ELSE
199   1 CD1                                                                         ELSE
200   1 CD********                                                                  ELSE
201   1 CD2                                                                         ELSE
202   1 CD2       INPUT                                                             ELSE
203   1 CD2    SCHEDULING PARAMETERS                                                ELSE
204   1 CD2        LU       - LOGICAL UNIT NO. OF THIS FDS USER                     ELSE
205   1 CD2        FLAGS    - DEBUG FLAGS -- BIT 11 ON WILL CAUSE DEBUG PRINT       ELSE
206   1 CD2                                                                         ELSE
207   1 CD********                                                                  ELSE
208   1 CD3                                                                         ELSE
209   1 CD3       OUTPUT                                                            ELSE
210   1 CD3                                                                         ELSE
211   1 CD3    RPARMS - RETURN PARAMETERS SENT TO FDS MANAGER VIA XPXIT             ELSE
212   1 CD3     (1)    = 3 => PROCESSOR EXECUTION SEQUENCE TO CONTINUE AT THE SEQUENCE ELSE
213   1 CD3                NO. GIVEN IN RPARMS(2)                                   ELSE
214   1 CD3           = 8 => ABNORMAL TERMINATION OF THE PROCESSOR EXECUTION SEQUENCE ELSE
215   1 CD3     (2)    = SEQUNCE NO. TO BE EXECUTED NEXT WHEN RMPARMS(1) = 3       ELSE
216   1 CD3                                                                         ELSE
217   1 CD********                                                                  ELSE
218   1 CD5                                                                         ELSE
219   1 CD5       REFERENCE ROUTINES                                                ELSE
220   1 CD5                                                                         ELSE
221   1 CD5    RMPAR                                                                ELSE
222   1 CD5    XPXIT                                                                ELSE
223   1 CD5    XREXT                                                                ELSE
224   1 CD5    XUDBG                                                                ELSE
225   1 CD5    XZFCL                                                                ELSE
226   1 CD5    XZSCM                                                                ELSE
227   1 CD5                                                                         ELSE
228   1 CD********                                                                  ELSE
```

```
230  1 BEGIN ELSE                                                    ELSE
231  2     CALL RMPAR TO GET SCHEDULING PARAMETERS                   ELSE
232  2     CALL XZFCL TO ESTABLISH FDS MANAGER'S CLASS NO. (XPCLS)   ELSE
233  2     CALL XZSCH TO SEARCH FOR MATCHING ENDIF COMMAND           ELSE
234  2     CALL XPXIT TO RETURN PARAMETERS TO FDS MANAGER            ELSE
235  1 END ELSE
```

```
237   1 CD********                                                                  ENDDO
238   1 CDO                                                                         ENDDO
239   1 CDO    ENDDO - LOOP TERMINATION PROCESSOR                                   ENDDO
240   1 CDO                                                                         ENDDO
241   1 CDO         SCHEDULED BY FDS                                                ENDDO
242   1 CDO                                                                         ENDDO
243   1 CD********                                                                  ENDDO
244   1 CD1                                                                         ENDDO
245   1 CD1    THE DO AND ENDDO UTILITY PROCESSOR PAIR PROVIDE FDS USERS WITH THE   ENDDO
246   1 CD1    CAPABILITY OF LOOPING THROUGH A BLOCK OF SEQUENCE TABLE ENTRIES      ENDDO
247   1 CD1    EITHER UNTIL A GIVEN CONDITION IS SATISFIED OR WHILE THE CONDITION   ENDDO
248   1 CD1    IS TRUE.                                                             ENDDO
249   1 CD1                                                                         ENDDO
250   1 CD********                                                                  ENDDO
251   1 CD2                                                                         ENDDO
252   1 CD2       INPUT FROM MANAGER VIA SCHEDULING PARAMETERS                      ENDDO
253   1 CD2    LU     - LOGICAL UNIT OF USER'S TERMINAL                            ENDDO
254   1 CD2    FLAGS  - DEBUG FLAGS FROM USER SIGN-ON                              ENDDO
255   1 CD2                                                                         ENDDO
256   1 CD2       INPUT/OUTPUT FROM/TO AWA                                          ENDDO
257   1 CD2    &DOSTK - CONTROL INFORMATION FOR DO/ENDDO PROCESSORS.  DIMENSIONED   ENDDO
258   1 CD2             (27,N) WHERE N IS THE CURRENT NUMBER OF ACTIVE LOOPS.       ENDDO
259   1 CD2             SEE DO PROCESSOR FOR CONTENTS                               ENDDO
260   1 CD2                                                                         ENDDO
261   1 CD2    THE FDS SYSTEM STATUS TABLE, XVSTB, IS ACCESSED FOR THE MANAGER/     ENDDO
262   1 CD2    PROCESSOR CLASS I/O COMMUNICATIONS NUMBER                            ENDDO
263   1 CD2                                                                         ENDDO
264   1 CD********                                                                  ENDDO
265   1 CD3                                                                         ENDDO
266   1 CD3       OUTPUT TO THE MANAGER VIA RETURN PARAMETERS                       ENDDO
267   1 CD3    RETURN - RETURN CODE TO MANAGER                                      ENDDO
268   1 CD3                   0 - NORMAL TERMINATION.  CONTINUE SEQUENTIAL         ENDDO
269   1 CD3                       EXECUTION                                        ENDDO
270   1 CD3                   3 - NORMAL TERMINATION.  SKIP TO SPECIFIED SEQUENCE  ENDDO
271   1 CD3                       NUMBER FOR CONTINUED EXECUTION                   ENDDO
272   1 CD3                   8 - ABNORMAL TERMINATION.  ABORT SEQUENCE EXECUTION  ENDDO
273   1 CD3    RESET  - SEQUENCE RESET NUMBER OF TOP OF LOOP IF RETURN = 3          ENDDO
274   1 CD3                                                                         ENDDO
275   1 CD********                                                                  ENDDO
276   1 CD4                                                                         ENDDO
277   1 CD4       INTERNAL VARIABLES                                                ENDDO
278   1 CD4    DOENT  - INDEX INTO &DOSTK FOR LAST 27 WORD ENTRY                    ENDDO
279   1 CD4    MRBUFF - 64 WORD MANAGER COMMUNICATIONS BUFFER.  EIGHT WORD          ENDDO
280   1 CD4             ENTRIES ARE OF THE FORM                                     ENDDO
281   1 CD4             REQST  - AWA MANAGEMENT REQUEST CODE                        ENDDO
282   1 CD4             CLASS  - CLASS AND TYPE OF DATA                             ENDDO
283   1 CD4             NAME   - SIX CHARACTER DATA NAME                            ENDDO
284   1 CD4             SIZE   - SIZE OF DATA                                       ENDDO
285   1 CD4             DISP   - DISPLACEMENT INTO DATA OF TRANSACTION ORIGIN       ENDDO
286   1 CD4             CLASNO - CLASS I/O NUMBER THROUGH WHICH DATA IS             ENDDO
287   1 CD4                      TRANSMITTED                                        ENDDO
288   1 CD4    XPCLS  - CLASS I/O NUMBER FOR MANAGER COMMUNICATIONS                 ENDDO
289   1 CD4                                                                         ENDDO
290   1 CD********                                                                  ENDDO
291   1 CD5                                                                         ENDDO
292   1 CD5       REFERENCED ROUTINES                                               ENDDO
293   1 CD5    EXEC, IAND, RMPAR, XPGET, XPREQ(XPGET), XPXIT, XRLOC, XRMOV,         ENDDO
294   1 CD5    XUDBG, XVPAW, XZMSG                                                  ENDDO
295   1 CD5                                                                         ENDDO
```

E0000

296 1 C0********

```
298   1 BEGIN ENDDO                                                                                                    ENDDO
299   2    FIND CLASS I/O NUMBER FOR MANAGER COMMUNICATIONS IN XVSTB                                                   ENDDO
300   2    CALL XVPAW TO RETRIEVE &DOSTK                                                                              ENDDO
301   2    IF RETRIEVAL WAS SUCCESSFUL, THEN                                                                          ENDDO
302   3       WRITE &INTAB TO CLASS I/O FROM LAST &DOSTK ENTRY                                                        ENDDO
303   3       WRITE NON-LITERAL PORTION OF &INTAB TO CLASS I/O (LEAVE FOR XPGET)                                      ENDDO
304   3       CALL XVPAW TO RESTORE &INTAB FROM FIRST CLASS BUFFER INTO AWA                                           ENDDO
305   3       IF RESTORE SUCCESSFUL, THEN                                                                            ENDDO
306   4          CALL XPGET TO RETRIEVE OPRNDS                                                                       ENDDO
307   4          CALL XZEVL TO EVALUATE RELATION                                                                     ENDDO
308   4          IF RELATION IS TRUE, THEN                                                                           ENDDO
309   5             CLEAR RESET NUMBER (CONTINUE SEQUENTIAL EXECUTION)                                               ENDDO
310   5             IF THIS IS LAST &DOSTK ENTRY, THEN                                                               ENDDO
311   6                DELETE &DOSTK FROM AWA                                                                        ENDDO
312   5             ELSE                                                                                             ENDDO
313   6                CALL XPREQ TO REPLACE &DOSTK LESS LAST ENTRY                                                  ENDDO
314   5             ENDIF                                                                                            ENDDO
315   4          ELSE                                                                                                ENDDO
316   5             SET RESET NUMBER TO TOP OF LOOP                                                                  ENDDO
317   4          ENDIF                                                                                               ENDDO
318   3       ELSE                                                                                                   ENDDO
319   4          TERMINATE WITH ERR06 FOR AWA OVERFLOW                                                               ENDDO
320   3       ENDIF                                                                                                  ENDDO
321   2    ELSE                                                                                                      ENDDO
322   3       TERMINATE WITH ERR03 FOR ENDDO WITH OUT DO                                                             ENDDO
323   2    ENDIF                                                                                                     ENDDO
324   1 END ENDDO                                                                                                    ENDDO
```

```
326    1 CD********                                                                                                    ENDIF
327    1 CD0                                                                                                           ENDIF
328    1 CD0       FORTRAN MAIN PROGRAM - STANDARD FDS SCHEDULING PARAMETERS                                           ENDIF
329    1 CD0                                                                                                           ENDIF
330    1 CD********                                                                                                    ENDIF
331    1 CD1                                                                                                           ENDIF
332    1 CD1       THE ENDIF UTILITY PROCESSOR RETURNS TO THE FDS MANAGER (VIA XPXIT) SO                               ENDIF
333    1 CD1       THAT NORMAL PROCESSOR EXECUTION SEQUENCE MAY CONTINUE                                               ENDIF
334    1 CD1                                                                                                           ENDIF
335    1 CD********                                                                                                    ENDIF
336    1 CD2                                                                                                           ENDIF
337    1 CD2       INPUTS                                                                                              ENDIF
338    1 CD2                                                                                                           ENDIF
339    1 CD2    SCHEDULING PARAMETERS                                                                                  ENDIF
340    1 CD2      LU     = LOGICAL UNIT NO. OF FDS USER                                                               ENDIF
341    1 CD2      FLAGS  = DEBUG FLAGS -- BIT 11 ON WILL CAUSE DEBUG PRINT                                            ENDIF
342    1 CD2                                                                                                           ENDIF
343    1 CD********                                                                                                    ENDIF
344    1 CD3                                                                                                           ENDIF
345    1 CD3       OUTPUTS                                                                                             ENDIF
346    1 CD3                                                                                                           ENDIF
347    1 CD3    RMPARMS - RETURN PARAMETERS FOR FDS MANAGER VIA XPXIT                                                  ENDIF
348    1 CD3     (1) = 0 => CONTINUE NORMAL PROCESSOR EXECUTION SEQUENCE                                               ENDIF
349    1 CD3                                                                                                           ENDIF
350    1 CD********                                                                                                    ENDIF
351    1 CD5                                                                                                           ENDIF
352    1 CD5       ROUTINES USED                                                                                       ENDIF
353    1 CD5                                                                                                           ENDIF
354    1 CD5    RMPAR                                                                                                  ENDIF
355    1 CD5    XREXT                                                                                                  ENDIF
356    1 CD5    XPXIT                                                                                                  ENDIF
357    1 CD5    XUDBG                                                                                                  ENDIF
358    1 CD5                                                                                                           ENDIF
359    1 CD********                                                                                                    ENDIF
```

```
361    1 BEGIN ENDIF
362    2    CALL RMPAR TO GET SCHEDULING PARAMETERS FROM FDS MANAGER
363    2    CALL XPXIT TO RETURN TO FDS MANAGER
364    1 END ENDIF
```

```
ENDIF
ENDIF
ENDIF
ENDIF
```

```
366    1 CD********                                                                    IF
367    1 CD0                                                                           IF
368    1 CD0       FORTRAN MAIN PROGRAM - STANDARD FDS SCHEDULING PARAMETERS           IF
369    1 CD0                                                                           IF
370    1 CD********                                                                    IF
371    1 CD1                                                                           IF
372    1 CD1       THE IF UTILITY PROCESSOR COMPARES TWO INPUT QUANTITIES ACCORDING TO A IF
373    1 CD1       SPECIFIED RELATION. WHEN THE COMPARISON IS FALSE, THE SEQUENCE RESET IF
374    1 CD1       NUMBER IS SET (VIA XPXIT) TO THE PROCESSOR FOLLOWING THE CORRESPONDING IF
375    1 CD1       ELSE (OR ENDIF).    IF THE COMPARISION IS TRUE THE NORMAL PROCESSOR IF
376    1 CD1       EXECUTION SEQUENCE IS FOLLOWED.                                     IF
377    1 CD1                                                                           IF
378    1 CD********                                                                    IF
379    1 CD2                                                                           IF
380    1 CD2       INPUT                                                               IF
381    1 CD2    SHEDULING PARAMETERS                                                   IF
382    1 CD2        LU     - LOGICAL UNIT NO. OF THIS FDS USER                        IF
383    1 CD2        FLAGS  - DEBUG FLAGS -- BIT 11 ON WILL CAUSE DEBUG PRINT          IF
384    1 CD2                                                                          IF
385    1 CD2    INTERFACE TABLE PARAMETERS                                            IF
386    1 CD2        OPRND1 - FIRST REAL VALUE TO COMPARE                              IF
387    1 CD2        RELATN - RELATION IDENTIFIER (>,>=,=>,=,<=,=<,<,8)                IF
388    1 CD2        OPRND2 - SECOND REAL VALUE TO COMPARE                             IF
389    1 CD2                                                                          IF
390    1 CD********                                                                   IF
391    1 CD3                                                                          IF
392    1 CD3       OUTPUT                                                             IF
393    1 CD3                                                                          IF
394    1 CD3    RPARMS - RETURN PARAMETERS SENT TO FDS MANAGER VIA XPXIT               IF
395    1 CD3       (1) = 0 => CONTINUE WITH NORMAL PROCESSOR EXECUTION SEQUENCE        IF
396    1 CD3           = 3 => PROCESSOR EXECUTION SEQUENCE TO CONTINUE AT THE SEQUENCE IF
397    1 CD3                  NO. GIVEN IN RPARMS(2)                                  IF
398    1 CD3           = 8 => ABNORMAL TERMINATION OF THE PROCESSOR EXECUTION SEQUENCE IF
399    1 CD3       (2) = SEQUENCE NO. TO BE EXECUTED NEXT WHEN RPARMS(1) = 3           IF
400    1 CD3                                                                          IF
401    1 CD********                                                                   IF
402    1 CD5                                                                          IF
403    1 CD5       REFERENCED ROUTINES                                                IF
404    1 CD5                                                                          IF
405    1 CD5    RMPAR                                                                 IF
406    1 CD5    XPGET                                                                 IF
407    1 CD5    XPXIT                                                                 IF
408    1 CD5    XZEVL                                                                 IF
409    1 CD5    XZSCH                                                                 IF
410    1 CD5    XREXT                                                                 IF
411    1 CD5    XUDBG                                                                 IF
412    1 CD5                                                                          IF
413    1 CD********                                                                   IF
```

```
415    1 BEGIN IF                                                                                                    IF
416    2    CALL RMPAR TO GET INPUT (SHEDULING) PARAMETERS                                                           IF
417    2    CALL XPGET TO RETRIEVE VALUES FOR INTERFACE TABLE INPUTS                                                 IF
418    2    VERIFY RELATIONAL OPERATOR INPUT AS VALID AND TRANSLATE IT TO A CODE                                     IF
419    2    ERREXIT IF RELATIONAL OPERATOR INVALID :ERR03:                                                           IF
420    2    CALL XZEVL TO EVALUATE THE RELATIONAL EXPRESSION                                                         IF
421    2    IF THE EXPRESSION IS FALSE, THEN                                                                         IF
422    3       CALL XISCH TO LOCATE THE ELSE OR ENDIF CORRESPONDING TO THIS IF                                       IF
423    3       AND SET SEQUENCE RESET NUMBER                                                                         IF
424    2    ELSE                                                                                                     IF
425    3       CLEAR SEQUENCE RESET NUMBER FOR NORMAL CONTINUATION OF THE SEQUENCE                                   IF
426    2    ENDIF                                                                                                    IF
427    1 EXIT IF                                                                                                     IF

428    2    :ERR03: CALL XZMSG 'INVALID RELATIONAL OPERATOR - MUST BE #,>=,=>,>,=,<,<=,OR =>                         IF
429    1 END IF                                                                                                      IF
```

```
431   1 CD********
432   1 CD0
433   1 CD0       FORTRAN CALLING PROCEDURE
434   1 CD0    LVALUE = XZEVL (OPRND1, RELATN, OPRND2)
435   1 CD0
436   1 CD********
437   1 CD1
438   1 CD1    XZEVL COMPARES TWO REAL SINGLE PRECISION VALUES UNDER A SPECIFIED
439   1 CD1    RELATION AND RETURNS A FUNCTION VALUE OF LOGICAL TRUE OR FALSE.
440   1 CD1
441   1 CD********
442   1 CD2
443   1 CD2       INPUT
444   1 CD2    OPRND1 - FIRST REAL VALUE TO COMPARE
445   1 CD2    RELATN - RELATION CODE AS FOLLOWS
446   1 CD2             0 - NOT EQUAL
447   1 CD2             1 - GREATER THAN
448   1 CD2             2 - GREATER THAN OR EQUAL
449   1 CD2             3 - EQUAL
450   1 CD2             4 - LESS THAN OR EQUAL
451   1 CD2             5 - LESS THAN
452   1 CD2    OPRND2 - SECOND REAL VALUE TO COMPARE
453   1 CD2
454   1 CD********
455   1 CD3
456   1 CD3       OUTPUT
457   1 CD3    XZEVL  - FUNCTION VALUE OF LOGICAL TRUE OR FALSE
458   1 CD3
459   1 CD********
460   1 CD4
461   1 CD4       INTERNAL VARIABLES
462   1 CD4    TTABLE - BIT MASK REPRESENTING TRUTH TABLE VALUES DERIVED AS
463   1 CD4             FOLLOWS
464   1 CD4                          OPRND1 - OPRND2                    OPRND1 - OPRND2
465   1 CD4             CODE RELATION   +   0   -        RELATION    +   0   -
466   1 CD4             ---------------------------      ------------------------
467   1 CD4               0     #       T   F   T        #, NOT=     T   F   T
468   1 CD4               1     >       T   F   F    OR  >, NOT<=    T   F   F
469   1 CD4               2     >=      T   T   F        >=, NOT<    T   T   F
470   1 CD4               3     =       F   T   F
471   1 CD4               4     <=      F   T   T     (BY COMPLEMENTING CODE > 2)
472   1 CD4               5     <       F   F   T
473   1 CD4             ---------------------------
474   1 CD4
475   1 CD4             THEREFORE FOR OPRND1 - OPRND2 POSITIVE (BITS 0-2) 1, 0, 1
476   1 CD4                       FOR OPRND1 - OPRND2 ZERO     (BITS 3-5) 0, 1, 1
477   1 CD4                   AND FOR OPRND1 - OPRND2 NEGATIVE (BITS 6-8) 1, 0, 0
478   1 CD4             (BITS 9-15 ARE NOT USED) IF RELATN IS 0, 1, OR 2 OR THE
479   1 CD4             COMPLEMENT IF RELATN IS 3, 4, OR 5
480   1 CD4
481   1 CD********
482   1 CD5
483   1 CD5       REFERENCED ROUTINES
484   1 CD5    XREXT
485   1 CD5
486   1 CD5    NO CHECKS FOR VALID RELATION CODES OR OVERFLOW/UNDER FLOW ARE MADE
487   1 CD5
488   1 CD********
```

```
490   1 BEGIN XZEVL
491   2    CASE (:♦:, :0:, :-:) DIFFERENCE OF OPRND1 AND OPRND2                                                    XZEVL
492   3       :♦: SET FIELD OFFSET TO ZERO (BITS 0-2 OF TTABLE)                                                    XZEVL
493   3       :0: SET FIELD OFFSET TO THREE (BITS 3-5 OF TTABLE)                                                   XZEVL
494   3       :-: SET FIELD OFFSET TO SIX (BITS 6-8 OF TTABLE)                                                     XZEVL
495   2    END CASE                                                                                                XZEVL
496   2    IF RELATN > 2 (BOTTOM OF TRUTH TABLE), THEN                                                             XZEVL
497   3       COMPLEMENT TTABLE                                                                                    XZEVL
498   3       DECREMENT RELATN BY 3                                                                                XZEVL
499   2    ENDIF                                                                                                   XZEVL
500   2    ADD RELATN TO FIELD OFFSET (INDEXES TO CORRECT BIT FOR RELATN AND DIFFERENCE)                           XZEVL
501   2    SET FUNCTION VALUE TO INDEXED BIT OF TTABLE                                                             XZEVL
502   1 END XZEVL                                                                                                  XZEVL
```

```
504    1 CD********                                                                                    XZSCN
505    1 CD0                                                                                           XZSCN
506    1 CD0       FORTRAN CALLING PROCEEDURE                                                          XZSCN
507    1 CD0                                                                                           XZSCN
508    1 CD0    CALL XZSCN (SRCHFG, RPARMS)                                                            XZSCN
509    1 CD0                                                                                           XZSCN
510    1 CD********                                                                                    XZSCN
511    1 CD1                                                                                           XZSCN
512    1 CD1       XZSCN IS CALLED TO LOCATE THE CORRESPONDING ELSE/ENDIF IN THE CURRENT                XZSCN
513    1 CD1       SEQUENCE TABLE. THE RETURN PARAMETERS FOR THE FDS MANAGER ARE OUTPUT                 XZSCN
514    1 CD1                                                                                           XZSCN
515    1 CD********                                                                                    XZSCN
516    1 CD2                                                                                           XZSCN
517    1 CD2       INPUTS                                                                              XZSCN
518    1 CD2                                                                                           XZSCN
519    1 CD2    CALLING SEQUENCE                                                                       XZSCN
520    1 CD2                                                                                           XZSCN
521    1 CD2       SRCHFG - FLAG INDICATING ORIGINATION OF THIS CALL                                   XZSCN
522    1 CD2                = 0 => CALLED BY IF TO FIND MATCHING ELSE OR ENDIF                         XZSCN
523    1 CD2                = 1 => CALLED BY ELSE TO FIND MATCHING ENDIF                               XZSCN
524    1 CD2                                                                                           XZSCN
525    1 CD2    COMMON                                                                                 XZSCN
526    1 CD2                                                                                           XZSCN
527    1 CD2       DEBUG  - FLAG INDICATING WHETHER ONLINE DEBUG TO BE OUTPUT                          XZSCN
528    1 CD2                = 0 => NO DEBUG                                                            XZSCN
529    1 CD2                > 0 => DEBUG                                                               XZSCN
530    1 CD2                                                                                           XZSCN
531    1 CD2       PARMS  - SCHEDULIN PARAMETERS FROM THE FDS MANAGER                                  XZSCN
532    1 CD2        (1)   - LOGICAL UNIT NO. OF THE FDS USER                                           XZSCN
533    1 CD2        (5)   - SEQDSP, INDEX INTO SEQUENCE TABLE (&SEQTAB) OF THE CURRENT                 XZSCN
534    1 CD2                COMMAND                                                                    XZSCN
535    1 CD2                                                                                           XZSCN
536    1 CD********                                                                                    XZSCN
537    1 CD3                                                                                           XZSCN
538    1 CD3       OUTPUTS                                                                             XZSCN
539    1 CD3                                                                                           XZSCN
540    1 CD3    CALLING SEQUENCE                                                                       XZSCN
541    1 CD3                                                                                           XZSCN
542    1 CD3       RPARMS - PARAMETERS TO BE RETURNED TO FDS MANAGER VIA XPXIT                         XZSCN
543    1 CD3        (1)   - = 0 => CONTINUE NORMAL PROCESSOR EXECUTION SEQUENCE                        XZSCN
544    1 CD3                = 3 => EXECUTE SEQUENCE NO. GIVEN BY RPARMS(2) NEXT                        XZSCN
545    1 CD3                = 8 => ABNORMALLY TERMINATE PROCESSOR EXECUTION SEQUENCE                   XZSCN
546    1 CD3        (2)   - SEQUENCE NO. TO BE EXECUTED NEXT IF RPARMS(1) = 3                          XZSCN
547    1 CD3                                                                                           XZSCN
548    1 CD********                                                                                    XZSCN
549    1 CD5       ROUTINES USED                                                                       XZSCN
550    1 CD5                                                                                           XZSCN
551    1 CD5    EXEC                                                                                   XZSCN
552    1 CD5    RMPAR                                                                                  XZSCN
553    1 CD5    XVPAM                                                                                  XZSCN
554    1 CD5    XUDBG                                                                                  XZSCN
555    1 CD5    XRCPR                                                                                  XZSCN
556    1 CD5    XZMSG                                                                                  XZSCN
557    1 CD5                                                                                           XZSCN
558    1 CD********                                                                                    XZSCN
```

```
560   1 BEGIN XZSCH                                                                                                    XZSCH
561   2    RETRIEVE &SEQTA3 FROM THE AWA USING XVPAW                                                                   XZSCH
562   2    STARTSEARCH UNTIL ALL COMMANDS IN &SEQTAB                                                                   XZSCH
563   2    EXIT IF CURRENT COMMAND IS FOUND                                                                            XZSCH
564   3       SET NUMBER OF IF NESTS TO 1                                                                              XZSCH
565   3       STARTSEARCH FROM NEXT COMMAND IN &SEQTAB UNTIL ALL FOLLOWING COMMANDS                                    XZSCH
566   4          IF COMMAND IS FOR ENDIF PROCESSOR, THEN                                                               XZSCH
567   5             DECREMENT NUMBER OF IF NESTS BY 1                                                                  XZSCH
568   4          ELSE                                                                                                  XZSCH
569   5             IF COMMAND IS FOR IF PROCESSOR, THEN                                                               XZSCH
570   6                INCREMENT NUMBER OF IF NESTS BY 1                                                               XZSCH
571   5             ELSE                                                                                               XZSCH
572   6                IF CALLED BY IF PROCESSOR, AND                                                                  XZSCH
573   7                   COMMAND IS FOR ELSE PROCESSOR, THEN                                                          XZSCH
574   7                   ERREXIT IF THIS IS THE END OF &SEQTAB :ERR01:                                                XZSCH
575   7                   IF NUMBER OF IF NESTS IS 1, THEN                                                             XZSCH
576   8                      DECREMENT NUMBER OF IF NESTS TO 0                                                         XZSCH
577   7                   ENDIF                                                                                        XZSCH
578   6                ENDIF                                                                                           XZSCH
579   5             ENDIF                                                                                              XZSCH
580   4          ENDIF                                                                                                 XZSCH
581   3       EXIT IF NUMBER OF IF NESTS IS 0                                                                          XZSCH
582   4          SET SEQUENCE RESET NUMBER (RPARMS(2)) TO BE SEQUENCE NUMBER OF THE                                    XZSCH
583   4          NEXT COMMAND IN THE TABLE                                                                             XZSCH
584   3       ENDLOOP                                                                                                  XZSCH
585   4          ERREXIT :ERR01:                                                                                       XZSCH
586   3       ENDSEARCH                                                                                                XZSCH
587   2    ENDLOOP                                                                                                     XZSCH
588   3       ERREXIT :ERR04:                                                                                          XZSCH
589   2    ENDSEARCH                                                                                                   XZSCH
590   1 EXIT XZSCH                                                                                                     XZSCH

591   2    :ERR01: CALL XZMSG - 'IF CANNOT BE EXECUTED WITHOUT MATCHING ENDIF'                                         XZSCH

592   2    :ERR04: CALL XZMSG - 'SYSTEM ERROR - NO &SEQTAB'                                                            XZSCH
593   1 END XZSCH                                                                                                      XZSCH
```

```
SYMBOL DEFINITION TABLE
  DO          124
  ELSE        230
  ENDDO       298
  ENDIF       361
  :ERR01 : :  591
  :ERR02 : :  186
  :ERR03 : :  428
  :ERR04 : :  187
  :ERR04 :    592
   IF         415
  :OTHER : :  175
  :UNTIL : :  173
  :WHILE : :  157
  XZEVL       490
  XZSCH       560
  :-     : :  494
  :+     : :  492
  :0     : :  493

0 FIN
```

6.0 DETAILED LOGIC FLOW LISTING - PROGRAM EXECUTION

The initial pages and tailsheet of the program execution that produced this volume is presented.

```
@E*B.029,LC

029      2.0-10/05/78-08:39 AM
    1:@SYM PRINT$,,UE0083
    2:@ASG,A FMM-I91897*FDSIPDL.
    3:@USE F.,FDSIPDL.
    4:@PACK F.
    5:@FRT,TL F.
    6:@XQT F.PDLIST
    7:@ADD F.COMMON
    8:@XQT F.PDLIST
    9:@ADD F.DIRECT
   10:@XQT F.PDLIST
   11:@ADD F.MESAGE
   12:@XQT F.PDLIST
   13:@ADD F.XUPDL
   14:@XQT F.PDLIST
   15:@ADD F.XA
   16:@XQT F.PDLIST
   17:@ADD F.XC
   18:@XQT F.PDLIST
   19:@ADD F.XD
   20:@XQT F.PDLIST
   21:@ADD F.XE
   22:@XQT F.PDLIST
   23:@ADD F.XI
   24:@XQT F.PDLIST
   25:@ADD F.XL
   26:@XQT F.PDLIST
   27:@ADD F.XM
   28:@XQT F.PDLIST
   29:@ADD F.XP
   30:@XQT F.PDLIST
   31:@ADD F.XR
   32:@XQT F.PDLIST
   33:@ADD F.XS
   34:@XQT F.PDLIST
   35:@ADD F.XT
   36:@XQT F.PDLIST
   37:@ADD F.XU
   38:@XQT F.PDLIST
   39:@ADD F.XW
   40:@XQT F.PDLIST
   41:@ADD F.XX
   42:@XQT F.PDLIST
   43:@ADD F.XZ
   44:@XQT F.PDLIST
   45:@ADD F.XZ1

END 029 - 45 IMAGES PROCESSED.

@SYM PRINT$,,UE0083

@ASG,A FMM-I91897*FDSIPDL.

@USE F.,FDSIPDL.
```

```
@PACK F.
FURPUR 27R2  RL72-8 10/05/78 08:39:23
END PACK. TEXT=144,TOC=1,SYM=24,REL=3,ABS=1

@PRT,TL F.

FMM-I91897*FDSIPDL(1)   ELEMENT TABLE

D  NAME          VERSION        TYPE          DATE          TIME      SEQ #  SIZE-PRE,TEXT  (CYCLE  WORD)  PSRMODE  LOCATION
   SORT2                       ELT SYMB    15 APR 77   12:55:40        1            53     5    0    1              1792
   FSTWRD                      ELT SYMB    15 APR 77   12:56:07        2            30     5    0    1              1845
   SORT2                       RELOCATABLE 15 APR 77   12:59:32        3     2      55                              1875
   XUPDL                       ELT SYMB    22 MAR 77   05:37:22        4            84     5    0    1              1932
   PDLIST                      ELT SYMB    08 AUG 77   09:10:28        5            87     5    2    3              2016
   MAP                         ELT SYMB    08 AUG 77   09:10:34        6             2     5    1    2              2103
   PDLIST                      ABSOLUTE    08 AUG 77   09:10:43        7           280                     SET      2105
   XC                          ELT SYMB    08 NOV 77   05:47:05        8           109     5    6    5              2385
   XS                          ELT SYMB    10 NOV 77   06:19:22        9           394     5    2    3              2494
   MESAGE                      ELT SYMB    11 JAN 78   23:22:28       10            55     5   12    5              2888
   XX                          ELT SYMB    18 JAN 78   00:40:28       11           373     5    8    5              2923
   XL                          ELT SYMB    15 FEB 78   19:17:22       12           824     5    9    5              3296
   XI                          ELT SYMB    18 FEB 78   12:48:36       13           627     5   14    5              4120
   XV                          ELT SYMB    22 FEB 78   23:50:36       14           120     5    5    5              4747
   COMMON                      ELT SYMB    03 MAR 78   22:20:04       15           200     5   20    5              4867
   PDLIST                      RELOCATABLE 11 MAR 78   03:16:51       16     2      19                              5067
   FSTWRD                      RELOCATABLE 11 MAR 78   03:17:56       17     1       5                              5088
   XA                          ELT SYMB    17 APR 78   10:54:29       18           101     5   10    5              5094
   XP                          ELT SYMB    17 APR 78   10:54:52       19           534     5   22    5              5195
   XT                          ELT SYMB    27 APR 78   11:45:48       20           384     5    6    5              5729
   XZ                          ELT SYMB    18 MAY 78   09:12:58       21          1548     5    4    5              6113
   XE                          ELT SYMB    28 AUG 78   12:52:31       22           403     5   20    5              7661
   XR                          ELT SYMB    28 AUG 78   12:52:39       23           508     5   20    5              8064
   XU                          ELT SYMB    28 AUG 78   12:52:42       24           151     5    5    5              8572
   DIRECT                      ELT SYMB    25 SEP 78   09:46:47       25           118     5   13    5              8723
   XD                          ELT SYMB    25 SEP 78   09:47:13       26          1201     5   21    5              8841
   XM                          ELT SYMB    25 SEP 78   09:47:39       27           600     5   29    5             10042
   XZ1                         ELT SYMB    25 SEP 78   09:47:57       28           326     5    2    3             10642
NEXT AVAILABLE LOCATION-                                                                                          10968

ASSEMBLER PROCEDURE TABLE EMPTY

COBOL PROCEDURE TABLE EMPTY

FORTRAN PROCEDURE TABLE EMPTY

ENTRY POINT TABLE EMPTY

@XQT F.PDLIST
```

```
RUNID: RT1000   ACCT: FM6            PROJECT: FMM-I91897
TIME:    TOTAL: 00:06:33.265      CBSUPS:  044692022
         CPU:   00:04:46.816      I/O:  00:00:33.115
         CC/ER: 00:01:13.334      WAIT: 00:00:00.050
SUAS USED:  $    26.63   SUAS REMAINING:  $     0.00
IMAGES  READ:   15995     PAGES:  459
START:    08:39:18 OCT 05,1978    FIN:  09:09:21 OCT 05,1978
```


☆ U.S. GOVERNMENT PRINTING OFFICE: 1979—671-099/28 16