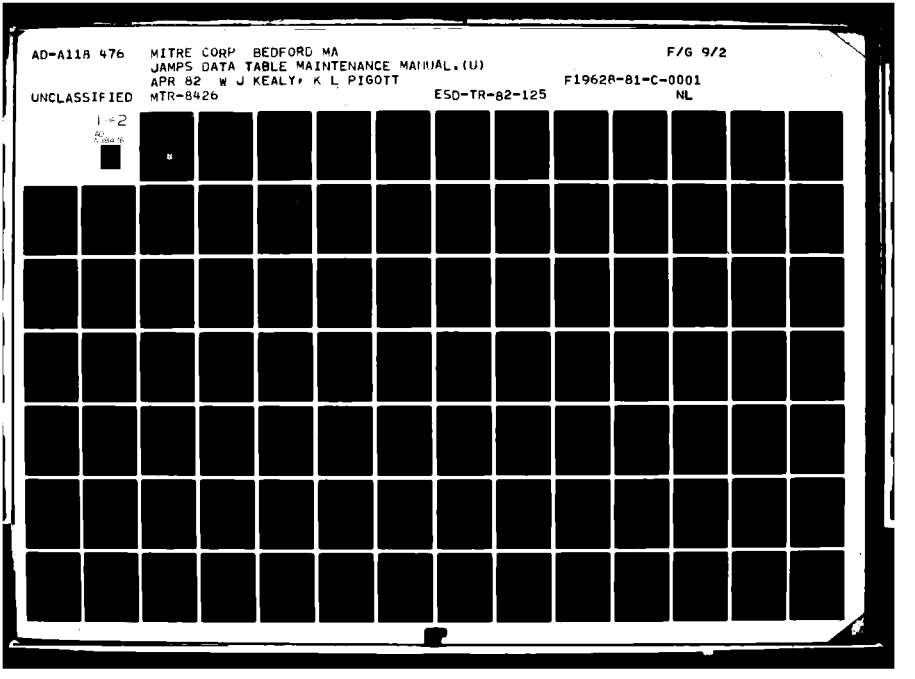

AD-A118 476
MITRE CORP BEDFORD MA
JAMPS DATA TABLE MAINTENANCE MANUAL.(U)
APR 82 W J KEALY, K L PIGOTT
F/G 9/2
F19628-81-C-0001
UNCLASSIFIED
MTR-8426
ESD-TR-82-125
NL
1 = 2

ESD-TR-82-125

MTR-8426

AD A118476

JAMPS
DATA TABLE MAINTENANCE MANUAL

APRIL 1982

Prepared for

DEPUTY FOR TACTICAL SYSTEMS
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Massachusetts

Project No. 4100
Prepared by

THE MITRE CORPORATION
Bedford, Massachusetts
Contract No. F19628-81-C-0001

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

Gregory T. Beers

GREGORY T. BEERS, Capt, USAF
ESD JINTACCS Program Manager

FOR THE COMMANDER

Theodore G. Moeller

THEODORE G. MOELLER, Col, USAF
Program Director, TABMS
Deputy for Tactical Systems

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-82-125	2. GOVT ACCESSION NO. AD-A118 476	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* JAMPS DATA TABLE MAINTENANCE MANUAL		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER MTR-8426
7. AUTHOR(s) W. J. Kealy, K. L. Pigott		8. CONTRACT OR GRANT NUMBER(s) F19628-81-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS The MITRE Corporation Burlington Road Bedford, MA 01730		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 4100
11. CONTROLLING OFFICE NAME AND ADDRESS Deputy for Tactical Systems Electronic Systems Division, AFSC Hanscom Air Force Base, MA 01731		12. REPORT DATE April 1982 13. NUMBER OF PAGES 142
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Data base
JAMPS
JINTACCS
Maintenance

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The JINTACCS (Joint Interoperability of Tactical Command and Control Systems) Automated Message Preparation System (JAMPS) uses extensive data tables to store the authorized JINTACCS message structures and data elements. This document describes the various data tables and provides instructions on how to add, change, and delete information as the JINTACCS messages change.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

FOREWORD

This document describes the various data bases required by the JAMPS program to aid in the preparation of JINTACCS messages.

Section 1 provides a brief overview of the structure of JINTACCS messages. The source files, which include the message files, help text, JANAP data and state tables, are the subject of section 2. The procedure to be followed to revise the data base is presented in section 3, and the method of rebuilding the data base is contained in section 4.

A description of the various checkout programs is given in section 5 and, finally, section 6 gives the source information for the data base contents.

A tutorial on the inclusion of a new message to the data base is given in appendix A.

ACKNOWLEDGMENTS

This report has been prepared by The MITRE Corporation under Project No. 4100. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Concluded)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Concluded)

LIST OF TABLES

SECTION 1

INTRODUCTION

This document describes the structure of the data base used by the JAMPS program to aid in the preparation of JINTACCS messages. In addition, the sources of data and the procedures to be followed to create, modify and maintain this data base are provided.

The understanding and use of the material and procedures provided in this document requires no extensive knowledge of computers, electronics or programming techniques. A familiarity with JINTACCS is assumed, and the ability to use a text editor, particularly a full-screen editor, is necessary in order to make revisions to the data base.

The purpose of JINTACCS is to increase the operational effectiveness of the services' tactical command and control operational systems and facilities used in support of joint ground and amphibious military operations through the 1980's. As a part of this objective, JINTACCS is developing standard messages and a common message language to facilitate the exchange and use of information among the JINTACCS affected systems and facilities. This exchange of information via a common language is not limited to facilities within a specific service organization, but was developed primarily for the exchange between the different services and allies such as the NATO forces. Each JINTACCS message type is structured to provide a means of modularizing the multitude of information types and values which are permissible. In this way, information required by

many different message types shall be treated identically in all uses. For example, all messages containing a field in which to specify the time of an activity (i.e., C0143 008) would use a common definition. This field's definition would provide the order of the components (day, hour, minute and time zone), the size or length of each subfield (i.e., two numeric characters for the day, hour and minute), and the set of values that is permissible for each subfield (i.e., day is a numerical value between 1 and 31). In this manner, the interpretation of the contents of all fields is standardized.

The overall structure of the JINTACCS message is a hierarchical system, with message types occupying the uppermost level. For convenience only, the set of JINTACCS messages has been divided into five classes; Air Operations, Operations Control, Intelligence, Fire Support and Amphibious. At the message level, the data content is established by the assignment of Keyword Data Sets (KDS's) to a message. A keyword data set is a logical collection of related items referred to as fields. For example, MSGID is the identity of the KDS that is required in every message. This KDS is composed of six fields which are Message Type, Originator, Serial Number, Month, Qualifier, and Serial Number of Qualifier. The hierarchical structure of JINTACCS is shown in figure 1. In this figure, the top, or uppermost level, shows the message itself, and the next lower level is the set of keyword data sets which completely define the message. The level below the keyword data sets are the fields which completely define the KDS's. In the same way that keyword data sets may be used in several different messages, a particular field in one KDS can also be a field in several different KDS's.

There are three distinct classes of KDS's; linear sets, columnar sets and free text sets. A free text set is an unformatted text

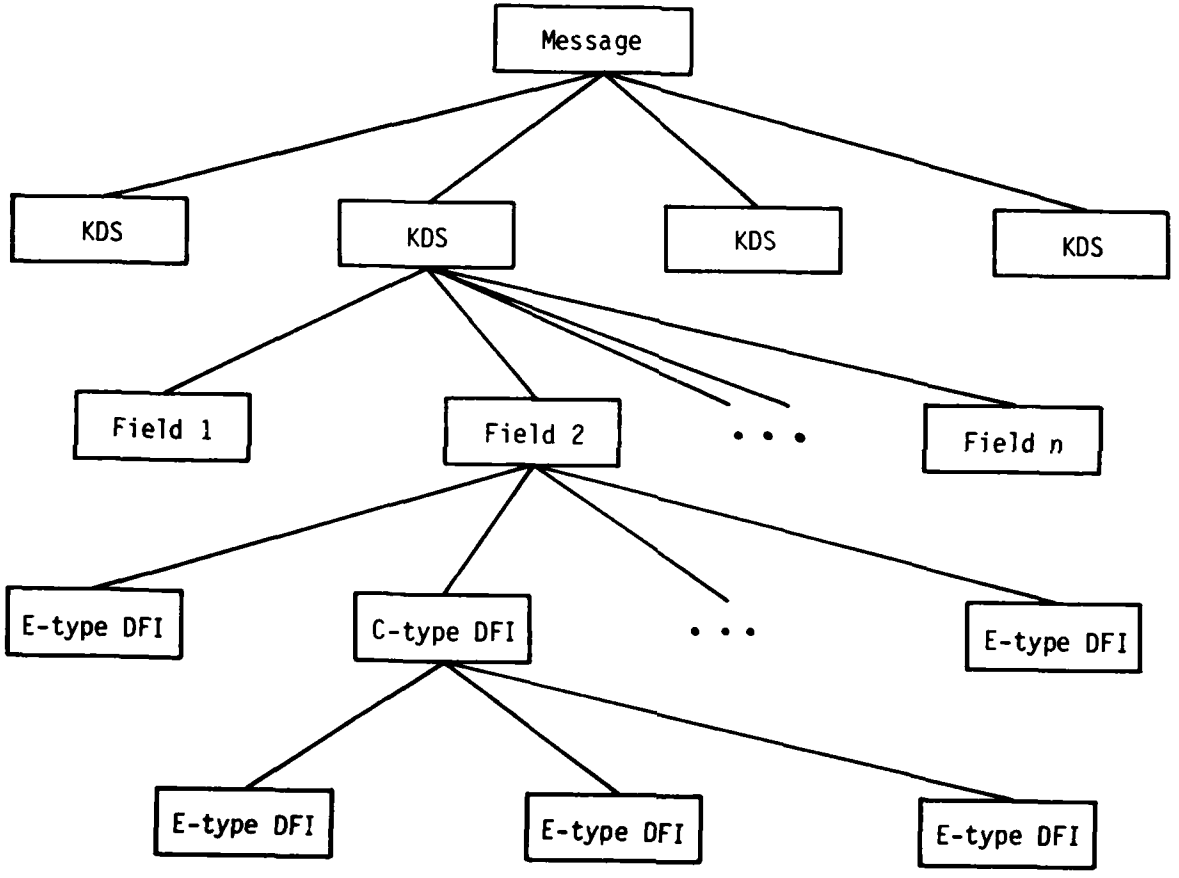


Figure 1. JINTACCS Structure Overview

statement of variable length. JINTACCS has three such KDS's; RMKS, AMPN, and NARR. All other KDS's are either linear or columnar types. These two types can be distinguished from each other by the naming convention used. All columnar KDS names begin with a number (i.e., 6BASIC), while linear KDS names are all alphabetic (such as MSGID). In messages, columnar KDS's allow data to appear in a message in tabular or columnar fashion. In addition, a heading is associated with each columnar set which provides a name or an identification for each field or column within the KDS. An example of this data structure is shown in figure 2. In this figure, the area printed in boldface is KDS CANX, which consists of four fields of data or information. In this example, the first field has the field descriptor MSG-TITLE. The outlined area in this figure is an example of a columnar KDS. The first line of this example is the associated heading for the KDS. Under each entry of the heading can be seen the entered data items. Several lines of data appear under this common heading, which is allowed under JINTACCS rules. KDS's and the fields which make up a KDS can be designated as repeatable, which allows several sets of data to appear under a common heading, as shown in the example. In addition to the repeat option, a KDS, or a field within a KDS, can be designated either optional or mandatory.

At the level below KDS fields are the two types of data elements: elemental or chain. An elemental data item consists of a single, standalone piece of information. In cases where the elemental item can be divided into two or more logically related parts, a chain element can be defined. For example, if, at the element level, the item "TIME" is defined, it can be further subdivided into two subitems: hours and minutes. The element "TIME" could then be a chained element and would be so identified by a "C" prefixed to its identity, as opposed to an "E" prefix for an elemental-type item.

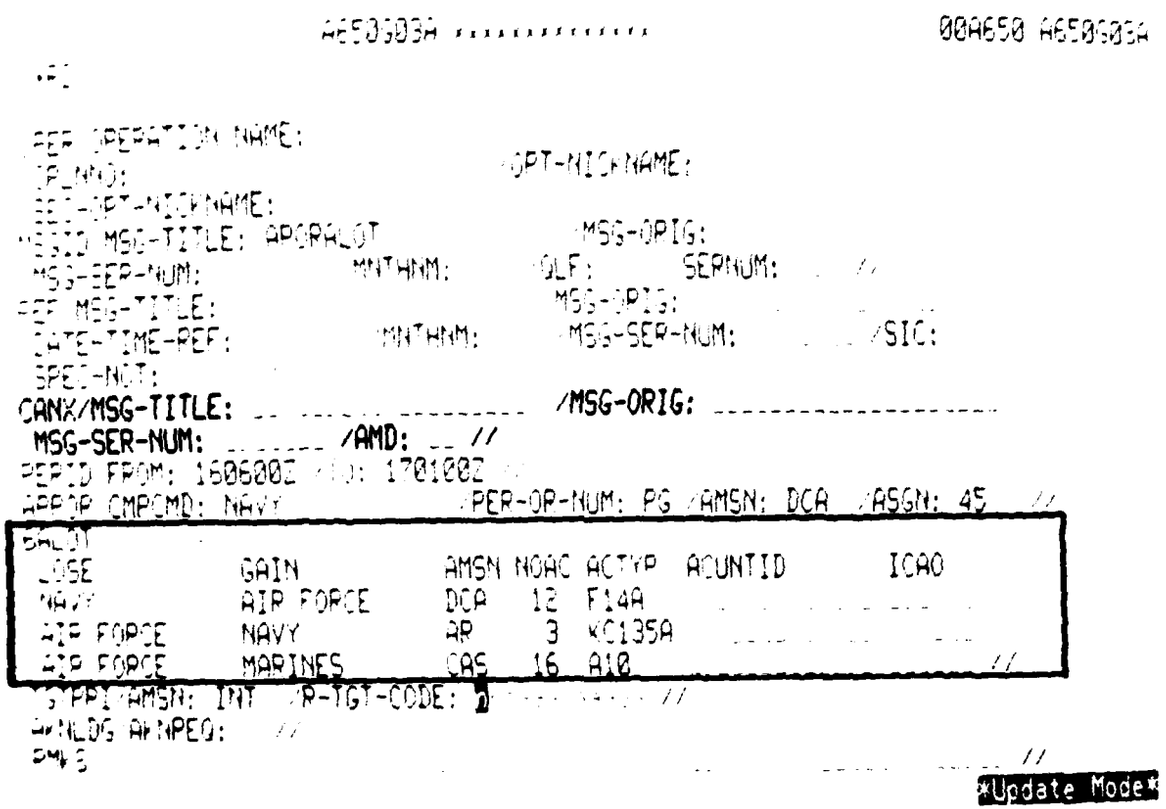


Figure 2. Typical JAMPS Display

This chain would then be defined as consisting of two E-type elements: one for hours and the other for minutes.

At the elemental level, items are referred to as Data Field Identifiers (DFI) and Data Use Identifiers (DUI). Each identifier has an identifying numerical value. A proper name of an item at the lowest level of this structure consists of the prefix letter, followed by a four-digit number, the DFI, and ending with a three-digit number, the DUI. If two or more DFI's are logically related, such as quantity destroyed (DFI/DUI E0031 003), and quantity damaged (DFI/DUI E0031 004), they will have the same DFI number, but unique DUI's, as shown by the examples.

SECTION 2

SOURCE FILES

In order to aid operators in creating JINTACCS messages, the JAMPS program makes use of several files. These files are structured in a hierarchical system similar to the JINTACCS structure. This section will discuss the format and contents of these files.

At the topmost level are the message files which provide a key to the contents and format of each message. The message index provides a linkage to the messages to be used during the final data base cross check. The message file, produced by the catmsg program, provides the descriptive information and format of each message. Related directly to the message files at the next lower level are the two KDS files: the KDS index file and KDS data file. The KDS index file links references in the message files to the proper location in the KDS data files. The KDS data file supplies the contents and format of each KDS.

The lowest level contains the elemental data in three files; the DFI/DUI file, the field descriptor/field header file and the codes file. The DFI number itself is used as an index for the KDS file to locate data at the elemental level. The DFI/DUI file defines the field size and content type such as numeric or alphabetic, while the field descriptor/field header file contains their descriptions. The codes file provides all valid character codes for each data element.

2.1 Message Files

There are three types of general source files used to contain the data required for building and validating the messages. These files contain information about all the messages in general, and define the elementary parts which comprise the messages. Two of these types of general source files are msgindex, and the catmsg file.

In addition to these two types, there are individual files for each JINTACCS message in the JAMPS system. Each file specifically defines the given message, its format and its component parts. A list of the messages implemented for the Air Force Participating Test Unit (PTU) Certification test appears in table 1. This list contains only the Air Ops messages and will be added to in the future.

2.1.1 Message Index (msgindex) File

The msgindex file, a portion of which is pictured in figure 3, is a list of all the messages and their respective index numbers. The very first entry to the msgindex file is the token ".msgindex", which serves to identify the file as being the message index file. A token is a signal to the processing routine to begin or alter processing, depending on the value of the token. In the JAMPS data base, tokens are generally designated by a leading period. The format of the file is the message number, followed by the short message name (or message acronym) and the index number which has been assigned to that message. The order of the messages in the list is not important. A new message may be added to the end of the list with the next sequential index number being assigned to that

Table 1

Message Files

Message Number	Message File Name
B704	abchange
B705	acsamstat
E710	airdefcom
E715	airdefwarn
B703	airevent
F541	aknldg
F631	almsnscd
A770	alord
D630	alreq
A650	aporalot
F654	crossconf
A653	crossdat
F750	desigarea
F751	ecmdat
A651	employaloc
B711	endsts
F632	fltcontinfo
F754	fltpln
F636	heloalconf
A635	heloaldat
E706	inithand
D660	jairreq
D665	jairsupreq
D200	jcasintreq
D666	jescreq
B750	jinflt
B702	jlnchrep
D667	jrecreq
D669	jsarreq
D668	jsupreq
C001	msgchangerep
E707	rechand
F625	reqconf
A661	reqstatask
C482	sarir
C420	sarsit
A652	sortalot
A690	tacopdat
A691	techopdat
F752	trkman
F753	trkrep

```
.msgindex

; field order is as follows:

;       message number
;       short message name
;       message index number
;       edition number (blank will be interpreted as edition 0)


d669,jsarreq,          1
a650,aporalot,         2
a653,crossdat,         3
a661,reqstatask,       4
a690,tacopdat,         5
a691,techopdat,        6
a770,alord,            7
b702,jlnchrep,         8
b703,airevent,         9
b704,abchange,        10
          .
          .
          .
f636,heloalconf,      40
d665,jairsupreq,      41
f755,trkintel,        42
c460,comspot,         43
```

Figure 3. Portion of msgindex File

message. Classified messages, however, should be grouped together, preferably at the end of the file so they can easily be extracted if an unclassified system is desired.

2.1.2 Concatenate Messages (catmsg) File

The concatenate messages file, **catmsg**, is a shell program which uses the UNIX[TM] system "cat" command to concatenate the message description files into one large file. This file is required in the building of a data base (see section 4).

As messages are added to or removed from the data base, it is necessary to appropriately update the shell program as it is the contents of this file which define the final contents of the data base. Figure 4 shows the **catmsg** shell program with its list of short message names (the names of the message files). Each entry in this file is terminated by a backslash (\) character.

Due to the number of arguments that the "cat" command will take, with the maximum number being approximately 40, **catmsg** first concatenates half of the messages with the result written to **msg1**, concatenates the second half with the result written to **msg2**, and then concatenates **msg1** and **msg2**, with the result written to **msgs**, and removes **msg1** and **msg2**. If a large number of files are added to the existing data base, more concatenations may have to be performed so that the number of arguments for each evocation of the "cat" command is less than 40. We are left with a single file, **msgs**, of all the messages.

[TM]: UNIX and PWB/UNIX are trademarks of Bell Telephone Laboratories, Inc.

```
 cat\
jsarreq\
aporalot\
crossdat\
reqstatask\
tacopdat\
techopdat\
alord\
jlnchrep\
airevent\
abchange\
acsamstat\
engsts\
jinflt\
msgchangerep\
sarsit\
sarir\
jcasintreq\
jairreq\
alr-aj\
> msgl
 cat\
jrecreq\
jescreq\
jsupreq\
inithand\
reqconf\
rechand\
airdefcon\
fltcontinfo\
airdefwarn\
aknldg\
almsnsoi\
crossconf\
trkrep\
fltpln\
desigarea\
ecmlat\
employaloc\
sortalot\
trkman\
heloaldat\
heloalconf\
jairsupreq\
comspot\
trkintel\
> msg2
cat msgl msg2 >msgs
rm msgl msg2
```

Figure 4. Portion of **catmsg** File

2.1.3 Message Description Files

The JINTACCS messages supported by the JAMPS program are in files which contain all of the necessary descriptive and processing information. Each of the 43 files of message information for the Air Ops messages which were installed for the Air Force PTU Certification are accessed according to the short message name of the message. It is these message files that are concatenated by **catmsg** to a single file (**msgs**) (see paragraph 2.3). Each message file consists of three parts, with an optional fourth part. An example of a message description file is shown in figure 5. This figure shows the four basic parts of the file to be described in the following paragraphs.

2.1.3.1 Identification. The first part of each file identifies the message by message number and edition number. The token ".msg" is used to identify the message number. This token is required in every message file, followed by a blank character, and the message number. In figure 5, the file **aporalot** is identified by its message number, A650. Messages will be retrieved from the JAMPS data base by their message number.

Following this entry is the edition number. Although this is not presently used by the JAMPS program, it must be entered as a zero value for proper operation. A future version of the JAMPS program may allow for two or more versions of the same message type. This will be permitted by the use of the edition number.

2.1.3.2 Keyword Data Set Table. This part of the message description file is a table of all the keyword data sets which comprise the message. The KDS presentation number is a sequential number

```
.msg a650
.edition 0

; field order is as follows:

;       presentation number of keyword data set
;       keyword data set name
;       state number
;       mandatory indicator (m or M, if mandatory)
;       repeatability indicator(r or R, if repeatable)
;       version number (optional, blank will use most recent version)
;       delete indicator (1=delete on input,2=delete on output,3=delete on
;         input and output, blank or 0=used for both input and output)

;  P,       KDS, ST, M, R, V, D          ;
;   ,          ,  ,  ,  ,  ,            ;

   1,      from,  1, m,  ,  ,            ;
   2,        to,  2, m, r,  ,            ;
   3,      info,  3,  , r,  ,            ;
   4,     class,  4, m,  ,  ,            ;
   5,       sic,  5,  ,  ,  ,            ;
   6,      exer,  6,  ,  ,  ,            ;
   7,      oper,  7,  ,  ,  ,            ;
   8,     msgid,  8, m,  ,  ,            ;
   9,       ref,  9,  , r,  ,            ; made repeatible by rev. 1, TIDP
  10,      canx, 10,  ,  ,  ,            ;
  11,     perid, 11, m,  ,  ,            ;
  12,     appor, 12, m, r,  ,            ;
  13,     6alot, 13,  ,  ,  ,            ;
  14,    tqtpri, 14,  , r,  ,            ;
  15,    aknldg, 15,  ,  ,  ,            ;
  16,      rmks, 16, m,  ,  ,            ;
  17,  dwngrade, 17,  ,  ,  ,            ;

.state

0, 1                            ;from mandatory
1, 2                            ;to   mandatory
2, 2, 3, 4                      ;to is also repeatable
3, 3, 4                         ;info is repeatable
4, 5, 6, 7, 8                   ;class mandatory
5, 6, 7, 8
6, 8                            ;choose either exer or oper
7, 8,
8, 9,10,11                      ;msgid is mandatory
9,9,10,11
10,11                           ;perid mandatory
11,12                           ;apor is mandatory
12,12,13,14,15,16               ;apor is repeatable
13,14,15,16
14,14,15,16                     ;tqtpri is repeatable
15,16
16,17,#                         ;rmks mandatory
17, #                           ;dwngrade optional as result of PTR

;  following area is optional and used for mandatory entries in
;    certain fields of keyword data sets

;  field order is as follows
;    presentation number for applicable keyword data set
;    relative field number to which entry applies
;    entry to be inserted (will be used as a character string)
;    protection code for data (p protected, u or blank unprotected)

  .man
  4,1,s e c r e t,u   ;class
  8,1,aporalot,p ;msgid
```

Figure 5. Message Description File

assigned to each different KDS, and is shown in the first column. This number is the KDS appearance order in the MED. Another number, the state number, is assigned to each KDS, as seen in the third column. In most cases, the presentation number is identical to the state number. The KDS name appears in the second column. A mandatory indicator ("m" or "M" if mandatory, blank otherwise), is in column four and the next column contains a repeatability indicator ("r" or "R" if repeatable, blank otherwise). The version number (optional, blank will use the most recent version) is in column six and the last column contains the delete indicator. The delete indicator is presently not implemented; it is reserved for future use.

2.1.3.3 State Tables. The third section of the message description file is the state table, which shows the interdependencies and interrelationships of the keyword data sets. This table is created from the data in the preceding table, mainly the mandatory and repeatable indicators, and the information found in the TIDP, specifically the group repeatability indicators. The state table is used solely for the message validation process. The creation of state tables is explained in greater detail in appendix A.

2.1.4 Prepopulation

Prepopulation data is data added to a message by the JAMPS system as a convenience to the operator when the field entry is invariant. The optional fourth part of the message description file is used for the prepopulation (i.e., initial or default values) of mandatory entries in certain KDS fields. The format of this part of the file is the presentation number of the KDS, the relative field number to which the entry applies, the entry to be inserted (used as

a character string), and a protection code ("p" if write-protected, and "u", or blank, if unprotected). If prepopulation data is write-protected, it cannot be changed by a JAMPS terminal operator.

2.2 Keyword Data Set Files

From the messages that make up the data base, a unique set of KDS's is extracted. This data and an index of the data comprise the KDS file. Following is a description of the KDS index and the KDS file.

2.2.1 Keyword Data Set Index (kdsindex) File

The **kdsindex** file, a portion of which is shown in figure 6, is a list of all the keyword data sets used in the messages. Similar to the message index file, the KDS index must have the ".kdsindex" token as the initial entry. The format of the file is the KDS name followed by its index number and version number. A blank entered for the version number will be interpreted by the system as a zero. The **kdsindex** file is arranged in alphabetical order, with the exception of the JANAP and MSGCHANGEREP keyword data sets. The JANAP keyword data sets are used to provide the information for framing the JINTACCS message for trasmittal according to JANAP 128(H) standards. This order was used for ease of maintainability, but there are no system requirements that this must be observed. Classified keyword data sets should be appended to the end of the file so that they can be easily extracted if it is necessary to declassify the file.

```
.kdsindex

;  field order is as follows:

;       keyword data set name
;       keyword data set number
;       version number ( blank is interpreted as zero (0))

FROM,            1
TO,              2
INFO,            3
AMPN,            4
NARR,            5
RMKS,            6
CLASS,           7
CHANGE,        199
CANCEL,          8
DELETE,          9
AFTER,          10
ADD,            11
6ALOT,          13
6APPDIS,        14
6BASIC,         15
          .
          .
          .
AAR,            58
AAW,            59
ABORT,          60
ACEMER,         61
ACLOSS,         62
ACLOST,         63
          .
          .
          .
TGTPRI,        188
TIMEAMP,       189
UNABLE,        190
UNITLOC,       191
ZZREF,         192
OTHERSAR,      193
COMEV,         194
INTELTK,       195
LOCN,          196
RELOC,         197
TKAMP,         198
```

Figure 6. Portion of **kdsindex** File

2.2.2 Keyword Data Set (kdss) File

The kdss file is a file of all the keyword data sets and their component parts. A section of the kdss file is shown in figure 7. The format of this file is similar to the message description files, however, all the keyword data sets are in one file, and there are only three parts to be completed for each keyword data set.

2.2.2.1. Identification. The first part of the kdss file is the identification and contains the KDS name, the KDS type (i.e., linear, columnar, free text or JANAP), and the version number. The KDS type identifier, JANAP, has been aded to the JINTACCS-defined types of linear, columnar and free text KDS's to aid in the operation of the JAMPS system. A JANAP-type KDS is a keyword data set which contains information used in the JANAP framing. If the KDS is of the free text type, the identification section is the only entry made to kdss.

2.2.2.2 Tables. The second part is a table containing the field number, the DFI number, the DUI number, the state number, the print position for the field header (only for a columnar-type KDS), a mandatory indicator ("m" or "M" if mandatory), a repeatability indicator ("r" or "R" if repeatable), a field descriptor indicator ("1" if required, "0", or blank, if not required), and a delete indicator. The delete indicator is presently unimplemented but is reserved for future use.

2.2.2.3 Keyword Data Set State Tables. The third part is the state table which defines the interdependencies and interrelationships that can not be properly conveyed in the preceding table. The state table is used solely for validation purposes.

```
.KDS FROM
.JANAP
.VERSION 0

.FIELDS
. FIELD ORDER IS AS FOLLOWS:

;       FIELD NUMBER
;       DFI NUMBER
;       DUI NUMBER
;       STATE NUMBER
;       PRINT POSITION FOR FIELD HEADER (ONLY USED FOR TYPE COLUMNAR)
;       MANDATORY INDICATOR (M OR M, IF MANDATORY)
;       REPEATABILITY INDICATOR(R OR R, IF REPEATABLE)
;       FIELD DESCRIPTOR INDICATOR (1=REQUIRED, 0 OR BLANK=NOT USED)
;       DELETE INDICATOR (1=DELETE ON INPUT,2=DELETE ON OUTPUT,3=DELETE ON
;         INPUT AND OUTPUT, BLANK OR 0=USED FOR BOTH INPUT AND OUTPUT)

. P,   DFI, DUI, ST, PR, M, R, F, D     ,
.  ,     ,   ,   ,  ,  ,  ,  ,  ,      ;
  1, E1400, 001,  1,   ,  ,  ,  ,      ;PLA
  2, E1401, 001,  2,   ,  ,  ,  ,      ;ROUTING INDICATOR
  3, E1402, 001,  3,   ,  ,  ,  ,      ;ADDRESSEE DESIGNATOR
  4, E1403, 001,  4,   , M,  ,  ,      ;CLASSIFICATION
  5, E1404, 001,  5,   ,  ,  ,  ,      ;PRIMARY PRECEDENCE
  6, E1405, 001,  6,   ,  ,  ,  ,      ;SECONDARY PRECEDENCE
  7, E1406, 001,  7,   , M,  ,  ,      ;DATE/TIME GROUP

.STATE

0, 1, 2                         ;MUST HAVE EITHER PLA OR RI AND ADDR
1, 2, 4                         ;CLASSIFICATION IS MANDATORY
2, 3                            ;IF RI THEN MUST INCLUDE ADDR
3, 4
4, 5, 6, 7
5, 6, 7
6, 7
7, #                            ;DATE TIME GROUP IS MANDATORY

.KDS BALOT
.COLUMNAR
.VERSION 1

.FIELDS
  1, F0987, 031,  1,  2, M,  ,  ,      ;
  2, F0987, 032,  2, 16, M,  ,  ,      ;
  3, F0107, 004,  3, 30, M,  ,  ,      ;
  4, F0031, 022,  4, 35, M,  ,  ,      ;
  5, F0513, 001,  5, 40,  ,  ,  ,      ;
  6, F0987, 023,  6, 47,  ,  ,  ,      ;
  7, F0527, 001,  7, 61,  ,  ,  ,      ;


.STATE

0,  1,
1,  2,
2,  3,
3,  4,
4,  5,  6,  7,  #,
5,  6,  7,  #,
6,  7,  #,
7,  #,

.KDS  AAR
.LINEAR
.VERSION 0

.FIELDS
  1, E0500, 024,  1,   , M,  ,  ,      ;
  2, E0491, 002,  2,   , M,  , 1,      ;
  3, E0651, 002,  3,   , M,  ,  ,      ;
  3, E0681, 002,  4,   , M,  ,  ,      ;
  4, E0651, 003,  5,   ,  ,  ,  ,      ;
  4, E0681, 003,  6,   ,  ,  ,  ,      ;
  5, C0469, 008,  7,   , M, R,  ,      ;
  5, C0497, 005,  8,   , M, R,  ,      ;
  5, C0498, 002,  9,   , M, R,  ,      ;


.STATE

0,  1,
1,  2,
2,  3,  4,
3,  5,  6,  7,  8,  9,
4,  5,  6,  7,  8,  9,
5,  7,  8,  9,
6,  7,  8,  9,
7,  7,  8,  9,  #,
8,  7,  8,  9,  #,
9,  7,  8,  9,  #,

.KDS RMKS
.FREETEXT
.VERSION 0
```

Figure 7. Portion of kdss File

Each KDS is entered in the **kdss** file in a relatively alphabetical order. Again, this order need not be preserved for system operability, but may be desired for ease of maintenance. Like the **kdsindex** file, classified keyword data sets should be grouped at the end of the file.

2.3 Elemental Files

The elemental files represent the lowest level of the JAMPS structure. As such, these files not only form part of the structure, like the message and KDS files, but also, in certain instances, define the contents. Three of these files will be described at the elemental level: the DFI/DUI file, the Field Descriptor/Field Header file, and the Codes file.

2.3.1 DFI/DUI (dfidui) File

The **dfidui** file is a unique set of all the DFI/DUI's used in support of the JAMPS system. A section of the **dfidui** file is shown in figure 8. Each DFI/DUI is handled according to its DUI indicator: "com", "dif", or "chn". All data elements which are chained; that is, if the DFI is a "C"-type DFI, use the DUI indicator "chn" (chain). All other DFI/DUI's fall into the remaining two categories which can be determined by examining all DUI's which appear for a given DFI. The "com" (common) indicator is used if all DUI's for a given DFI have the same minimum and maximum code lengths, message map characters and left or right justification flags within the field. If one or more of the DUI's differ from any other with respect to any of these characteristics, the "dif" (different) indicator is used as the DUI indicator. Originally a fourth type, "altchn" (alternate chain), was possible. However, JINTACCS removed

```
.com e0001,1,v,1,6,x

.dif e0002
  .dui 001,1,v,1,17,x
  .dui 007,1,v,1,17,x
  .dui 010,1,v,1,17,x
  .dui 015,1,v,1,17,x
  .dui 018,1,v,1,17,x
  .dui 019,1,v,1,17,x
  .dui 020,1,v,1,12,x
  .dui 021,1,v,1,12,x
  .dui 022,1,v,1,12,x
  .dui 023,1,v,1,17,x
  .dui 024,1,v,1,17,x
  .dui 025,1,v,1,17,x
  .dui 027,1,v,1,17,x
  .dui 028,1,v,1,12,x
  .dui 031,1,v,1,12,x

.dif e0003
  .dui 001,r,v,2,4,n
  .dui 002,1,v,1,6,x

.dif e0007
  .dui 001,1,f,,1,a

.com e0008,1,f,,6,aannnn

.com e0009,1,f,,9,x

.chn c0011,1,f,,15,nnnnnnannnnnnna
  .chain
     e0017,001
     e0156,002
     e0157,002
     e0158,002
     e0018,001
     e0156,003
     e0157,003
     e0158,003
```

Figure 8. Portion of **dfidui** File

all but one of the alternate chain-type DFI's. For ease of operation, the remaining alternate chain was replaced in the data base by two chains. This replacement does not change the operation of the program, and it is transparent to the operators.

The format of the entry for a "com"-type DUI is the DUI indicator, followed by the DFI number, a left/right justification flag ("l" for left, and "r" for right), a field length indicator ("v" for variable length, and "f" for fixed length), the minimum field length, the maximum field length, and the message map characters. If only one message map character appears, all characters in the field are of that type.

When a DUI is a "dif" type, the entry is the DUI indicator followed by the DFI number. This, in turn, is followed by each DUI number and its left/right justification flag, field length indicator, minimum field length, maximum field length, and message map characters.

The format for the entry of a "chn"-type DFI is the DUI indicator, followed by the DFI number, left/right justification flag, field length indicator, minimum field length, maximum field length, and message map characters. This, in turn, is followed by the elemental components of the chain; the DFI number and the DUI number.

2.3.2 Field Descriptor/Field Header (fdfh) File

The **fdfh** file is a list of all DFI/DUI's and their respective field descriptors and field headers. The format of the file is the DFI number, followed by the DUI number, the field descriptor and field header. Figure 9 shows sections of the **fdfh** file.

```
.fdfh

e0001, 001,msnno,msnno
e0002, 001,callsign,callsign
e0002, 007,authcs,authcs
e0002, 010,intunt,intunt
e0002, 015,cntcs
e0002, 018,from,from
e0002, 019,to,to
e0002, 020,acsign,acsign
e0002, 021,tankercs,tankercs
e0002, 022,reccs,reccs
e0002, 023,initcon,initcon
e0002, 024,fnlcon,fnlcon
e0002, 025,dclauth,dclauth
e0002, 027,mcorcs,mcorcs
e0002, 028,rendcs,rendcs
e0002, 031,atkaccs,atkaccs
e0003, 002,reqno,reqno
          .
          .
          .
e0987, 031,lose,lose
e0987, 032,gain,gain
e0987, 037,btlgrp,btlgrp
e0992, 001,frqpri,frqpri            ;missing from med
e1028, 001,typalt,typalt
e1029, 001,hzd,hzd
e1030, 001,ldtyp,ldtyp
e1031, 001,ldref
e1031, 002,onloc,onloc
e1031, 003,offloc,offloc
e1032, 001,geofil
e1040, 001,sidenr,sidenr
e1042, 001,spec-not                   ;no field descriptor in MED
e1400, 001,pla                        ;janap 128 dfis
e1401, 001,ri
e1402, 001,addr
e1403, 001,class
e1404, 001,priprec
e1405, 001,secprec
c1406, 001,dtg
```

Figure 9. Portion of **fdfh** File

2.3.3 Codes (codes) File

The codes file contains all valid code entries for all DFI/DUI's. This file is used for validation of operator entries and for the display of minimum and maximum code lengths. The codes file, as well as the itemcodes file (paragraph 2.4.2), was extracted from a System Development Corporation (SDC) print tape of the JINTACCS field coding report.

Due to the size of the codes file and to aid in its maintenance, codes has been divided into four sections, codes1, codes2, codes3 and codes4, according to the DFI/DUI number. Before processing the source files, the four sections are concatenated and written to codes. For the remainder of this manual, the four sections will not be referred to separately, but as a single entity, codes. A portion of the codes file is shown in figure 10.

The first entry to the codes file is the DFI type, i.e., "E" or "C", followed by the DFI number and DUI number, with no separating space between the two, and the DUI name. This is followed by the data items in a column on the left, and the data item codes in a corresponding column on the right. The data items and data item codes appear in the codes file in the same form as they appear in the MED, i.e., if a code or data item spans more than one line, it will span more than one line in the codes file. The codes file makes use of manually entered concatenation symbols to properly concatenate, upon processing, the codes which span more than one line. The concatenation symbols used are the sharp sign, "#", to indicate a blank should be left between the two sections of the code, and an equal sign, "=", to indicate a hyphen which should be deleted upon concatenation. The data items and codes are then

```
E 001001      MISSION NUMBER
*      (LITERAL)                * (LITERAL)      *
*      (SEE INSTRUCTIONS)       *                *
E 002001      CALL SIGN
*      (LITERAL)                * (LITERAL)      *
*      (SEE INSTRUCTIONS)       *                *
*    NONE                       * NONE           *
*    UNKNOWN                    * UNK            *
E 002007      CLEARANCE AUTHORITY CALL SIGN
*      (LITERAL)                * (LITERAL)      *
*      (SEE INSTRUCTIONS)       *                *
*    NONE                       * NONE           *
*    UNKNOWN                    * UNK            *
                    .
                    .
                    .
E 515001      POINT TYPE
                    .
                    .
                    .
*    HAZARD, IMPACT POINT       * IMPACT POINT   *
*    HAZARD, GROUND ZERO        * GROUND ZERO    *
*    HAZARD, AIR/WEAPON ENTRY   * AIR WEAPON#    *
*      POINT                    * ENTRY POINT    *
*    HAZARD, MISSILE LAUNCH     * LAUNCH POINT   *
*      POINT                    *                *
*    HAZARD, ECM DECOY          * ECM DECOY      *
*    GENERAL REFERENCE POINT    * GENERAL REF=   *
*                               * ERENCE POINT   *
*    MARSHAL POINT              * MARSHAL#       *
*                               * POINT          *
*    WAY POINT                  * WAY POINT      *
*    CORRIDOR TAB               * CORRIDOR TAB   *
*    PIM                        * PIM            *
*    DISPOSITION CENTER         * DISPOSITION#   *
*                               * CENTER         *
                    .
                    .
                    .
E 534001      MINUTE NORTHING, GEOREF
*      (00                      * (00            *
*      THRU                     *   THRU         *
*      59)                      * 59)            *
                    .
                    .
                    .
E 627001      CODE (IFF/SIF)
*      (00                      * (00            *
*      THRU                     * THRU           *
*      7777)                    * 7777)          *
*                               * (OCTAL)        *
                    .
                    .
                    .
E 674001      BEACON CODE
*    A                          * A              *
*    B                          * B              *
*    C                          * C              *
*    D                          * D              *
*    E                          * E              *
*    F                          * F              *
                    .
                    .
                    .
E1405001      JANAP SECONDARY PRECEDENCE
*    FLASH                      * Z              *
*    IMMEDIATE                  * O              *
*    PRIORITY                   * P              *
*    ROUTINE                    * R              *
C1406001      JANAP DATE/TIME GROUP
E1407001      JANAP DAY OF MONTH
E1408001      JANAP BLANK
```

Figure 10. Portion of codes File

delimited by asterisks. After processing the SDC tape, the asterisks appeared in columns 1, 32, and 48. The asterisks are not required to appear in these columns, but are used merely as a convenient delimiter in processing the file.

The DFI/DUI's appear one right after the other with no separation. The codes file appears in DFI order although, once again, this is not required for system operation.

2.4 Help Text

There are two files which provide the data used in conjunction with the "HELP" capability of the JAMPS program. A file named **helptext** contains the defining textual information, while the **itemcodes** file provides the discrete data items and data item codes. Like the codes file, the **helptext** and **itemcodes** files are stored in four sections according to DFI/DUI number, which are concatenated before processing of the source files. For the remainder of the manual, however, they will be referred to as single files, **helptext** and **itemcodes**.

The **helptext** file was the result of a combined Air Force and MITRE effort. Help text coding sheets were completed by Air Force personnel and entered into the system by System Architecture Incorporated (SAI) contractors. This method provided approximately one-third of the **helptext** file. The remaining portion of the file was extracted from the SDC print tape and manually edited into usable form.

2.4.1 Help Text (helptext) File

Figure 11 shows sections of the **helptext** file. The first entry to the **helptext** file is the token ".DFI". This is followed by the DFI type ("E" or "C"), then the DFI and DUI numbers, with this information appearing exactly as it appeared in the **codes** file, followed by the DUI name.

The next section of the file is the DFI/DUI definition, and information on how to enter data into the given field. The definition is preceded by "DEFINITION:", immediately followed by the actual information on the same line. The entry format of the definition should be in a readable form since it will be displayed exactly as entered. The "DEFINITION:" token is mandatory; likewise, a line of defining text cannot exceed 79 characters, the maximum width of the display screen. If the help text information spans more than 18 lines, the ".PAGE" token must be used to show the page separation.

The final entry to the file must be the ".DFI" token in order for all DFI's to be processed correctly.

2.4.2 Data Items and Data Item Codes (itemcodes) File

The **itemcodes** file, portions of which are shown in figure 12, is used in conjunction with the **helptext** file for display of field definition and entry data in HELP or CONVERSATION mode.

The format of the **itemcodes** file is identical to the **codes** file. The DFI type, DFI number, DUI number, and DUI name are followed by the data items and data item codes, delimited by asterisks. Like the **codes** file, the positioning of the asterisks and the width

```
.DFI
E 001001
MISSION NUMBER

DEFINITION: THE IDENTIFYING NUMBER ASSIGNED TO A MISSION.

ENTER THE MISSION NUMBER. THIS NUMBER CANNOT BE LONGER THAN 6 ALPHANUMERIC
   CHARACTERS.

.DFI
E 001002
COLLECTOR MISSION NUMBER

DEFINITION: THE IDENTIFYING NUMBER ASSIGNED TO A COLLECTIVE MISSION.

ENTER THE MISSION NUMBER. THIS NUMBER CANNOT BE LONGER THAN 6 ALPHANUMERIC
   CHARACTERS.
                                        .
                                        .
                                        .
.DFI
   001002
MISSION LOCATION, LAT/LONG

DEFINITION: THE LOCATION AT WHICH AN AIR MISSION IS TO BE PERFORMED.  THESE
   COORDINATES CAN BE A TARGET, ORBIT, COMBAT AIR PATROL POINT, STATION,
   ETC.

SEE NEXT PAGE FOR INSTRUCTIONS ON HOW TO ENTER DATA FOR THIS FIELD.
.PAGE
THIS IS AN EXAMPLE OF A COMPLETE LOCATION ENTRY:

     553030N1203030E

ENTER THE FOLLOWING SERIES OF CHARACTERS AND NUMBERS:
   1. THE LATITUDINAL DEGREES (00-90)
   2. THE LATITUDINAL MINUTES (00-59)
   3. THE LATITUDINAL SECONDS (00-59)
   4. THE LATITUDINAL HEMISPHERE
       N-NORTH HEMISPHERE
       S-SOUTH HEMISPHERE
   5. THE LONGITUDINAL DEGREES (000-180)
   6. THE LONGITUDINAL MINUTES (00-59)
   7. THE LONGITUDINAL SECONDS (00-59)
   8. THE LONGITUDINAL HEMISPHERE
       E EAST HEMISPHERE
       W WEST HEMISPHERE
```

Figure 11. Portion of **helptext** File

```
E 001001      MISSION NUMBER
*      (LITERAL)                * (LITERAL)      *
*      (SEE INSTRUCTIONS)       *                *
E 002001      CALL SIGN
*      (LITERAL)                * (LITERAL)      *
*      (SEE INSTRUCTIONS)       *                *
*    NONE                       * NONE           *
*    UNKNOWN                    * UNK            *
E 002007      CLEARANCE AUTHORITY CALL SIGN
*      (LITERAL)                * (LITERAL)      *
*      (SEE INSTRUCTIONS)       *                *
*    NONE                       * NONE           *
*    UNKNOWN                    * UNK            *
                    .
                    .
                    .
E 515001      POINT TYPE
                    .
                    .
                    .
*    HAZARD, IMPACT POINT       * IMPACT POINT   *
*    HAZARD, GROUND ZERO        * GROUND ZERO    *
*    HAZARD, AIR/WEAPON ENTRY   * AIR WEAPON     *
*      POINT                    * ENTRY POINT    *
*    HAZARD, MISSILE LAUNCH     * LAUNCH POINT   *
*      POINT                    *                *
*    HAZARD, ECM DECOY          * ECM DECOY      *
*    GENERAL REFERENCE POINT    * GENERAL REF-   *
*                               * ERENCE POINT   *
*    MARSHAL POINT              * MARSHAL        *
*                               * POINT          *
*    WAY POINT                  * WAY POINT      *
*    CORRIDOR TAB               * CORRIDOR TAB   *
*    PIM                        * PIM            *
*    DISPOSITION CENTER         * DISPOSITION    *
*                               * CENTER         *
                    .
                    .
                    .
E  `4001      MINUTE NORTHING, GEOREF
*      (00                      * (00            *
*      THRU                     *   THRU         *
*      5?)                      * 59)            *
                    .
                    .
                    .
E 627001      CODE (IFF/SIF)
*      (00                      * (00            *
*      THRU                     * THRU           *
*      7777)                    * 7777)          *
*                               * (OCTAL)        *
                    .
                    .
                    .
E 674001      BEACON CODE
*    A THRU F                   * A THRU F       *
                    .
                    .
                    .
E1405001      JANAP SECONDARY PRECEDENCE
*    FLASH                      * Z              *
*    IMMEDIATE                  * O              *
*    PRIORITY                   * P              *
*    ROUTINE                    * R              *
C1406001      JANAP DATE/TIME GROUP
E1407001      JANAP DAY OF MONTH
E1408001      JANAP BLANK
```

Figure 12. Portion of **itemcodes** File

of the columns are flexible. Entries to **itemcodes** will be displayed in the HELP mode as they appear in the source file. For this reason it may be desirable to concatenate codes that span more than one line in **itemcodes** rather than having them displayed in sections. This may be done, however the total line length cannot exceed 79 characters, the width of the display screen.

2.5 JANAP

The JANAP data consists of three files which provide the means to prepopulate certain fields of a new message. This information is used as the contents of the TO and FROM KDS's and consists mainly of address information. The JANAP files are the **Adr** file which contains addresses, the **Orig** file containing data pertaining to the message originator, and the **Rout** file, the list of addressees for each message.

2.5.1 Address (Adr) File

The first of these files is **Adr**, or the address file. This file is accessed by the Plain Language Address (PLA), which is part of each entry. A complete entry consists of three pieces of information: a PLA, a routing indicator, and an ordinary address. The PLA, which is also the keyword of the file, is a 14-character, or less, alphanumeric abbreviation of the address. The PLA is used by operators to specify an addressee for a message. The routing indicator is a 7-character, or less, alphabetic entry. The routing indicator is used by the AUTODIN Line Controller as an address for the switching (or routing) of messages through the network. The ordinary address can be up to 57 alphanumeric characters, and is the real-world address of an Operational Facility (opfac).

2.5.2 Message Originator (Orig) File

The Orig file consists of an entry for each operational facility supported by the JAMPS system. Each entry in this file is identified and retrieved by a four- or five-character name, which is also the login identity for that opfac. The remainder of the data in the entry satisfies the FROM fields of the KDS. In addition to this data, a file entry has the transmission channel number and name, and a designator. This information, although not part of JINTACCS, is required for local software use and is obtained from the Test Coordinator. The information from this file is used to prepopulate the FROM KDS of every message composed, using the login identity as the keyword for the retrieval.

2.5.3 Message Routing (Rout) File

The third file, Rout, is used to prepopulate messages with the TO and INFO addressees. This information is configured as separate files for each opfac supported by the JAMPS system. A file for a specific opfac will contain an entry for each message type that can be originated at that opfac. For each message type, the data will consist of a classification code, a primary and secondary precedence, a list of both TO and INFO PLA's.

When a message is to be composed, the JAMPS program will use the login identity and the message type entered as the "compose" argument to find the prepopulation data in the file. This information will then be copied into the message format and displayed to the operator.

SECTION 3

REVISING THE DATA BASE

3.1 Message Revision

The message data base files are designed in a hierarchical order, with the lower levels being the component parts of the upper levels. The hierarchical order of the JAMPS data base is shown in figure 13.

A change to any of the upper levels may have a cascading effect on the lower levels. For example, a change to add or delete a message may involve changing as few as three files or as many as ten.

3.1.1 Addition/Deletion of a Message

In order to add or delete a message from the JAMPS data base, changes must be made to the **msgindex** and **catmsg** files. The short message name must be entered into or deleted from **catmsg**. Each entry in this file most be followed by a backslash. The **msgindex** file is updated by adding the message number, followed by the short message name, and ended with the next sequential index number. The message number and short message name can be obtained from the TIDP. Information sources will be discussed in further detail in section 6. When a message is deleted, it is advisable to renumber the message index numbers in **msgindex**. Although the program will work properly if there is a break in these numbers, the cross check program (see paragraph 5.2), will cease operating when it encounters a break; all messages will not be cross checked.

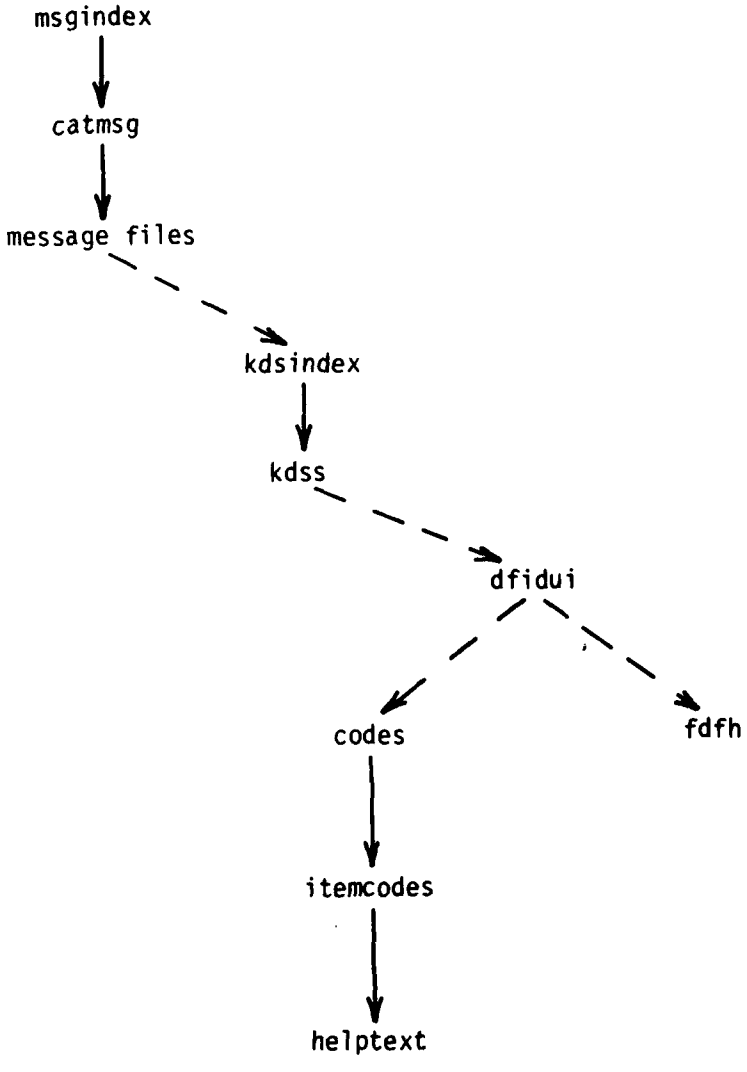


Figure 13. Hierarchical Structure of JAMPS Source Files

The deletion of a message is completed by removing the separate message description file for that message. Any keyword data sets or fields that were used solely by the deleted message may be removed, however, it is not necessary to do so. It may be desirable to retain the message file and the related keyword data sets and fields. This would eliminate unnecessary reentry at a later date should the message be reestablished or a message added which would use those fields and sets.

To add a message to the data base, it is necessary to create a message description file for the message. A new file can be created by issuing the UNIX system "edit" command, followed by the short message name. Upon issuance of the "edit" command, the system will attempt to edit an existing file of that name should one exist, or will create an empty file and assign the given name to that file.

The first entry to the file is the message number, a lower case alphabetic character followed by three digits. This is preceded by the token ".msg". The next line should contain the edition number preceded by the token ".edition". It is necessary that the tokens be included in the file, as the processing routine will search for them and process accordingly. Comments may be added to this file at any place when preceded by a semicolon. The comment field begins at the semicolon and continues to the end of the line. The next entry should be a table in the format discussed in paragraph 2.1.3. The information must be separated by commas. The spacing of the columns is totally arbitrary, however, an easily readable format should be used.

The next entry is the state table. The state table entry must be preceded by the token ".state". This table should be built using

the information provided in the previous table, combined with the information provided in the TIDP. If any fields are to be prepopulated, the state table should be followed by the token ".man", and the table of fields to be populated along with their field entries.

When all entries have been made to the message description file, the **kdsindex** must be checked to ensure that the data base contains all keyword data sets comprising the new message. If any are missing, an appropriate entry must be made to **kdsindex** and subsequently to **kdss**.

3.1.2 Change Keyword Data Set Indicator

Revisions to the TIDP will occasionally remove or impose mandatory or repeatability standards to keyword data sets that previously were not defined in that manner. To make a change to the KDS indicators for a given message, the operator must edit that message's description file. The appropriate flag should be added to or deleted from the corresponding data set. The revision of KDS indicators will effect part or all of the state table. It will be necessary to evaluate the impact of the change and revise the state table accordingly.

3.1.3 Addition/Deletion of Keyword Data Sets

The addition of a new message to the data base will usually entail adding keyword data sets. Keyword data sets may also be added or deleted from existing messages through JINTACCS revisions. In order to add or delete a KDS, **kdsindex** must first be consulted to see if the keyword data set exists in the data base. If a KDS is to be added to **kdsindex**, it is advisable to retain the alphabetic

order, unless the KDS is classified. The index number assigned to the KDS should be the next sequential number in the list. No re-numbering of the KDS numbers is required when adding or deleting a keyword data set.

The addition or deletion of the keyword data sets must be made to the **kdss** file. The deletion of a KDS involves the removal of the corresponding entry from **kdss**. When adding a KDS to the fields, the alphabetic order should be preserved, unless the KDS is classified. The KDS name entry is preceded by ".KDS". This is followed by the KDS type entered as ".LINEAR", ".COLUMNAR", ".FREETEXT" or ".JANAP", followed by ".VERSION" and the version number. This information may be entered as lowercase characters and converted to uppercase characters, using a lower-to-upper processing routine. An uppercase conversion must take place before processing of the JAMPS data tables if **kdss** was entered in lowercase characters. The table of fields and state table are completed in a manner similar to the message table of keyword data sets and state table.

The table of fields should be preceded by ".FIELDS" and the state table preceded by ".STATE". Note that a number of fields may have the same presentation number in the **kdss** file. If the presentation number is repeated, it indicates that the field is an alternate contents field and that all DFI/DUI's listed for that field are valid choices.

A change to the **kdsindex** file, and subsequently to the **kdss** file, may require changes to **dfidui**. The **dfidui** file should be checked to ensure that all the fields entered in **kdss** do exist in the data base.

3.1.4 Change Field Indicator

A change in a field's mandatory or repeatability definition changes the field state table. After changing the field indicator entry in the table of fields, make sure that the state table correctly reflects this change.

3.1.5 Field Changes

If a field is to be added to the data base, at least one entry must be made to **dfidui**. Multiple entries may have to be made to **dfidui** if a chain-type DFI is being added and not all the elemental DFI's comprising the chain-type DFI are in the data base. Entries must be made to the **itemcodes** and **helptext** files. These entries will be discussed in paragraphs 3.2.1 and 3.2.2, respectively. An entry must be made to **fdfh** if the field can appear by itself (i.e., not as part of a chain), and an entry must be made to **codes**.

In adding a DFI/DUI to **dfidui**, the format of the entry is as specified in paragraph 2.3.1. The addition of a DFI/DUI may change the DUI indicator from "com" (common) to "dif" (different) if the new entry introduces new field lengths, left/right justification, or message map characters differing from ones existing in **dfidui**. In this case, all DUI's must be made into separate entries, as specified by the "dif" format.

In adding a DFI/DUI to **fdfh**, the format is that specified in paragraph 2.3.2. In order for the JAMPS system to perform correctly, a field descriptor must be entered for each DFI/DUI. A field header need only be entered if the DFI/DUI is part of a columnar KDS. If a field descriptor is not provided in the MED, the field

header may be entered as both the field descriptor and field header. If neither is provided, an appropriate abbreviation of the DUI name or usage should be entered for the field descriptor and, if required, the field header.

If a field is being added to the data base, an entry must be made to **codes**. The format discussed in paragraph 2.3.3 should be followed in making the addition. Any codes which span more than one line should be examined to determine if a concatenation symbol is necessary.

Changes to data fields may be made through **fdfh**, **dfidui** or **codes**, depending on what is being changed. A change to the message map characters, field length, or left/right justification is made through the **dfidui** file, whereas a change to the field header or field descriptor is made to **fdfh**. A change to the valid code entries for a field is made to **codes**.

3.2 Help Text Revision

Sections of the help text may have to be revised in order to clarify definitions, improve readability, or update the available information. There are two files, **helptext** and **itemcodes**, which, when combined, provide the help text. Revisions to the DFI/DUI definitions would be made in the **helptext** file, whereas changes to specific valid codes would be made in the **itemcodes** file.

3.2.1 Code Changes

The codes that are displayed as part of the help text are in the source file, **itemcodes**. The **itemcodes** file is in the same

format as the codes file and may be revised in the same manner as discussed in paragraph 3.1.5. Initially, codes which spanned more than one line in the MED also span more than one line in the codes and itemcodes files. However, since the entry in the itemcodes file is what is displayed to the operators in the HELP and CONVERSATION modes, it may be desirable to display the code in a more legible form. The code may be changed in the source form so that it appears all on one line. Likewise, this will be the displayed format of the code in the HELP mode. The only requirement is that a line in itemcodes is no greater than 79 characters, including the delimiting asterisks.

3.2.2 DFI Definition Revision

Some of the phrases expressing the essential nature of the meaning of the DFI/DUI's that are published in the MED are verbose and provide superfluous information of a technical nature. Other definitions are brief and to the point. The definitions of the DFI/DUI's which appear in the HELP and CONVERSATION modes are in the source file helptext. Initially these were gathered from the MED definitions. It may be necessary to revise some of the helptext definitions so as to make them informative and concise. The format described in paragraph 2.4.1 should be followed when revising the helptext file.

3.3 JANAP File Update

The three JANAP files containing prepopulation data as described in section 2 can be revised by programs specifically written for that purpose. These programs, which ensure proper formatting and provide for data entry validation, are the subject of the following paragraphs.

3.3.1 Address (Adr) File

This file, which is basic to the other two JANAP data files, can only be altered by the program **jbldadr**. When this program is called, it will first read the entire file from the disk into memory. A menu is then displayed on the CRT which allows the operator to review the entire file, make additions or deletions, or modify existing entries.

The routing indicators are seven alphabetic characters or less, and the ordinary address can be as long as 56 alphanumeric characters. Both of these items are checked before being entered into the file. Any errors will cause the display of an appropriate message and allow for reentry of data.

As changes are being made to the file, all entered data is checked for validity to the extent possible. PLA's must be fourteen characters or less. Each newly added PLA is checked against the file to prevent duplicate names from occurring.

When all changes, deletions and additions have been completed, exiting from the program will cause the updated file to be rewritten on the disk file for use by JAMPS and for future modification.

3.3.2 Originator (Orig) File

The opfac data can be modified through the use of a program named **jbldorig**. This program operates in a manner similar to **jbldadr** by use of a displayed menu and appropriate keyboard inputs to control the program's functions.

Whenever a PLA is changed or added to, as in the case of a new entry, the **Adr** file is used to verify a legal entry. PLA's that do not appear in the **Adr** file are rejected. When a new opfac is being added to the **Orig** file, the new PLA must be added to the **Adr** file, after which it can be used by **jbldorig**.

3.3.3 Routing (Rout) File

The data which makes up the TO and INFO KDS's is found in the opfac files, a separate file for each opfac attached to a JAMPS system. For each message type that can be created and transmitted by the system, a file entry may exist. Each entry for a message type contains, in addition to the form of the message identification, the classification and precedences for that message and two lists of PLA's: one for the TO addressees and the other for the INFO addressees. The number of PLA's is presently limited to twenty for each list.

The program **jbldrtg** provides the means to manage these files. In a manner similar to the other two JANAP programs, a menu and instructions are displayed for operator use. All data entered is checked for validity, and error messages are displayed as appropriate.

SECTION 4

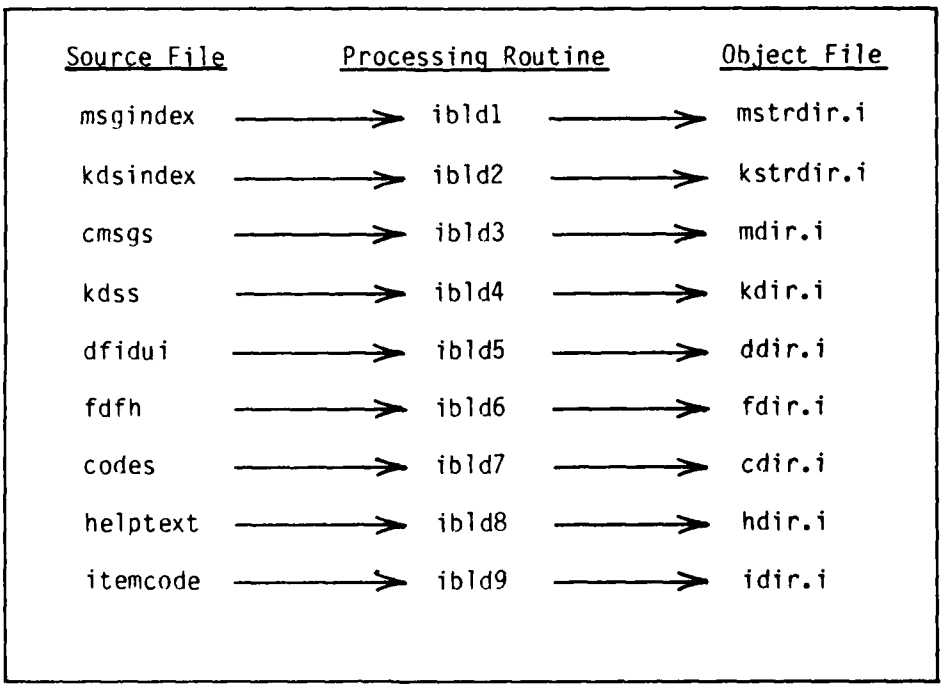
REBUILDING THE JAMPS DATA BASE

In order to use the data in the source files, the JAMPS data base must be rebuilt. The source files which currently exist in the data base have been written in a combination of upper- and lowercase alphabetic characters. Files which were produced by programs extracting data from the SDC print tape appear in uppercase, whereas files that were generated by manual entry were written in lowercase. Whether upper or lowercase is the preferred means of entry, all files must be converted to uppercase before they may be converted to object form. The object file counterparts of the data base source files have names ending in "dir.i". The building and maintaining of the object files has been automated in the JAMPS system by use of the UNIX system's routine "makefile". The building routines, the file interdependencies, and the "makefile" convention are discussed in this section.

4.1 Source Processing Routines

The programs used to build the JAMPS data base are **ibld** programs. The **ibld** programs process the human-readable source files and pack the data in machine-readable object files. There are nine **ibld** programs, one for each type of source file. Table 2 shows the source files, the processing **ibld** programs and the resultant object files.

Table 2. **ibld** Processing Routines



.2 File Interdependencies

In rebuilding the JAMPS data base, the order in which the files are processed is important. The object file **mdir.i** is dependent upon **mstrdir.i** and **kstrdir.i**, that is, **mstrdir.i** and **kstrdir.i** must be built before **mdir.i** can be built. The **kdir.i** file is dependent upon **kstrdir.i**, and **kstrdir.i** must be built before **kdir.i**. The rest of the files are totally independent of each other and the order in which they have been built.

4.3 Makefile

The process of building and maintaining the object files to be used by the JAMPS system has been automated by use of "makefile", which allows the building order of modules, libraries and interfile dependencies to be specified. The JAMPS system "makefile" is named precisely that: **makefile**. Figure 14 shows **makefile** used to process the JAMPS files into object files.

Before any processing is done by **makefile**, all file interdependencies are scanned and checked to see which of the files have been updated. Only the object files specifically associated with the updated versions of the source files and associated dependent object files, if any, are updated. Assume for example, a change has been made to **fdfh**. Rather than rebuilding all the message files, **mstrdir.i**, **kstrdir.i**, **mdir.i**, etc., **makefile** can be used so that only **fdfh**'s object counterpart, **fdir.i**, is updated. Similarly, if **kdsindex** is revised, a call to **makefile** will cause the rebuilding of **kstrdir.i**, **kdir.i** and **mdir.i**. The interdependencies of the object files are included in **makefile**; the object files to be rebuilt are determined from the source files that are updated, with all building taken place sequentially.

```
datatables: mstrdir.i kstrdir.i fdir.i ddir.i kdir.i mdir.i cdir.i hdir.i\
            idir.i
        echo 'data table build and checkout complete'

fdir.i: fdfh
        -upper fdfh cfdfh
        ibld -i <cfdfh
        rm cfdfh

ddir.i: dfidui
        -upper dfidui cdfi
        ibld -i <cdfi
        -rm ibld.str
        -rm ibld.tmp
        rm cdfi

kdir.i: kstrdir.i kdss
        -upper kdss ckdss
        ibld -i <ckdss
        rm ckdss

mdir.i: mstrdir.i kstrdir.i catmsg
        sh catmsg
        -rm cmsgs
        -upper msgs cmsgs
        ibld -i <cmsgs
        rm msgs
kstrdir.i: kdsindex
        -upper kdsindex ckdsindex
        ibld -i <ckdsindex
        rm ckdsindex

mstrdir.i: msgindex
        -upper msgindex cmsgindex
        ibld -i <cmsgindex
        rm cmsgindex

cdir.i: codes1 codes2 codes3 codes4
        cat codes1 codes2 codes3 codes4 >codes
        -ibld7 codes cdir.i

hdir.i: helptext1 helptext2 helptext3 helptext4
        cat helptext1 helptext2 helptext3 helptext4 >helptext
        -ibld8 helptext hdir.i

idir.i: itemcode1 itemcode2 itemcode3 itemcode4
        cat itemcode1 itemcode2 itemcode3 itemcode4 >itemcodes
        -ibld9 itemcodes idir.i
```

Figure 14. **makefile**

The use of **makefile** may be accomplished in two ways. The command "make" followed by an argument may be used, or a shell program called **makeall** may be invoked by the command "sh makeall". The shell program **makeall** in turn issues the command "make", followed by names of the object files that are to be built, i.e., **mstrdir.i**, **kstrdir.i**, **mdir.i**, etc.

The arguments to the "make" command can be the names of any or all of the object files. If all the object files are to be rebuilt, the argument "datatables" may be given rather than listing all of the object files separately. At the end of an editing session, the easiest way to ensure that the data base has been properly rebuilt is by issuing the command "make datatables". If none of the source files have been updated, the system will respond with the message "data table build and checkout complete". The command "make datatables" will build only the object files which correspond with, or are dependent upon, the updated source files.

If any of the source files are not properly completed, the "make" action will be terminated with an error message. By looking at the command given prior to the error message, the building routine that returned the error message can be determined. Likewise, by identifying the building program that gave the error message, the erroneous source file may be determined.

SECTION 5

CHECKOUT PROGRAMS

A number of programs, the **iocheck** programs, have been written to aid in the verification of the contents of the object files. A visual scan of the output produced by any of the programs may be performed to confirm accuracy. All **iocheck** programs utilize the retrieval routines used by the JAMPS display management program so that the **iocheck** programs serve a dual purpose: verifying object files and retrieval routines. A cross check program, **xcheck**, has also been written to filter any conflicting data from the source files on a message-by-message basis. If there are any conflicts in the source file, an error message is outputted. These programs will be discussed in greater detail in the following paragraphs.

5.1 Input/Output Check Programs

The input/output check programs, **iocheck**, take as input a message number, message index number, KDS name, KDS index number or DFI/DUI, depending on which program is being called. The output produced from the **iocheck** programs has been kept compact. Generally, no headers or identifying names are used in presenting the data. The **iocheck** programs, each object file checked, and the required input are presented in table 3.

5.1.1 Message Index Check

The first input/output check program, **iocheck1**, checks the object file **mstrdir.i**, for accuracy. Input to **iocheck1** is the message

Table 3

iocheck Routines

iocheck Programs	Object File	Input
iocheck1	mstrdir.i	Message number or message index number
iocheck2	kstrdir.i	KDS name or KDS index number
iocheck3	mdir.i	Message name or message index number
iocheck4	kdir.i	KDS name or KDS index number
iocheck5	ddir.i (cdir.i)	DFI/DUI
iocheck6	fdir.i	DFI/DUI
iocheck7	cdir.i	DFI/DUI
iocheck8	hdir.i	Directional indicator (0 or 1)
	idir.i	DFI/DUI

number or message index number. If the message number is given for input, the output is the message index number and the edition number. Therefore, upon invoking "iocheck1 A650", the output is "msgnum = 2 edition = 0". If iocheck1 is invoked using the message index number, the output is the message number, short message name, message index number and the edition number. The output from "iocheck1 2" is "A650 APORALOT 2 0".

5.1.2 Keyword Data Set Index Check

The KDS index check, iocheck2, may be invoked using the KDS name or the KDS index number as input. If the KDS name is given for input, the KDS index number and version number is returned. The *command* "iocheck2 6ALOT" returns "kdsnum = 13 version = 0". The output returned when the KDS index number is used for input is the KDS name, the KDS index number and the version number; "6ALOT 13 0" is the returned output from the command "iocheck2 13".

5.1.3 Message Description Check

The input/output check program, iocheck3, checks the message description file. Input to iocheck3 can be the message index number or message number, with the output being the same in either case. The output is in four parts as shown in Figure 15. The first part has general information about the message: the length of the message information in bytes, the message number, the message edition number, the message index number, the number of KDS's in the message, the length of the state table in bytes, and the number of prepopulated field entries. The next part of the output is the KDS table, followed by the state table and the prepopulation information.

```
mmsgln = 712
mmsgnm = A650
mmedit = 0
mmsgnum = 2
mnumkdss =17
mstateln = 75
mnumman = 2
 1 FROM      0 1 0   1    1 0
 2 TO        0 1 1   2    2 0
 3 INFO      0 0 1   3    3 0
 4 CLASS     0 1 0   7    4 0
 5 SIC       0 0 0 171    5 0
 6 EXER      0 0 0 114    6 0
 7 OPER      0 0 0 146    7 0
 8 MSGID     0 1 0 140    8 0
 9 REF       0 0 1 155    9 0
10 CANX      0 0 0  87   10 0
11 PERIO     0 1 0 149   11 0
12 APPOR     0 1 1  75   12 0
13 6ALOT     0 0 0  13   13 0
14 TGTPRI    0 0 1 188   14 0
15 AKNLDG    0 0 0  71   15 0
16 RMKS      0 1 0   6   16 0
17 DWNGRADE 0 0 0 106   17 0
 0  1  1
 1  1  2
 2  3  2  3  4
 3  2  3  4
 4  4  5  6  7  8
 5  3  6  7  8
 6  1  8
 7  1  8
 8  3  9 10 11
 9  3  9 10 11
10  1 11
11  1 12
12  5 12 13 14 15 16
13  3 14 15 16
14  3 14 15 16
15  1 16
16  2 17 -1
17  1 -1
 4  1 0 S E C R E T
 8  1 1 APORALOT
```

Figure 15. Sample Output from **iocheck3**

The KDS table is not in the same order as it appears in the source file, nor does it have an explanatory header. The first column of the table is the KDS presentation number. This column is followed by columns for the KDS name, KDS version, mandatory indicator, repeatability indicator, KDS index number, state number and delete indicator. The mandatory and repeatability indicators have been replaced with "0's" and "1's"; "1" if mandatory or repeatable, "0" otherwise.

The KDS table is followed by the state table. The state table is similar to the source version of the state table minus the separating commas, and with the sharp sign, "#", replaced by "-1". An additional column appears as the second column of the retrieved state table. This column indicates the number of states to which the associated state may traverse.

The table of prepopulation information is presented in the following order: KDS presentation number, field number within the KDS, write-protect/unprotect indicator, and the prepopulation information. Again, the write-protect/unprotect indicators have been replaced by "0's" and "1's"; "1" for write-protect, "0" for unprotected. Again, the output from the commands "iocheck3 2" or "iocheck3 A650" are shown in figure 15.

5.1.4 Keyword Data Set Description Check

The program to check the KDS description files, **iocheck4**, takes as input the KDS name or the KDS index number. As in **iocheck3**, **iocheck4** will provide the same output regardless of the form of the input. The output is in three sections: a general information section, the DFI/DUI table, and a state table. The general information

consists of the length of the KDS information in bytes, the KDS name, the version number, the KDS index number, the KDS type, the number of DFI's, and the length of the state table in bytes. The KDS type is represented as an integer value; "1" represents a linear KDS, "2" represents columnar, "3" represents freetext and "4" represents JANAP.

Again, as with **iocheck3**, the DFI/DUI table is not in the same order as it appeared in the source files, nor does it have any explanatory headers. The order of the table is the presentation number followed by the DFI type ("1" is for a C-type DFI, "2" for an E-type), the DFI number, DUI number, the columnar print position, state number, mandatory indicator, repeatability indicator, mandatory field descriptor indicator, and the delete indicator. Again, "0's" and "1's" are used for the mandatory and repeatability indicators rather than the "M's" and "R's" that were used in the source files.

The state table is similar to the one that appeared in the source files, without the separating commas and with the sharp sign, #, replaced by a "-1". As in the state table retrieved by **iocheck3**, an additional column indicating the number of states to which the associated state may go has been added. Figure 16 shows the output from the commands "iocheck4 13" or "iocheck4 6ALOT".

5.1.5 DFI/DUI Check

The check out program for the DFI/DUI file, **iocheck5**, will also call upon the data item codes file. This allows for a more complete picture of each DFI/DUI. The input to **iocheck5** is the DFI type ("E", "e", "C", or "c"), the DFI number and the DUI number, again, no separation between the DFI type and the DFI number on input.

```
kkdsln = 222
ksetid = 6ALOT
kkvers = 0
kkdsnum = 13
ktype = 2
knumdfis =7
kstateln = 30
 1 2 987  31  2  1 1 0 0 0
 2 2 987  32 16  2 1 0 0 0
 3 2 107   4 30  3 1 0 0 0
 4 2  31  22 35  4 1 0 0 0
 5 2 513   1 40  5 0 0 0 0
 6 2 987  23 47  6 0 0 0 0
 7 2 527   1 61  7 0 0 0 0
 0  1  1
 1  1  2
 2  1  3
 3  1  4
 4  4  5  6  7 -1
 5  3  6  7 -1
 6  2  7 -1
 7  1 -1
```

Figure 16. Sample Output from **tocheck4**

The output of **iocheck5** is rather cryptic. An example of the output from the command "iocheck5 E148 1" appears in figure 17.

The first line of output defines the DUI indicator. The codes for the DUI indicators are the following: "1" is chain-type DFI, "2" is "COMNOTAB", "3" is "COMTAB", "4" is "DIFNOTAB", "5" is "DIFTAB", and "6" is "ALTCHN". The code "6" is no longer in use. As previously defined, a chain-type DFI is a DFI comprised of elemental parts. This is denoted in JINTACCS documentation by a "C" preceding the DFI number. A "COMNOTAB" is a DFI which has been defined in the source files as being a "com" type and does not have discrete character codes for its DUI's. A "COMTAB" is a DFI which has been defined as a "com"-type DFI in the source file and has discrete character codes for the DUI's. Likewise for "DIFNOTAB" and "DIFTAB". A "DIFNOTAB" is a DFI which has been defined as a "dif" type in the source file and has no discrete codes for any of its DUI's. A "DIFTAB" is a DFI which has been defined as being a "dif" type and has discrete codes for its DUI's. A "DIF" is a DFI which has been defined as being a "dif" type in the source file and some of the DUI's have discrete codes and others do not.

The next line of the output has the DFI type and DFI number, followed by the length of the DFI information in bytes and the number of DUI's. The entry for the number of DUI's will be "-1", unless the DFI was defined as being a "dif" type in the source files. In this case, the number of DUI's will be the number of DUI's entered in the source file. The next line, which is only entered if the DFI has a number other than "-1" for the number of DUI's on the preceding line, identifies the DUI that is being retrieved.

```
dfi type = 5

 E148 DIFTAB   ddfiln =  110 dnumduis = 2
DIFTAB DUI = 1
V,L,6,33,A
#CHNROWS = -1 #TABROWS = 10 STATE TAB LN = -1 TAB STR LN = 225
                    T O P   S E C R E T
                            S E C R E T
                C O N F I D E N T I A L
                    R E S T R I C T E D
                                UNCLAS
                  N A T O   C O S M I C
                  N A T O   S E C R E T
        N A T O   C O N F I D E N T I A L
            N A T O   R E S T R I C T E D
        N A T O   U N C L A S S I F I E D
```

Figure 17. Sample Output from **iocheck5**

The next line presents the information about the DUI in the same manner as in **dfidui**. The first character, "V" or "F", represents variable or fixed field lengths, respectively. The next character is the justification flag, "L" for left and "R" for right. This is followed by the minimum and the maximum field lengths. The last character, or characters, are the message map codes. If only one character appears, the entire field is of that type.

The next line contains information pertaining to the number of elements comprising a chain, the number of codes for that particular DFI/DUI, and the length of codes in bytes. These are indicated by "#CHNROWS", "#TABROWS" and "TAB STR LN", respectively. The indicator "STATE TAB LN" is an artifact that was used in handling alternate chains, which will be stripped from the program at some future point. The number of elemental DFI's compressing a chain DFI is the value given for "#CHNROWS". The value following "#TABROWS" is the number of codes for a given DFI/DUI, whereas "TAB STR LN" is the length of the code in bytes. The value of these indicators will be "-1" if it does not apply to the given DFI/DUI. For example, the value after "#CHNROWS" will always be "-1", unless the DFI is a C-type DFI.

The information that follows this will depend on whether the DFI was a C- or E-type DFI. If the DFI is a C type, the next information is the elemental parts of the chain. If the DFI is an E type and has discrete codes, the codes will be the next piece of information, otherwise no further information is presented.

5.1.6 Field Descriptor/Field Header Check

The program to check the field descriptor/field header file, **iocheck6**, receives as input the DFI type ("E", "e", "C", or "c"),

the DFI number and the DUI number. There is no separation between the DFI type and the DFI number. The output consists of the DFI number, the DUI number, the field descriptor, and field header, if there is one. The output from the command "iocheck6 e513 1" is "E513 1 ACTYP ACTYP".

5.1.7 Date Item Codes Check

The check out program for the data item codes file, **iocheck7**, has the input of the DFI type ("E", "e", "C", or "c"), the DFI number and the DUI number. There should be no separation between the DFI type and the DFI number.

The output from **iocheck7** is in two parts: the general information about the DFI/DUI and a list of the valid code entries, if any, for that particular DFI/DUI. The general information consists of the number of codes, the length in bytes of the codes, the minimum code length, the maximum code length, the range type, and the lower and upper bounds of the range. A range type of "-1" indicates no range. A range of integers is indicated by a "1", "2" represents a range of floating point values, and "3" represents a range of octal values.

A sample of the output from the command "iocheck7 e148 1" is shown in figure 18.

5.1.8 Help Text Check

The program to check the help text, **iocheck8**, ensures the correct processing of the help text definitions and the data items and

```
E 148 1
    Number of Codes = 10
 Number of Bytes    = 225
Minimum Code Length = 6
Maximum Code Length = 33
         Range Type = -1
  Lower Range Value = 0
  Upper Range Value = 0
T O P   S E C R E T
S E C R E T
C O N F I D E N T I A L
R E S T R I C T E D
UNCLAS
N A T O   C O S M I C
N A T O   S E C R E T
N A T O   C O N F I D E N T I A L
N A T O   R E S T R I C T E D
N A T O   U N C L A S S I F I E D
```

Figure 18. Sample Output from **iocheck7**

data item codes for a given DFI/DUI. Input to **iocheck8** is the directional indicator, the DFI type ("E", "e", "C", or "c"), the DFI number and the DUI number. The directional indicator can be either "1" or "0"; "1" for DFI/DUI definition followed by data items and data item codes, or "0" for data items and data item codes followed by the DFI/DUI definition. The output, of course, is dependent upon the directional indicator. Figure 19 shows the output from the command "iocheck8 1 e148 1"; and figure 20, output from the command "iocheck8 0 e148 1".

5.2 Cross Check Program

The program **xcheck** is used to cross check the data base on a message-by-message basis. This program begins with the message description file and filters down through all the KDS's within the message and all the fields within the KDS. A check is made that all entries necessary for the message are there and that there is no conflicting information about the message in any of the files. It is worthless to run xcheck unless all the object files, **mstrdir.i, kstrdir.i, mdir.i,** etc., have been successfully built (i.e., no error messages have been outputted when building the files).

The following checks are made by **xcheck** to verify the accuracy of the message:

a. The message description file for the message to be checked is in the data base
b. All KDS's that are referenced in the message description file have an entry in **kdir.i**
c. The DFI's listed for each KDS in a given message have been entered in **ddir.i**

E 148 1

SECURITY CLASSIFICATION

DEFINITION: A CATEGORY ASSIGNED TO CLASSIFIED MESSAGE INFORMATION OR MATERIAL.

ENTER ONE OF THE CODES.

TOP SECRET MESSAGE CLASSIFICATION	T O P S E C R E T
SECRET MESSAGE CLASSIFICATION	S E C R E T
CONFIDENTIAL MESSAGE CLASSIFICATION	C O N F I D E N T I A L
RESTRICTED MESSAGE CLASSIFICATION	R E S T R I C T E D
UNCLASSIFIED MESSAGE CLASSIFICATION	UNCLAS
NATO COSMIC	N A T O C O S M I C
NATO SECRET	N A T O S E C R E T
NATO CONFIDENTIAL	N A T O C O N F I D E N T I A L
NATO RESTRICTED	N A T O R E S T R I C T E D
NATO UNCLASSIFIED	N A T O U N C L A S S I F I E D

Figure 19. Help Text Output: Definition Followed by Data Items and Codes

E 148 1

SECURITY CLASSIFICATION

TOP SECRET MESSAGE CLASSIFICATION	T O P S E C R E T
SECRET MESSAGE CLASSIFICATION	S E C R E T
CONFIDENTIAL MESSAGE CLASSIFICATION	C O N F I D E N T I A L
RESTRICTED MESSAGE CLASSIFICATION	R E S T R I C T E D
UNCLASSIFIED MESSAGE CLASSIFICATION	UNCLAS
NATO COSMIC	N A T O C O S M I C
NATO SECRET	N A T O S E C R E T
NATO CONFIDENTIAL	N A T O C O N F I D E N T I A L
NATO RESTRICTED	N A T O R E S T R I C T E D
NATO UNCLASSIFIED	N A T O U N C L A S S I F I E D

DEFINITION: A CATEGORY ASSIGNED TO CLASSIFIED MESSAGE INFORMATION OR MATERIAL.

ENTER ONE OF THE CODES.

Figure 20. Help Text Output: Definition Preceded by Data Items and Codes

d. An entry is found in fdir.i for each given DFI

e. All DFI's in a linear KDS have a field descriptor entry in fdir.i; all DFI's in a columnar KDS have a field header entry in fdir.i

f. All alternate contents entries in a columnar KDS have the same field header and print positions

g. In a columnar KDS with no alternate contents fields, no field headers are overlapped by the maximum length of the preceding data field or of the preceding field's header

h. The length of the message map characters equals the maximum field length as in ddir.i

i. An entry is found in hdir.i and idir.i for each DFI/DUI

j. An entry is found in cdir.i for each DFI/DUI

k. The value of the minimum and maximum code length that appears in cdir.i falls within the limits defined in ddir.i

l. All E-type DFI's comprising a C-type DFI are entered in ddir.i

m. The length of the message map characters of a C-type DFI is equal to the sum of the lengths of the message map characters of the comprising E-type DFI's

n. If a C-type DFI is of variable length, at least one of the comprising E-type DFI's is of variable length

o. If a C-type DFI is of fixed length, all the comprising E-type DFI's are of fixed length

p. The sum of the minimum field lengths of the comprising E-type DFI's in a C-type DFI equals the minimum field length of the C-type DFI

q. The sum of the maximum field lengths of the comprising E-type DFI's in a C-type DFI equals the maximum field length of the C-type DFI

In order it run the **xcheck** program, the command "xcheck" may be invoked, followed by the message index number or the message number of the message that is to be cross checked. If all the messages in the data base are to be cross checked, the command "xcheckall" will run **xcheck** for all the messages found in **mstrdir.i**. If an error is found in the object files, an error message will be outputted.

5.3 Data Base Error Debugging

The error messages which are outputted by **xcheck** give an indication to where the problems lie in the source and object files. When inserting messages into the data base from scratch, it is advisable to run **xcheck** one message at a time rather than using **xcheckall**. Error messages may be outputted with a cascading effect; i.e., one actual error may manifest itself as a number of error messages. In correcting the errors that have been outputted, it is advisable to handle them in the order that they are produced. This is due, again, to the cascading effect of the error messages; by fixing the first few occurrences, all errors may be eliminated. Table 4 shows the error messages which are outputted from **xcheck**, the source files that have to be changed to correct the error, and the corrective actions to be taken.

Table 4

xcheck Error Message Output

xcheck ERROR MESSAGE	SOURCE FILE	CORRECTIVE ACTION
*** ERROR *** kds number (index number) not found	**kdsindex**	Add the KDS index (kds number) to **kdsindex**
*** ERROR *** ("E" or "C") (DFI)(DUI) not found in **fdfh** tables	**fdfh**	Make entry of DFI/DUI to **fdfh**
*** ERROR *** missing field header ("E" or "C") (DFI) (DUI) set (KDS name)	**fdfh**	Enter field header for DFI/DUI
*** ERROR *** missing field desc ("E" or "C") (DFI) (DUI) set (KDS name)	**fdfh**	Enter field descriptor for DFI/DUI
*** ERROR *** All A/C print pos's must be identical: ("E" or "C") (DFI)(DUI) set (KDS name)	**kdss**	Change print position entries so all are same in **kdss**
*** ERROR *** All A/C fld hdrs must be identical: ("E" or "C") (DFI) (DUI) set (KDS name)	**fdfh**	Change entry in **fdfh** so all field headers are for alternate contents fields
*** ERROR *** Prior field's max of (maximum field length) > field space of (field space from print positions) ("E" or "C") (DFI) (DUI) set (KDS name)	**kdss** **dfidui**	Determine if error is in the maximum field length entry in **dfidui** or in the print position in **kdss** (only appears in columnar sets)
*** ERROR *** Prior fields header length too long for field: ("E" or "C") (DFI) (DUI) set (KDS name)	**fdfh** **kdss**	Determine if problem is field header or print position

Table 4 (Continued)

xcheck Error Message Output

xcheck ERROR MESSAGE	SOURCE FILE	CORRECTIVE ACTION
*** ERROR *** ("E" or "C") (DFI) (DUI) in set (KDS name) not found in **dfi** tables	**dfidui**	Make DFI/DUI entry to **dfidui**
*** ERROR *** msg map lngth (length of msg map) ! = field max (max field length) in ("E" or "C") (DFI) (DUI) set (KDS name)	**dfidui**	Fix message map or maximum field length, whichever is in error
*** ERROR *** request error *in gethelp ask* (("E" *or* "C")) (DFI) (DUI) got (("E" or "C")) (DFI) (DUI)	**hdir.i**	FATAL ERROR - consult system engineer (should not occur!)
*** ERROR *** ("E" or "C") (DFI) (DUI) in set (KDS name) caused error return in **gethelp**	**hdir.i**	FATAL ERROR - consult system engineer (should not occur!)
*** WARNING *** missing help in **hdir.i** ("E" or "C") (DFI) (DUI) in set (KDS name)	**helptext**	Make addition in text **helptext** for missing DFI/DUI
*** WARNING *** Missing data item/codes in **idir.i** ("E" or "C") (DFI) (DUI) in set (KDS name)	**itemcodes**	Make addition in **itemcodes** for missing DFI/DUI
*** WARNING *** missing entries in **hdir.i** of **idir.i** ("E" or "C") (DFI) (DUI) in set (KDS name)	**helptext** **itemcodes**	Make addition to **helptext** and **itemcodes** for missing DFI/DUI
*** ERROR *** dberr detected *in retrieving* E (DFI) (DUI) in set (KDS name) from data item code tables	**idir.i**	FATAL ERROR - consult system engineer (should not occur!)

Table 4 (Continued)

xcheck Error Message Output

xcheck ERROR MESSAGE	SOURCE FILE	CORRECTIVE ACTION
*** WARNING *** E (DFI) (DUI) in set (KDS name) missing from cdir.i	**codes**	Make addition to codes for missing DFI/DUI
*** ERROR *** requested E (DFI) received E (DFI)	cdir.i	FATAL ERROR - consult system engineer (should not occur!)
*** ERROR *** Bad maxlen E (DFI) (DUI) set (KDS name) actual = (max. field length) specified = (max. code length)	**dfidui** **codes**	Determine if error is maximum field length entered in **dfidui** or one of the codes
*** ERROR *** Bad minlen E (DFI) (DUI) set (KDS name) actual = (min. field length) specified = (min. code length)	**dfidui** **codes**	Determine if error is minimum field length entered in **dfidui** or one of the codes
*** ERROR *** component E (DFI) (DUI) of C (DFI) (DUI) not found in dfi tables	**dfidui**	Make entry to **dfidui** for missing DFI/DUI
*** ERROR *** C (DFI) (DUI) is f type E (DFI) (DUI) is v type	**dfidui**	Change C-type DFI to variable or E-types to fixed
*** ERROR *** C (DFI) (DUI) v type, all E's are f type	**dfidui**	Change C-type DFI to is fixed or E-type DFI's to variable
*** ERROR *** C (DFI) map = (length) not equal to sum E map = (length)	**dfidui**	Determine if error is C-type DFI message map or E-type DFI's message map
*** ERROR *** C (DFI) (length) not equal to sum E max = (length)	**dfidui**	Determine if error is max = maximum length of C-type DFI or maximum length of E-type DFI

Table 4 (Concluded)

xcheck Error Message Output

xcheck ERROR MESSAGE	SOURCE FILE	CORRECTIVE ACTION
*** ERROR *** C (DFI) min = (length) not equal sum E min = (length)	dfidui	Determine if error is minimum length of to C-type or minimum length of E-type DFI

SECTION 6

SOURCE DOCUMENTS

This section describes the source documents used to create and make changes to the data base. The three primary documents are the Technical Interface Design Plan (TIDP), the Catalog of Data Sets, and the Message Element Dictionary (MED). Each of these sources is described in the following paragraphs.

6.1 Technical Interface Design (TIDP)

The TIDP, which is a classified document, contains descriptions of each of the JINTACCS messages. Figure 21 is a typical page from the TIDP, and shows the makeup of the APORALOT message. The first main text line of the figure, entitled MESSAGE NUMBER, gives the message number, A650 and short message name for the APORALOT message. The next line of this figure gives the full name or title, Apportionment/Allotment message. Following the general description of the purpose of this message, as shown in this figure, is a list of the keyword data sets for this message under the column heading SET IDENT. Since this message actually covers three pages in the TIDP, only a portion of the KDS's are shown in this figure. Those that are shown are the standard introductory sets which are contained in all messages.

Two of the standard introductory data sets, EXER and OPER, have restrictions on their use which are not shown in figure 21. Figure 22 is an extract from part of the special instructions contained in

R1 INTACTS TIDP-TE VOL VI

MESSAGE CONTENT

MESSAGE NUMBER: 4553 APORAL_3T PAGE 1 OF 3

TITLE: APPORTIONMENT/ALLOTMENT MESSAGE

PURPOSE: TO PROVIDE GENERAL GUIDELINES AND PRIORITIES TO COMPONENT COMMANDERS FOR THE PLANNED USE OF AIR RESOURCES AND TO ALLOT FORCES FROM ONE COMPONENT TO ANOTHER.

SET IDENT	CAT	FIELD NO	MANDATORY ENTRY/ FLD DESC/COL HEADER	FIELD NAME	START COL #	NO/TYPE		OFI NO	DUI NO
(CLASS)	M								
	M	1		SECURITY CLASSIFICATION	5-31	AB	E	148	001
	O	2		DESIGNATOR	1-30	A	F	587	001
SIC	C								
	M	1		BASIS CODE		1 A	F	332	001
	O/R	2		BASIS CODE		3 A	F	332	001
EXER	C								
	M	1		EXERCISE NICKNAME	1-56	AN3S	F	335	001
	O	2		EXERCISE MESSAGE ADDITIONAL IDENTIFIER	1-14	A	E	335	002
OPER	C								
	M	1		OPERATION CODEWORD	1-32	A	E	336	001
	O	2		PLAN DESIGNATOR AND NUMBER	3-23	ANS	F	925	001
	O	3		OPTION NICKNAME	1-23	ANAS	E	585	001
	O	4		SECONDARY OPTION NICKNAME	1-23	ANAS	E	585	002
MSGID	M								
	M	1	APORAL3T	MESSAGE TYPE	1-20	ANAS	F	357	001
	M	2		ORIGINATOR	1-23	ANAS	F	146	001
	O	3		SERIAL NUMBER		7 N	E	147	001
	O	4		MONTH		3 A	E	587	001
	O	5		QUALIFIER		3 A	F	588	001
	O	6		SERIAL NUMBER OF QUALIFIER	1-	3 N	E	497	017
REF	O/R								
	M	1		MESSAGE TYPE	1-20	ANAS	C	353	001
	M	2		ORIGINATOR	1-23	ANAS	F	146	001
	M	3		DATE-TIME OF REFERENCE		7 AN	C	143	005
	M	4		MONTH		3 A	E	590	001
	O	5		SERIAL NUMBER		7 N	E	147	001
	O	6		SIC		5 A	E	332	002
	O	7		SPECIAL NOTATION		5 A	E	1342	001

UNCLASSIFIED

Figure 21. Typical Page from TIDP - Message Contents

MESSAGE NUMBER: All JINTACCS Character-Oriented Messages (U)

TITLE: Standard Introductory Sets [SIS] (U)

SPECIAL INSTRUCTIONS (continued) (U)

(U) Set Identifier: (CLASS) [M] (continued)

b. For NATO messages, use only those data items and codes listed under the heading of "SPECIAL HANDLING" (DFI #E587, DUI 001).

(U) Special Identifier: SIC [C]. The use of this set is mandatory to define the subject matter of the message when attached to NATO Commands. This set is not used in intra-U.S. messages. Note that there is no field marker following the set identifier. An end of set marker is not used with this set.

Field 1, NASIS Code [M]. Enter the appropriate NASIS code as determined from NATO Supplement 2 to ACP 121. (DFI #E332, DUI 001).

Field 2, NASIS Code [O,R]. The NASIS code field is repeated as necessary to describe the message contents (DFI #E332, DUI 001).

(U) Set Identifier: EXER [C]. This set is conditional upon the message being applicable to an exercise. **This set is not used for operational purposes or if an "OPER" set is used.**

Field 1, Exercise Nickname [M]. Enter the code name or nickname of the exercise to which the message pertains (DFI #E335, DUI 001).

Field 2, Exercise Message Additional Identifier [O]. Enter an additional exercise nickname or identifier (DFI #E335, DUI 002).

(U) Set Identifier: OPER [C]. This set is conditional upon the message being applicable to an operation that has a code name or nickname. It is not used if an operation name has not been established, and **it is not used for exercise purposes or if the set "EXER" is used.**

Field 1, Operation Codeword [M]. Enter the assigned operation name or nickname as established by appropriate authority (DFI #E336, DUI 001).

2-15

Figure 22. Extraction of Message Instructions from TIDP

the TIDP. From these instructions it is seen that EXER and OPER are mutually exclusive, that is, if one is used, the other cannot be used in that message even though the category for the two KDS's listed in figure 21 shows both KDS's as being mandatory. It is important that all special instructions of the TIDP are checked before attempting changes to the data base.

Continuing with the description of the APORALOT message as found in the TIDP, figure 23 provides the list of KDS's that form this message and are not part of the standard introductory set. As shown in this figure under the SET IDENT column heading, the first KDS is CANX. Moving to the next column, the "C" on the same line is the KDS for the category for that KDS, which in this case is conditional. The next KDS, PERID, has an "m" in this column, making that KDS mandatory. The APORALOT KDS is listed in this column as both mandatory and repeatable.

Following in this category column are individual qualifiers for each field within a KDS. In the case of CANX, there are a total of seven fields as shown in the next column, the first four of which are mandatory and the last three are optional. For the APORALOT KDS, fields 3 and 4 have been designated both mandatory and repeatable. The KDS 6ALOT is an example of the columnar KDS identified by the numerical prefix to its name. In the column to the right of the field numbers of this KDS are the nine column headers that will appear in this message. The interpretation of data in the "MANDATORY ENTRY/FLD DESC/COL HEADER" column can easily lead to confusion. When data appears in this column and the associated KDS is columnar, the data is column header. For linear KDS's, data entries in this column ending with a colon ":" are field descriptors and will precede the field in the message. A linear entry without the colon is a mandatory entry for that field.

UNCLASSIFIED

TEST NUMBER: 45 PAGE 2 OF 3

TITLE: APPORTIONMENT REALLOTMENT MESSAGE

SET IDENT	CAT	FLD	MANDATORY ENTRY / FLD DESCRIPTOR	ITEM NAME	START COL	NO/TYPE	DFI NO	DUI NO
CAMX								
	M	1		MESSAGE TYPE		1-20 ANBS	E 350	001
	M	2		ORIGINATOR		1-20 ANBS	E 146	001
	M	3		DATE-TIME OF REFERENCE		7 AN	C 143	003
	M	4		MONTH		3 A	E 599	001
		5		SERIAL NUMBER		7 N	F 147	001
	O	6		SIC		3 A	F 332	002
	O	7		SPECIAL NOTATION		5 A	E 1342	001
PERID	M							
	M	1		FROM		7 AN	C 143	006
	M	2		TO		7 AN	C 143	007
APPOR	M,R							
	M	1		COMPONENT COMMAND		1-15 ANBS	E 987	318
	M	2		PERCENT OR NUMBER INDICATOR		2 A	E 922	001
	M,R	3		AIR MISSION TYPE		2- 4 A	F 107	004
	M,R	4		PERCENTAGE ASSIGNED		1- 2 N	E 727	016
				NUMBER ASSIGNED		1- 4 AN	E 331	017
SALOT	C							
	M	1	LOSE	LOSING COMPONENT COMMAND	2 L	1-13 ANBS	E 987	031
	M	2	GAIN	GAINING COMPONENT COMMAND	15 L	1-13 ANBS	E 987	032
	M	3	AMSN	AIR MISSION TYPE	31 L	2- 4 A	F 107	004
	M	4	NOAC	NUMBER OF AIRCRAFT	35 R	1- 3 N	F 331	022
	C	5	ACTYP	AIRCRAFT TYPE	40 L	2- 6 AN	F 513	001
	C	6	ACUNTID	AIRCRAFT UNIT IDENTIFICATION	47 L	1-13 ANBS	F 987	023
	C	7	ICAO	LOCATION IDENTIFIER, ICAO	51 L	4- 6 A	E 527	001
TGTPRI	O,R							
	M	1	AMSN:	AIR MISSION TYPE		2- 4 A	E 107	004
	M,R	2		TARGET TYPE		2- 6 A	E 319	001
				RECONNAISSANCE TARGET CODE		2-12 AN	C 183	001
				GENERIC TYPE		3- 7 A	E 209	002
				TYPE OF ENEMY MISSION		3- 5 A	F 930	011
ACKLDG	C							
	M	1		ACKNOWLEDGE INDICATOR, MANUAL		1 A	F 319	003

UNCLASSIFIED

Figure 23. TIDP Message Definitions

Figure 24 is part of the message map for the A650 message. The line following the title line is there for convenience of determining the length and starting position of any field within the message. The map itself begins on the line following this index. That line is the classification KDS, showing it is two fields. The first field is three characters of alphabetic information, and after a single blank, a thirty-character field, also of alphabetic information. The seventh line in this map shows the format for the MSGID KDS. Field 1 of this KDS is shown filled in with a mandatory entry, the message name, APORALOT.

Returning briefly to figure 21, and checking the MSGID KDS in field 1, a mandatory entry is shown. This data must appear in the message. The last field in the MSGID KDS consists of three numerical characters as specified by the "nnn" in the message map. The message map shown in figure 25 provides an example of a columnar data set and how it should appear in the message. The 6ALOT KDS is shown with the seven column headers that appear in this KDS, as shown in figure 23. An alternate contents field is shown in figure 25. Field 2 of TGTPRI KDS has four alternate entries that can be used for this field, each of different lengths.

6.2 Catalog of Keyword Data Sets

The Catalog of Keyword Data Sets is an extract of information from the TIDP. When making additions to the JAMPS data base, either the TIDP or the Catalog of Keyword Data Sets may be used as the information source. The advantage of using the Catalog over the TIDP is that the Catalog is not classified, and as such is more convenient.

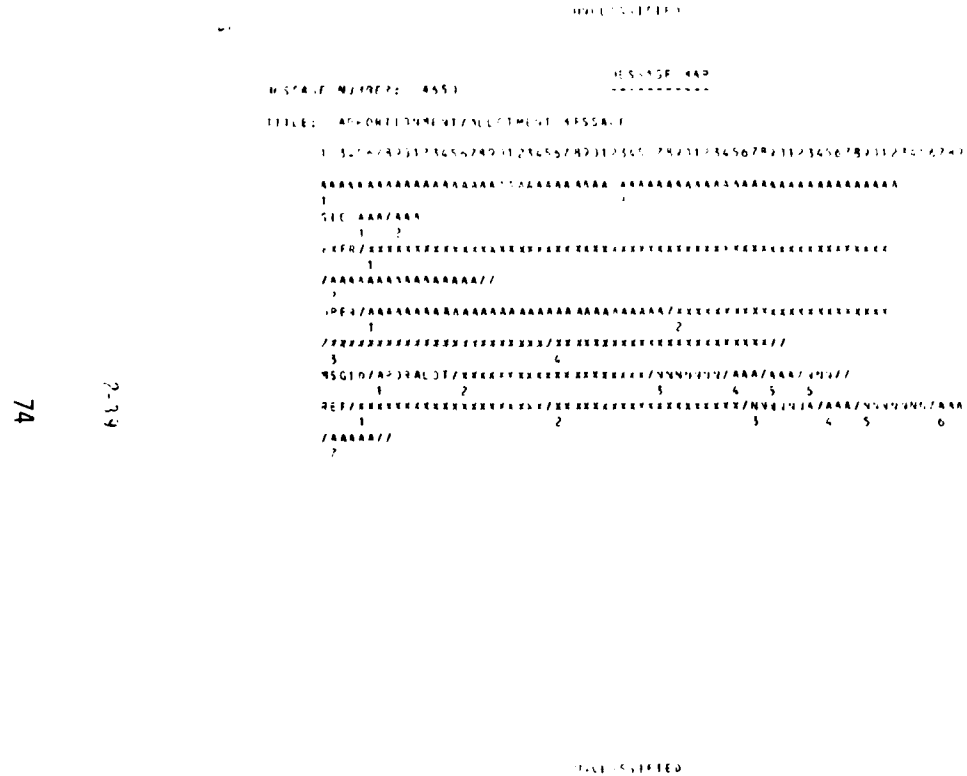


Figure 24. TIDP Message Map

Figure 25. TIDP Message Map Showing Alternate Contents Field

A page from the Catalog of Keyword Data Sets is reproduced in figure 26. The KDS name can be found in the column designated SET IDENTIFIER, and in this example is ACLOST.

The KDS description contains a map similar to that for the message description found in the TIDP (see figure 24). In this example, two of the fields (fields 4 and 5) have alternates. When a field of a KDS has alternates, the longest field size is shown in the map, and all choices are shown below in their proper size and with the proper data designators. For field 5, the five alternates range in size from a length of eight alphanumeric characters to a maximum of twenty.

Following SET MAP, as shown in this figure, is SET CONTENT. This portion of the figure defines the category of each field. The meanings here are the same as used for messages discussed earlier. An "M" is a mandatory entry, an "R" signifies repeatable, and a "C" is conditional. (Note that when building a data base for JAMPS, a "C" is never entered; a blank is interpreted as conditional.)

The three columns to the right of the category column give the DFI/DUI number of the element. The "NO/TYPE" column specifies the number of characters in the field, both a minimum and a maximum can be specified, and the data type. The data types presented in the SET MAP, however, are more accurate and should be used.

An example of a columnar KDS from the Catalog is given in figure 27. The set of column headers as listed in the TIDP for a message using this KDS are shown in their proper orientation in the "SET MAP".

UNCLASSIFIED

JINTACCS MS

CAT EXTRACT, VOL VI (U) 1JUL81

ACLOST
PAGE 1 OF 2

SET IDENTIFIER ACLOST

SET TITLE: FRIENDLY AIRCRAFT LOSS REPORT

STATUS:

SET MAP: ACLOST/XXXXXXXXXXXXXXXXX/XXXXX</AAAAAA/TAILNR:XXXXXXXXXX
1 2 3 4
TAILNR:XXXXXXXXXX
SIDENR:XXXXX

/NNNSXXXXXXXXXXXXXXXNNN/AAA/NNNNNNA:XXXXXXXXXXX/POB:NNN/AAAA//
5 6 7 8 9 10
NNNNANNNNNA
AAAANNN
NNNSXXXXXXXXXXXXXXXNNN
XXXXXXXXXXX
NNAAANNNNNN

SET CONTENT

FIELD	CAT	DFI NO	DUI NO	U	NO/TYPE	DUI NAME
1	M	E 002	001		1-17 ANBS	CALL SIGN
2	C	E 513	001		2- 6 AN	AIRCRAFT TYPE
3	C	E 467	001		1- 6 A	COLOR
4	C	E 465	001		1-10 AN	TAIL NUMBER, AIRCRAFT
OR	C	E 1040	001		2- 5 AN	AIRCRAFT SIDE NUMBER
5	M	C 469	010		11 AN	ACTIVITY LOCATION, LAT/LONG
OR	M	C 497	013		8 AN	ACTIVITY LOCATION, GEOREF
OR	M	C 498	003		7-20 ANBS	ACTIVITY LOCATION, BEARING/ RANGE
OR	M	E 500	035		1-11 ANBS	ACTIVITY LOCATION NAME
OR	M	C 542	004		11 AN	ACTIVITY LOCATION, UTM
6	C	E 128	001		3 A	LOCATION QUALIFIER
7	M	C 143	008		7 AN	TIME OF ACTIVITY
8	C	E 043	001		3-11 AS	CAUSE OF LOSS OF AIRCRAFT
9	M,R	E 031	061		1- 3 N	PERSONNEL ON BOARD
10	M,R	E 044	001		3- 4 A	DOWNED AIRCRAFT CREW STATUS

ACLOST - 1 (REVISED 1CJUN81)

UNCLASSIFIED

77

Figure 26. Typical Page from Catalog of Keyword Data Sets

UNCLASSIFIED

CAT EXTRACT, VOL VI 01 1JUL81 JINTACCS MS

7FACSCD
PAGE 1 OF 1

SET IDENTIFIER 7FACSCD

SET TITLE: FORWARD AIR CONTROLLER AIRCRAFT ALLOCATION SCHEDULE

STATUS:

SET MAP: 7FACSCD

/ATKACCS	MSNNO	TOT	NOTEAC	ORD	HOLDLOC
/XXXXXXXXXXX	XXXXXX	NNNNNNA	NXXXXXXX	XXXXXX	NNNNANNNNNA//
1	2	3	4	5	6
		ASNNNN		XXX	NNNNANNNNNA
		NNNNNNA		XXXXXX	AAAANNNN
		AAAAA			XXXXXXXXXX
					NNAAANNNNNN

SET CONTENT:

FIELD	CAT	DFI NO	DUI NO	J	NO. TYPE	DUI NAME
1	M	E 002	031	L	1-12 ANBS	ATTACK AIRCRAFT CALL SIGN
2	M	E 001	001	L	1 6 ANBS	MISSION NUMBER
3	M	C 084	002	R	6 ANS	TIME ON ARGET
OR	M	C 143	047	L	7 AN	DAY-TIME ON TARGET
OR	M	E 617	002	L	5 A	TIME ON ARGET
4	M	C 496	001	L	8 AN	NUMBER AND TYPE OF AIRCRAFT
5	M	E 495	001	L	1- 4 AN	ORDNANCE LOAD CODE
OR	M	E 532	002	L	5- 6 AN	ORDNANCE TYPE
6	C	C 469	014	L	11 AN	HOLDING POINT, LAT/LONG
OR	C	C 497	008	L	8 AN	HOLDING POINT, GEOREF
OR	C	E 500	040	L	1-11 ANBS	HOLDING POINT NAME
OR	C	C 542	006	L	11 AN	HOLDING POINT, UTM

MESSAGE APPLICABILITY.

MESSAGE NAME	MESSAGE NO	TIDP USAGE BY VOL REF NUMBER
CROSSCAT	A653	VI
JSUPREQ	D668	VI
JSARREQ	D669	VI

7FACSCD - 1 (REVISED 1CJUN81)

UNCLASSIFIED

Figure 27. Columnar KDS from Catalog of Keyword Data Sets

The remainder of the information is as described for a linear KDS in the preceding paragraphs, except for the "J" column of the "SET CONTENT" section. For a columnar KDS, the "J" column contains an "L" or an "R" for left or right justification of the field entry.

6.3 Message Element Dictionary (MED)

The MED extract contains all necessary information relating to elements which make up the KDS's. The MED is extracted from a specific volume of the TIDP, in this case Volume IV, and contains only those DFI/DUI's that are referenced by that volume of the TIDP.

There are two basic types of DFI/DUI's to be considered. First is the chain type, identified by the starting letter "C". Figure 28 is a page from the MED describing a chain-type DFI/DUI, in this case C 143, which has as its DFI name, Military Day-Time. The elemental, or "E"-type, DFI/DUI's which make up this chain are listed in the central part of this figure, and consist of four elements. For each element, its DFI and DUI numbers are given, as well as the DUI name. In the right-most column, the length of each field is provided, and the type of data to be used. For example, the first element is E 023, and its DUI number is 001. The element name is Day, and the field consists of two numeric characters.

Following the description of each DFI/DUI is a set and message applicability section. This information for the C 143 DFI is shown in figure 29. Information is presented here for each DUI of the DFI. In this figure the applicable messages for DUI's 001, 002, 003 and 005 are listed. For DUI 005, there are two KDS's which reference the DFI, the first being CANX, and the second, REF. This provides a complete cross reference from chain-type DFI's to KDS's to messages.

UNCLASSIFIED

MED EXTRACT, VOL VI R1, 1JUL81 C 143 PAGE 1 OF 10

DFI NO	DFI NAME	DFI ABBREVIATION	STATUS	NO/TYPE
C 143	MILITARY DAY-TIME	MIL-DAY-TIM	US PROPOSED	7 AN

DATA STANDARD:

USAGE: AIR OPS
VOL VI

DEFINITION

THE DAY OF A MONTH AND TIMEKEEPING IN HOURS AND MINUTES OF THE TWENTY-FOUR (24) HOUR PERIOD OF THE CALENDAR DAY, USING THE 24-HOUR CLOCK SYSTEM AND AN ASSOCIATED TIME ZONE.

SEQUENCE OF ELEMENTS WITHIN A CHAIN

DFI NO	DUI NO	ELEMENT DUI NAME	NO/TYPE	REMARKS
E 023	001	DAY	2 N	
E 024	001	HOUR (TIME)	2 N	
E 025	001	MINUTE (TIME)	2 N	
E 026	001	TIME ZONE	1 A	

DATA USE IDENTIFIER

DATA USE IDENTIFIER	*FLD DESC/COL HEADER*	*EXPLANATION*
001 MILITARY DAY-TIME	MILDTM	THE DAY OF A MONTH AND TIMEKEEPING IN HOURS AND MINUTES OF THE TWENTY-FOUR (24) HOUR PERIOD OF THE CALENDAR DAY, USING THE 24-HOUR CLOCK SYSTEM AND AN ASSOCIATED TIME ZONE.
[7 AN] [LEFT JUSTIFY]	MILDTM	

DFI NAME MILITARY DAY-TIME (REVISED 6JUL81) DFI NO C 143

UNCLASSIFIED

Figure 28. Typical Page from MED

UNCLASSIFIED

MED EXTRACT, VOL VI R1, 1JUL81 C 143 PAGE 5 OF 10

DFI NO	DFI NAME	DFI ABBREVIATION	STATUS	NO/TYPE
C 143	MILITARY DAY-TIME	MIL-DAY-TIM	US PROPOSED	7 AN

SET AND MESSAGE APPLICABILITY BY DUI(CON'T)

DUI	SET	MESSAGES			
001	TIMEAMP	D660	D666	D667	D668
		JAIRREQ	JESCREQ	JRECREQ	JSUPREQ
		D669			
		JSARREQ			
002	BEACON	A653	A770	D600	
		CROSSDAT	ALORD	JCASINTREQ	
003	BEACON	A653	A770	D600	
		CROSSDAT	ALORD	JCASINTREQ	
005	CANX	A653	A650	A651	A652
		HELOALDAT	APORALOT	EMPLOYALOC	SORTALOT
		A653	A661	A690	A691
		CROSSDAT	REQSTATASK	TACOPDAT	TECHOPDAT
		B702	B703	B704	B705
		JLNCHREP	AIREVENT	ABCHANGE	ACSAMSTAT
		D600	D630	D634	D660
		JCASINTREQ	ALREQ	HELOALREQ	JAIRREQ
		D666	D667	D668	D669
		JESCREQ	JRECREQ	JSUPREQ	JSARREQ
		F625	F631	F632	F636
		REQCONF	ALMSNSCD	FLTCONTINFO	HELOALCONF
		F654	F750	F754	
		CROSSCONF	DESIGARE1	FLTPLN	
	REF	A635	A650	A651	A652
		HELOALDAT	APORALOT	EMPLOYALOC	SORTALOT
		A653	A661	A690	A691
		CROSSDAT	REQSTATASK	TACOPDAT	TECHOPDAT
		A770	B702	B703	B704
		ALORG	JLNCHREP	AIREVENT	ABCHANGE
		B7C5	B711	C001	C420
		ACSAMSTAT	ENGSTS	MSGCHANGEREP	SARSIT

DFI NAME MILITARY DAY-TIME (REVISED 6JUL81) DFI NO C 143

UNCLASSIFIED

Figure 29. Extract From MED for Message Applicability

An elemental, or "E"-type DFI is shown in figure 30. A DFI refers to a family of related items. Each item is distinguished by a different DUI number. In this example, the root of the family, i.e., the DFI, is named "Percentage". The four DUI's of this family shown in the figure are "PERCENTAGE", "PERCENT DAMAGE", "PERCENT DESTROYED" and "PERCENT COVERAGE", respectively. Each of the four DUI's is specified to be one to three numerical characters in length, and right-justified in their use.

Some of the DFI's have specified data entries that can be made in the field. Figure 31 is an example of an elemental DFI that has a defined set of codes that can only be used when filling the defined field. In this example, the columns "DATA ITEM" and "CHAR CODE" provide a guide to what may be entered in the field.

MED DRAFT 13, 10 NOV 1980 E 027 PAGE 1 OF 3

DFI NO	DFI NAME	DFI ABBREVIATION	STATUS	NO/TYPE
E 027	PERCENTAGE	PRCNTG	JINTACCS STANDARD US PROPOSED	1- 3 N

DATA STANDARD: DOD PE-JM (MOD)

USAGE: INTEL VOL II; FIRE SUP VOL III; AIR OPS VOL VI

DEFINITION

THE RATE OR PROPORTION IN EVERY HUNDRED OF THE VALUE (IN HUNDREDS) RESULTING FROM THE DIVISION OF ANY NUMBER BY A BASE NUMBER WITH WHICH COMPARED.

DATA USE IDENTIFIER

DATA USE IDENTIFIER		FLD DESC/COL HEADER	EXPLANATION
001 PERCENTAGE [1- 3 N]	[RIGHT JUSTIFY]	PRCNTG	THE RATE OR PROPORTION IN EVERY HUNDRED OR THE VALUE (IN HUNDREDS) RESULTING FROM THE DIVISION OF ANY NUMBER BY A BASE NUMBER WITH WHICH COMPARED.
002 PERCENT DAMAGE [1- 3 N]	[RIGHT JUSTIFY]	PCTDAM PCTDAM	A SUBJECTIVE EVALUATION OF THE FRACTION OF A TARGET DAMAGED AND UNUSABLE THAT IS BELIEVED TO BE REPAIRABLE EXPRESSED IN PERCENT.
003 PERCENT DESTROYED [1- 3 N]	[RIGHT JUSTIFY]	PCTDES PCTDES	A SUBJECTIVE EVALUATION OF THE FRACTION OF A TARGET THAT IS SO DAMAGED THAT IT CANNOT FUNCTION AS INTENDED NOR BE RESTORED TO A USABLE CONDITION EXPRESSED IN PERCENT.
004 PERCENT COVERAGE [1- 3 N]	[RIGHT JUSTIFY]	PCTCOV PCTCOV	THE FRACTION OF GROUND AREA REPRESENTED ON IMAGERY, PHOTOMAPS, MOSAIC MAPS OR OTHER GEOGRAPHIC

(CON'T ON NEXT PAGE)

Figure 30. Extract From MED for E-Type DFI's

UNCLASSIFIED

MED EXTRACT, VOL VI RL, 1JUL81 — E 471 PAGE 1 OF 2

DFI NO	DFI NAME	DFI ABBREVIATION	STATUS	NO/TYPE
E 471	AIR MISSION RESULT	AIR-MSN RES	JINTACCS STANDARD US PROPOSED	3- 6 A

DATA STANDARD:

USAGE: AIR OPS
VOL VI

DEFINITION

THE PILOT'S ASSESSMENT OF THE RESULT OF AN AIR MISSION.

DATA USE IDENTIFIER

DATA USE IDENTIFIER	FLD DESC/CODE HEADER	EXPLANATION
001 AIR MISSION RESULT [3- 6 A] [LEFT JUSTIFY]	RESULT RESULT	THE PILOT'S ASSESSMENT OF THE RESULT OF AN AIR MISSION.

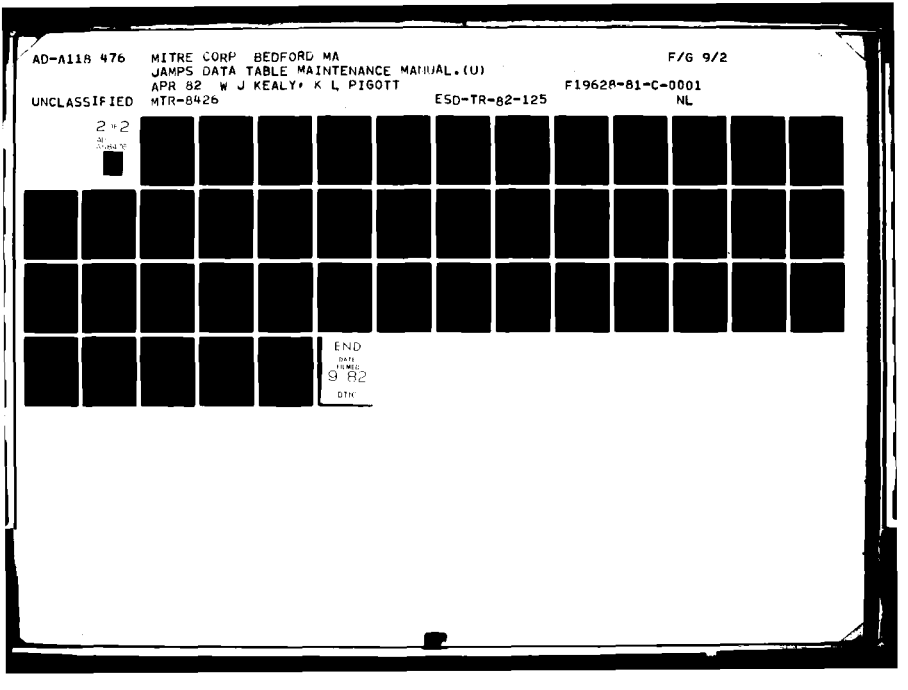
DATA ITEMS AND CODES

DATA ITEM	DUI REF	CHAR CODE	EXPLANATION
COMPLETE COVERAGE	001	COMCOV	TARGET WAS COMPLETELY COVERED. BUT NO REPORTABLE RESULTS.
COMPLETE DESTRUCTION	001	COMDES	COMPLETE DESTRUCTION OF THE TARGET.
NEUTRALIZED	001	NEUT	TARGET WAS NEUTRALIZED.
NO COVERAGE	001	NOCOV	ORDNANCE MISSED THE TARGET.
PARTIAL COVERAGE	001	PARCOV	TARGET WAS PARTIALLY COVERED WITH NO REPORTABLE RESULTS.
PARTIAL DESTRUCTION	001	PARDES	PARTIAL DESTRUCTION OF THE TARGET.
UNKNOWN	001	UNK	UNABLE TO EVALUATE ORDNANCE EFFECTS ON THE TARGET.

DFI NAME AIR MISSION RESULT (REVISED 140CT80) — DFI NO E 471

UNCLASSIFIED

Figure 31. Extract From MED for DFI Codes


AD-A118 476
MITRE CORP BEDFORD MA
JAMPS DATA TABLE MAINTENANCE MANUAL.(U)
F/G 9/2
APR 82 W J KEALY, K L PIGOTT
F19628-81-C-0001
UNCLASSIFIED
MTR-8426
ESD-TR-82-125
NL
2 OF 2
END
DATE FILMED
9 82
DTIC

APPENDIX A

Tutorial: Adding a JINTACCS Message to the JAMPS Data Base

The most complex revision to the JAMPS data base is the addition of a message, therefore, once this skill is mastered, data base revision is mastered. This section describes the step-by-step procedure for entering a message into the data base. The assumption will be made that no parts comprising the message presently exist in the data base. The Approtionment/Allotment (APORALOT) message, A650, will be used in this tutorial.

The first step is to determine the message name, short message name and message number, which can be found in the TIDP. The message number and short message name appear on the first line of the message content. The message name, or message title, follow on the next line. When inserting a message to the data base, the message name should appear on the message menu used in the JAMPS display. Although this is not an advertised part of the data base, it should be kept up-to-date to avoid operator confusion. The Air Operations messages used in the JAMPS message menu are found in the file, **airops.db**. As other categories such as Intelligence, Operations Control, Amphibious, and Fire Support are added to the JAMPS data base, additional message menus will also be added. Pictured in figure A-1 is the **airops.db** file prior to the addition of the APORALOT message. The message number, short message name and long message name should be entered, using the same format as existing entries. This file must have record lengths of 80 characters. The carriage return character must appear in column 80, and no tab characters

```
AIROPS-- 00 +   0042 U N C L A S S I F I E D          231430E AUG 81
     MSG-NO                MESSAGE TITLE
     B704        ABCHANGE          Airbase Change Report
     B705        ACSAMSTAT         Alert Aircraft/SAM Status Report
     E715        AIRDEFWARN        Air Defense Warning Message
     B703        AIREVENT          Significant Air Event Report
     F541        AKNLDG            Acknowledge Message
     F631        ALMSNSCD          Airlift Mission Schedule
     A770        ALORD             Alert Launch Order
     D630        ALREQ             Airlift Request
     E710        ARDEFCOM          Air Defense Command Message
                          .
                          .
                          .
     A691        TECHOPDAT         Technical Operational Data Message
     F755        TRKINTEL          Track Intelligence Message
     F752        TRKMAN            Track Management Message
     F753        TRKREP            Track/Point Report
```

Figure A-1. **airops.db** File in Raw Mode Before Insertion of A650

should appear in the file. Presently this must be accomplished by manually editing the file. At some point, a shell program will be written which will provide automatic reformatting of this table.

Upon entering the message information for the APORALOT message to **airops.db**, the number of records appearing in the header line of the **airops.db** file must be updated to account for the increase in size due to the added record. The updated **airops.db** file appears in figure A-2. Note the carriage return character which appears as a "¿" sign in raw mode in column 80, and the update of number of records in the header line.

Next, the **msgindex** file must be updated to include the APORALOT message. Figure A-3 depicts the **msgindex** file before insertion of the APORALOT message. The information to be entered in **msgindex** is the short message name, the message number and the message index number. There is a long-range plan to incorporate the use of an edition number along with the messages. This will allow different editions of the same message to exist in the data base. The **msgindex** number is the next sequential number in the list. Keeping the same format as existing data in the file, the message number, the short message name, and the **msgindex** number, all separated by commas, are entered into the data base. For trackability, comments may be inserted following the entry. These comments can provide explanations for the entry (e.g., "Revision to the TIDP", or "Developmental Interface Change Proposal (DICP)"). The completed **msgindex** entry is shown in figure A-4.

All the message description files are required to be concatenated into a single file before processing the data base into object files. An entry must be made to the **catmsg** file so that the message

```
AIROPS-- 00 +   0043 U N C L A S S I F I E D          231430E AUG 81
     MSG-NO                MESSAGE TITLE
     B704        ABCHANGE          Airbase Change Report
     B705        ACSAMSTAT         Alert Aircraft/SAM Status Report
     E715        AIRDEFWARN        Air Defense Warning Message
     B703        AIREVENT          Significant Air Event Report
     F541        AKNLDG            Acknowledge Message
     F631        ALMSNSCD          Airlift Mission Schedule
     A770        ALORD             Alert Launch Order
     D630        ALREQ             Airlift Request
     A650        APORALOT          Apportionment/Allotment Message
     E710        ARDEFCOM          Air Defense Command Message
                                   .
                                   .
                                   .
     A691        TECHOPDAT         Technical Operational Data Message
     F755        TRKINTEL          Track Intelligence Message
     F752        TRKMAN            Track Management Message
     F753        TRKREP            Track/Point Report
```

Figure A-2. **airops.db** File in Raw Mode After Insertion of A650

```
.msgindex

; field order is as follows:

;       message number
;       short message name
;       message index number
;       edition number (blank will be interpreted as edition 0)

d669,jsarreq,          1
a653,crossdat,         2
a661,reqstatask,       3
a690,tacopdat,         4
a691,techopdat,        5
a770,alord,            6
b702,jlnchrep,         7
b703,airevent,         8
b704,abchange,         9
b705,acsamstat,       10
b711,engsts,          11
b750,jinflt,          12
c001,msgchangerep,    13
c420,sarsit,          14
c482,sarir,           15
d600,jcasintreq,      16
d660,jairreq,         17
d630,alreq,           18
d667,jrecreq,         19
d666,jescreq,         20
d668,jsupreq,         21
e706,inithand,        22
f625,reqconf,         23
e707,rechand,         24
e710,airdefcom,       25
f632,fltcontinfo,     26
e715,airdefwarn,      27
f541,aknldg,          28
f631,almsnscd,        29
f654,crossconf,       30
f753,trkrep,          31
f754,fltpln,          32
f750,desigarea,       33
f751,ecmdat,          34
a651,employaloc,      35
a652,sortalot,        36
f752,trkman,          37
a635,heloaldat,       38
f636,heloalconf,      39
d665,jairsupreq,      40
f755,trkintel,        41
c460,comspot,         42
```

Figure A-3. **msgindex** Prior to Insertion of A650 Message

```
.msgindex

; field order is as follows:

;       message number
;       short message name
;       message index number
;       edition number (blank will be interpreted as edition 0)

d669,jsarreq,          1
a653,crossdat,         2
a661,reqstatask,       3
a690,tacopdat,         4
a691,techopdat,        5
a770,alord,            6
b702,jlnchrep,         7
b703,airevent,         8
b704,abchange,         9
b705,acsamstat,       10
                .
                .
                .
d665,jairsupreq,      40
f755,trkintel,        41
c460,comspot,         42
a650,aporalot,        43
```

Figure A-4. APORALOT Message in msgindex

description file to be created for the APORALOT message will be concatenated along with the other files. The **catmsg** file appears in figure A-5. Again, this file requires an entry of the short message name, using the format existing in the file. The new **catmsg**, with the APORALOT message entered, is shown in figure A-6.

The next step is to create the message description file for the APORALOT message. The TIDP contains all the information needed to create this file. The name of the file to be created is **aporalot**, the short message name. The first entry is ".msg", followed by the message number. The next entry is ".edition", followed by the edition number, in this case "0". Even in updating a message, the edition number is kept at "0".

Most message description files have comments inserted which outline the format of the table used in part two of the file. It is recommended that comments be used to define the table and provide easier readability.

The first three sets to be entered in the table are common to every message and therefore to every message description file. The FROM, TO, and INFO KDS's are used for the JANAP header information. According to JINTACCS rules, any information which can be transmitted must be framed using JANAP 128 headers and trailers. The initial entry to APORALOT is pictured in figure A-7. The FROM and TO sets are mandatory; TO and INFO sets are repeatable.

Following the JANAP header information, entries to be made to the table are the keyword data sets appearing in the TIDP for the APORALOT message. The KDS name appears in the left-most column of the TIDP under the heading "SET IDENT". The mandatory and repeat-

```
 cat\
jsarreq\
crossdat\
reqstatask\
tacopdat\
techopdat\
alord\
jlnchrep\
airevent\
abchange\
acsamstat\
engsts\
jinflt\
msgchangerep\
sarsit\
sarir\
jcasintreq\
jairreq\
alreq\
> msg1
 cat\
jrecreq\
jescreq\
jsupreq\
inithand\
reqconf\
rechand\
airdefcom\
fltcontinfo\
airdefwarn\
aknldg\
almsnscd\
crossconf\
trkrep\
fltpln\
desigarea\
ecmdat\
employaloc\
sortalot\
trkman\
heloaldat\
heloalconf\
jairsupreq\
comspot\
trkintel\
> msg2
cat msg1 msg2 >msgs
rm msg1 msg2
```

Figure A-5. **catmsg Before Insertion of A650 Message**

```
 cat\
jsarreq\
crossdat\
reqstatask\
    .
    .
    .
jcasintreq\
jairreq\
alreq\
> msg1
 cat\
jrecreq\
jescreq\
jsupreq\
inithand\
reqconf\
rechand\
airdefcom\
fltcontinfo\
airdefwarn\
aknldg\
almsnscd\
crossconf\
trkrep\
fltpln\
desigarea\
ecmdat\
employaloc\
sortalot\
trkman\
heloaldat\
heloalconf\
 jairsupreq\
 comspot\
 trkintel\
 aporalot\
 > msg2
cat msg1 msg2 >msgs
rm msg1 msg2
```

Figure A-6. APORALOT Message Inserted into **catmsg**

```
.msg a650
.edition 0

; field order is as follows:

;       presentation number of keyword data set
;       keyword data set name
;       state number
;       mandatory indicator (m or M, if mandatory)
;       repeatability indicator(r or R, if repeatable)
;       version number (optional, blank will use most recent version)
;       delete indicator (1=delete on input,2=delete on output,3=delete on
;         input and output, blank or 0=used for both input and output)

;  P,       KDS, ST, M, R, V, D        ;
;   ,          ,   ,  ,  ,  ,          ;

   1,      from,  1, m,  ,  ,          ;
   2,        to,  2, m, r,  ,          ;
   3,      info,  3,  , r,  ,          ;
```

Figure A-7. JANAP Sets Entered into aporalot

ability indicators appear on the same line as the KDS name, but under the column headed "CAT". Only mandatory and repeatability conditions are reflected in the KDS table, not conditionality, as represented by a "C" in the TIDP. The version number and delete indicator entries are left blank.

The presentation number is the number which represents the location of the keyword data set in the message; i.e., the third KDS has a presentation number of "3", etc. In general, the presentation number is the same as the state number; however, this need not be the case. This will be discussed in greater detail later in this tutorial. The completed entry for the KDS table for the APORALOT message is pictured in figure A-8.

A state table is then built by combining the information displayed in the keyword data set table, the message content as displayed in table form in the TIDP, and the descriptive text information from the TIDP. The first entry for the state table is the mandatory ".state" token, followed by the body of the state table.

The state table uses the state numbers as established by the keyword data set table, and the sharp sign, "#", which represents the end of the message. The table begins from a state of "0", the start state. From "0", the FROM set, state "1", must be completed since it is mandatory, as reflected in the keyword data set table. Therefore, the first entry to the state table is "0,1".

From state "1", the TO set, state 2, must be completed, again due to the mandatory condition set forth in the keyword data set table. State 2 may be repeated, or state 3, the INFO set, or state 4, the CLASS set may be entered. No matter what path is taken from

```
.msg a650
.edition 0

; field order is as follows:

;       presentation number of keyword data set
;       keyword data set name
;       state number
;       mandatory indicator (m or M, if mandatory)
;       repeatability indicator(r or R, if repeatable)
;       version number (optional, blank will use most recent version)
;       delete indicator (1=delete on input,2=delete on output,3=delete on
;         input and output, blank or 0=used for both input and output)

;  P,       KDS, ST, M, R, V, D        ;
;   ,         ,   ,  ,  ,  ,          ;

    1,      from,  1, m,  ,  ,         ;
    2,        to,  2, m, r,  ,         ;
    3,      info,  3,  , r,  ,         ;
    4,     class,  4, m,  ,  ,         ;
    5,       sic,  5,  ,  ,  ,         ;
    6,      exer,  6,  ,  ,  ,         ;
    7,      oper,  7,  ,  ,  ,         ;
    8,     msgid,  8, m,  ,  ,         ;
    9,       ref,  9,  , r,  ,         ; made repeatible by rev. 1, TIDP
   10,      canx, 10,  ,  ,  ,         ;
   11,     peril, 11, m,  ,  ,         ;
   12,    unable, 12,  ,  ,  ,         ;
   13,  taskunit, 13, m,  ,  ,         ;
   14,     reqno, 14, m,  ,  ,         ;
   15,     almsn, 15, m,  ,  ,         ;
   16,  6liftsod, 16, m,  ,  ,         ;
   17,    aknldg, 17,  ,  ,  ,         ;
   18,  dwngrade, 18,  ,  ,  ,         ;
```

Figure A-8. Completed KDS Table for A650 Message

state 2, state 4 must be entered. That is, state 4 may be entered directly from state 2, or state 3 may be entered, which forces the next entry to be state 4. Therefore, the state tables preserve the mandatory condition by forcing an entry into a particular field defined as mandatory by JINTACCS. The state table is completed in the same way, with the final entry appearing as shown in figure A-9.

Particular attention must be paid to the descriptive text appearing in the TIDP. Note that from state 6, the EXER set, the next entry that can be made is to state 8, the MSGID set. An entry is not allowed to state 7, the OPER set. This condition is not reflected in the keyword data set table. In fact, if looking solely at this table, an entry from state 6 into state 7 looks valid. There is also no obvious definition of this exclusiveness in the table form which appears in the TIDP. However, the EXER set does appear flagged as being conditional in this table. By looking at the descriptive text in the TIDP, as shown in figure A-10, this exclusiveness is definitively established.

Group repeatability is a condition which is not defined in the keyword data set table, but must be established in the state table. The TIDP uses the left square bracket ([) to represent group repeatability. Use of the bracket indicates the beginning and ending KDS's that, as a group, are repeatable.

The APORALOT message does not allow any group repeatability; however, this condition is reflected at its worst in the Airlift Mission Schedule (ALMSNSCD) message, F631. The keyword data set entry for this message appears as shown in figure A-11.

```
.state

 0, 1                          ;from mandatory
 1, 2                          ;to   mandatory
 2, 2, 3, 4                    ;to is also repeatable
 3, 3, 4                       ;info is repeatable
 4, 5, 6, 7, 8                 ;class mandatory
 5, 6, 7, 8
 6, 8                          ;choose either exer or oper
 7, 8,
 8, 9,10,11                    ;msgid is mandatory
 9,9,10,11
10,11                          ;perid mandatory
11,12                          ;apor is mandatory
12,12,13,14,15,16              ;apor is repeatable
13,14,15,16
14,14,15,16                    ;tgtpri is repeatable
15,16
16,17,#                        ;rmks mandatory
17, #                          ;dwngrade optional as result of PTR
```

Figure A-9. State Table for APORALOT Message

MESSAGE NUMBER: All JINTACCS Character-Oriented Messages (U)

TITLE: Standard Introductory Sets [SIS] (U)

SPECIAL INSTRUCTIONS (continued) (U)

(U) Set Identifier: (CLASS) [M] (continued)

b. For NATO messages, use only those data items and codes listed under the heading of "SPECIAL HANDLING" (DFI #E587, DUI 001).

(U) Special Identifier: SIC [C]. The use of this set is mandatory to define the subject matter of the message when attached to NATO command. This set is not used in intra-U.S. messages. Note that there is no field marker following the set identifier. An end of set marker is not used with this set.

Field 1, NASIS Code [M]. Enter the appropriate NASIS code as determined from NATO Supplement 2 to ACP 121. (DFI #E332, DUI 001).

Field 2, NASIS Code [O,k]. The NASIS code field is repeated as necessary to describe the message contents (DFI #E332, DUI 001).

(U) Set Identifier: EXER [C]. This set is conditional upon the message being applicable to an exercise. This set is not used for operational purposes or if an "OPER" set is used.

Field 1, Exercise Nickname [M]. Enter the code name or nickname of the exercise to which the message pertains (DFI #E335, DUI 001).

Field 2, Exercise Message Additional Identifier [O]. Enter an additional exercise nickname or identifier (DFI #E335, DUI 002).

(U) Set Identifier: OPER [C]. This set is conditional upon the message being applicable to an operation that has a code name or nickname. It is not used if an operation name has not been established, and it is not used for exercise purposes or if the set "EXER" is used.

Field 1, Operation Codeword [M]. Enter the assigned operation name or nickname as established by appropriate authority (DFI #E336, DUI 001).

Figure A-10. Extract of Message Instructions from TIDP

```
; field order is as follows:

;       presentation number of keyword data set
;       keyword data set name
;       state number
;       mandatory indicator (m or M, if mandatory)
;       repeatability indicator(r or R, if repeatable)
;       version number (optional, blank will use most recent version)
;       delete indicator (1=delete on input,2=delete on output,3=delete on
;         input and output, blank or 0=used for both input and output)

;  P,       KDS, ST, M, R, V, D        ;
;   ,          ,   ,  ,  ,  ,          ;

   1,      from,  1, m,  ,  ,          ;
   2,        to,  2, m, r,  ,          ;
   3,      info,  3,  , r,  ,          ;
   4,     class,  4, m,  ,  ,          ;
   5,       sic,  5,  ,  ,  ,          ;
   6,      exer,  6,  ,  ,  ,          ;
   7,      oper,  7,  ,  ,  ,          ;
   8,     msgid,  8, m,  ,  ,          ;
   9,       ref,  9,  , r,  ,          ; made repeatible by rev. 1, TIDP
  10,      canx, 10,  ,  ,  ,          ;
  11,     perid, 11, m,  ,  ,          ;
  12,    unable, 12,  ,  ,  ,          ;
  13,  taskunit, 13, m,  ,  ,          ;
  14,     reqno, 14, m,  ,  ,          ;
  15,     almsn, 15, m,  ,  ,          ;
  16,  6liftscd, 16, m,  ,  ,          ;
  17,    aknldg, 17,  ,  ,  ,          ;
  18,       rmks, 18, m,  ,  ,          ;
  19,  dwngrade, 19,  ,  ,  ,          ;
```

Figure A-11. KDS Table Entry for F631, ALMSNSCD, Message

According to the TIDP, the TASKUNIT set through the 6LIFTSCD set can be repeated. Nested within that repeatability, the REQNO set through 6LIFTSCD may be repeated, and nested within that, ALMSN through 6LIFTSCD may be repeated. This is represented in the state table pictured in figure A-12. The mandatory conditions force continuation downward through the states to the 6LIFTSCD set, state 16. From state 16, the TASKUNIT group, the REQNO group, or the ALMSN group may be repeated, the AKNLDG set or the DWNGRADE set may be entered, or the message may be terminated.

The final section of the message description file for the APORALOT message is the table used for prepopulation. In order to make an entry to this section, it is necessary to decide which field in a KDS is to be prepopulated. For example, the classification field of the CLASS KDS is prepopulated, as is the message type field of the MSGID KDS in all messages. The APORALOT message which will be displayed by the JAMPS system will be prepopulated with a classification of SECRET which is changeable, and a message type of APORALOT which will remain fixed. The **aporalot** file is now complete and appears as shown in figure A-13.

As mentioned above, the presentation number and the state number of a keyword data set need not be the same. In adding a keyword data set to the message, the presentation numbers of any KDS following the entry location of the new KDS must be changed accordingly. It is possible to avoid changing all of the state numbers by letting the state number of the new KDS be the next sequential number in the list.

Assume, for example, that during testing it was discovered that the CANX set had been inadvertently omitted from the APORALOT message. The incorrect entry of A650 is pictured in figure A-14. The

```
.state

 0, 1                          ;from mandatory
 1, 2                          ;to   mandatory
 2, 2, 3, 4                    ;to is also repeatable
 3, 3, 4                       ;info is repeatable
 4, 5, 6, 7, 8                 ;class mandatory
 5, 6, 7, 8
 6, 8                          ;choose either exer or oper
 7, 8
 8, 9,10,11                    ;msgid is mandatory
 9,9,10,11
10,11                          ;perid is mandatory
11,12,13
12,13                          ;taskunit is mandatory
13,14                          ;reqno is mandatory taskunit starts rptbl group
14,15                          ;almsn is mandatory reqno starts nstd group
15,16                          ;6liftscd is mandatory almsn starts 2nd nest
16,13,14,15,17,18              ;6liftscd ends all rptable groups
17,18                          ;rmks is mandatory
18,19, #                       ;dwngrade is optional as result of PTR
19,#
```

Figure A-12. State Table for F631

```
.msg a650
.edition 0

; field order is as follows:

;       presentation number of keyword data set
;       keyword data set name
;       state number
;       mandatory indicator (m or M, if mandatory)
;       repeatability indicator(r or R, if repeatable)
;       version number (optional, blank will use most recent version)
;       delete indicator (1=delete on input,2=delete on output,3=delete on
;         input and output, blank or 0=used for both input and output)

;  P,       KDS, ST, M, R, V, D        ;
;   ,          ,  ,  ,  ,  ,          ;

   1,      from,  1, m,  ,  ,          ;
   2,        to,  2, m, r,  ,          ;
   3,      info,  3,  , r,  ,          ;
   4,     class,  4, m,  ,  ,          ;
   5,       sic,  5,  ,  ,  ,          ;
   6,      exer,  6,  ,  ,  ,          ;
   7,      oper,  7,  ,  ,  ,          ;
   8,     msgid,  8, m,  ,  ,          ;
   9,       ref,  9,  , r,  ,          ; made repeatible by rev. 1, TIDP
  10,      canx, 17,  ,  ,  ,          ;
  11,     perid, 10, m,  ,  ,          ;
  12,     appor, 11, m, r,  ,          ;
  13,     6alot, 12,  ,  ,  ,          ;
  14,    tgtpri, 13,  , r,  ,          ;
  15,    aknldg, 14,  ,  ,  ,          ;
  16,       rmks, 15, m,  ,  ,          ;
  17,  dwngrade, 16,  ,  ,  ,          ;

  .state

  0, 1                          ;from mandatory
  1, 2                          ;to   mandatory
  2, 2, 3, 4                    ;to is also repeatable
  3, 3, 4                       ;info is repeatable
  4, 5, 6, 7, 8                 ;class mandatory
  5, 6, 7, 8
  6, 8                          ;choose either exer or oper
  7, 8,
  8, 9,10,17                    ;msgid is mandatory
  9,9,10,17
 10,11                          ;perid mandatory
 11,11,12,13,14,15              ;apor is mandatory
 12,13,14,15                    ;apor is repeatable
 13,13,14,15
 14,15                          ;tgtpri is repeatable
 15,16,#
 16,#                           ;rmks mandatory
 17,10                          ;dwngrade optional as result of PTR

;  following area is optional and used for mandatory entries in
;    certain fields of keyword data sets

;  field order is as follows
;    presentation number for applicable keyword data set
;    relative field number to which entry applies
;    entry to be inserted (will be used as a character string)
;    protection code for data (p protected, u or blank unprotected)

  .man
  4,1,s e c r e t,u   ;class
  8,1,aporalot,p ;msgid
```

Figure A-13. Complete A650 Entry to Data Base

```
.msg a650
.edition 0

; field order is as follows:

;       presentation number of keyword data set
;       keyword data set name
;       state number
;       mandatory indicator (m or M, if mandatory)
;       repeatability indicator(r or R, if repeatable)
;       version number (optional, blank will use most recent version)
;       delete indicator (1=delete on input,2=delete on output,3=delete on
;         input and output, blank or 0=used for both input and output)

;  P,       KDS, ST, M, R, V, D        ;
;   ,          ,   ,  ,  ,  ,          ;

    1,      from,  1, m,  ,  ,          ;
    2,        to,  2, m, r,  ,          ;
    3,      info,  3,  , r,  ,          ;
    4,     class,  4, m,  ,  ,          ;
    5,       sic,  5,  ,  ,  ,          ;
    6,      exer,  6,  ,  ,  ,          ;
    7,      oper,  7,  ,  ,  ,          ;
    8,     msgid,  8, m,  ,  ,          ;
    9,       ref,  9,  , r,  ,          ; made repeatible by rev. 1, TIDP
   10,     perid, 10, m,  ,  ,          ;
   11,     apor, 11, m, r,  ,           ;
   12,     6alot, 12,  ,  ,  ,          ;
   13,    tgtpri, 13,  , r,  ,          ;
   14,    aknldg, 14,  ,  ,  ,          ;
   15,  dwngrade, 15,  ,  ,  ,          ;

    .state

    0, 1                        ;from mandatory
    1, 2                        ;to   mandatory
    2, 2, 3, 4                  ;to is also repeatable
    3, 3, 4                     ;info is repeatable
    4, 5, 6, 7, 8               ;class mandatory
    5, 6, 7, 8
    6, 8                        ;choose either exer or oper
    7, 8,
    8, 9,10                     ;msgid is mandatory
    9,9,10
   10,11                        ;perid mandatory
   11,11,12                     ;apor is mandatory
   12,13,14,15,#                ;apor is repeatable
   13,13,14,15,#
   14,15,#                      ;tgtpri is repeatable
   15,#
```

Figure A-14. A650 Message with CANX Set Omitted

CANX set now has to be added to the message. The keyword data set table has to be rewritten from the PERID set downward. Although this is not difficult, it would be if this had been a large message with forty to fifty keyword data sets. In addition to rewriting the keyword data set table, the state table would also have to be rewritten. A shortcut can be taken by adding the CANX set, renumbering the presentation numbers and assigning a state number of 16 to the CANX set. The new entry of A650 with the CANX set is shown in figure A-15.

The next step is to enter the keyword data sets in **kdsindex**. Since the standard introductory sets and the JANAP KDS's appear in every message, there is no need to enter them into the data base. However, it is advisable to verify that they do appear. It will be necessary to enter the CANX, PERID, APPOR, 6ALOT, TGTPRI, AKNLDG, and DWNGRADE sets in **kdsindex**.

Most of these KDS's appear as KDS's in other messages, in fact, they may already exist in the data base. For this tutorial, however, the assumption is made that they do not. The keyword data set name is inserted in **kdsindex**, preserving the alphabetical order, followed by the KDS index number. The next sequential number is chosen for the 6ALOT KDS, and so on through the list of new KDS's. The completed **kdsindex** entry is shown in figure A-16.

There is no KDS with an index number of 12 in the **kdsindex** file as it now appears, since it has been deleted. Deletion of the KDS did not cause a reordering of the KDS index numbers. Also, while an alphabetical order makes it easy to determine whether or not a KDS is in the **kdsindex**, it makes it very difficult to determine the next

```
.msg a650
.edition 0

; field order is as follows:

;       presentation number of keyword data set
;       keyword data set name
;       state number
;       mandatory indicator (m or M, if mandatory)
;       repeatability indicator(r or R, if repeatable)
;       version number (optional, blank will use most recent version)
;       delete indicator (1=delete on input,2=delete on output,3=delete on
;         input and output, blank or 0=used for both input and output)

;  P,       KDS, ST, M, R, V, D        ;
;   ,          ,   ,  ,  ,  ,         ;

   1,      from,  1, m,  ,  ,         ;
   2,        to,  2, m, r,  ,         ;
   3,      info,  3,  , r,  ,         ;
   4,     class,  4, m,  ,  ,         ;
   5,       sic,  5,  ,  ,  ,         ;
   6,      exer,  6,  ,  ,  ,         ;
   7,      oper,  7,  ,  ,  ,         ;
   8,     msgid,  8, m,  ,  ,         ;
   9,       ref,  9,  , r,  ,         ; made repeatible by rev. 1, TIDP
  10,      canx, 16,  ,  ,  ,         ;
  11,     perid, 10, m,  ,  ,         ;
  12,     appor, 11, m, r,  ,         ;
  13,     6alot, 12,  ,  ,  ,         ;
  14,    tgtpri, 13,  , r,  ,         ;
  15,    aknldg, 14,  ,  ,  ,         ;
  16,  dwngrade, 15,  ,  ,  ,         ;

  .state

  0, 1                          ;from mandatory
  1, 2                          ;to   mandatory
  2, 2, 3, 4                    ;to is also repeatable
  3, 3, 4                       ;info is repeatable
  4, 5, 6, 7, 8                 ;class mandatory
  5, 6, 7, 8
  6, 8                          ;choose either exer or oper
  7, 8,
  8, 9,10                       ;msgid is mandatory
  9,9,10,16
 10,11                          ;perid mandatory
 11,11,12                       ;apor is mandatory
 12,13,14,15,#                  ;apor is repeatable
 13,13,14,15,#
 14,15,#                        ;tgtpri is repeatable
 15,#
 16,10                          ;canx set added
```

Figure A-15. APORALOT Entry with CANX Set Added

```
.kdsindex

;  field order is as follows:

;       keyword data set name
;       keyword data set number
;       version number ( blank is interpreted as zero (0))

FROM,           1
TO,             2
INFO,           3
AMPN,           4
NARR,           5
RMKS,           6
CLASS,          7
CHANGE,       192
CANCEL,         8
DELETE,         9
AFTER,         10
ADD,           11
6ALOT,        193
6APPDIS,       13
      .
      .
      .
AIRBASE,       66
AIRDROP,       67
AIRSPACE,      68
AIRTKS,        69
AKNLDG,       194
ALERT,         70
ALMSN,         71
ALT,           72
APPOR,        195
APPROVE,       73
ARDAT,         74
ARDEPCOM,      75
      .
      .
      .
BASESTAT,      81
BCNREF,        82
BEACON,        93
CANX,         196
CAP,           84
CATCON,        85
CIRCLE,        86
      .
      .
      .
DUALMSG,       99
DUPTRK,       100
DUTIES,       101
DWNGRADE,     197
ECMDAT,       102
ECMEFF,       103
EFFECTIV,     104
      .
      .
      .
OPER,         141
ORDAVAIL,     142
PEREFF,       143
PERID,        198
PIREP,        144
PROTFREQ,     145
RCTCOMP,      146
      .
      .
      .
TGTCOD,       179
TGTDAT,       180
TGTDES,       181
TGTPRI,       199
TIMEAMP,      182
UNABLE,       183
      .
      .
      .
LOCN,         189
RELOC,        190
TKAMP,        191
```

Figure A-16. kdsindex with A650 Sets Added

number that should be used as the index number. However, alphabetical order is not a system requirement, but a personal preference. Sequential ordering of the KDS index numbers is also a personal preference; the only requirement is that each KDS have a unique KDS index number.

The next entry is to kdss, and all the information needed is found in the Catalog of Keyword Data Sets. Entries need not be made for the standard introductory sets since they should already exist in kdss; verifying that they exist is sufficient. Entries are made in alphabetical order.

The first set to be entered is the 6ALOT set. The first entry is ".KDS 6ALOT", followed by the KDS type. The set, 6ALOT, is a columnar set, as determined by the numeric as the first character in the set name. Therefore, the next entry would be ".COLUMNAR", followed by ".VERSION 0". The field table can now be entered.

The first entry for the field table preceding the actual tabular information is ".FIELD". The DFI/DUI's are then listed with the information as described in paragraph 2.2.2. Since 6ALOT is a columnar set, the print positions must be entered for the fields. The print positions can be determined by looking at the SET MAP in the KDS catalog. A warning should be given at this point that the Catalog of Keyword Data Sets has been known to contain errors in the SET MAP. This can be checked by doing a field-by-field verification by referring to the MED, or it can later be verified when a cross check is done against the data base. Using the information appearing in this table, the state table can be built. The entry to the data base for the 6ASLOT set appears in figure A-17.

```
.KDS 6ALOT
.COLUMNAR
.VERSION 0

.FIELDS
; FIELD ORDER IS AS FOLLOWS:
;       FIELD NUMBER
;       DFI NUMBER
;       DUI NUMBER
;       STATE NUMBER
;       PRINT POSITION FOR FIELD HEADER (ONLY USED FOR TYPE COLUMNAR)
;       MANDATORY INDICATOR (M IF MANDATORY)
;       REPEATABILITY INDICATOR (R IF REPEATABLE)
;       FIELD DESCRIPTOR INDICATOR (1=REQUIRED, 0 OR BLANK=NOT USED)
;       DELETE INDICATOR (1=DELETE ON INPUT,2=DELETE ON OUTPUT,3=DELETE ON
;         INPUT AND OUTPUT, BLANK OR 0=USED FOR BOTH INPUT AND OUTPUT)

;  P,   DFI, DUI, ST, PR, M, R, F, D        ;
;   ,     ,    ,   ,   ,  ,  ,  ,          ;

   1, E0987, 031,  1,  2, M,  ,  ,          ;
   2, E0987, 032,  2, 16, M,  ,  ,          ;
   3, E0107, 004,  3, 30, M,  ,  ,          ;
   4, E0031, 022,  4, 35, M,  ,  ,          ;
   5, E0513, 001,  5, 40,  ,  ,  ,          ;
   6, E0987, 023,  6, 47,  ,  ,  ,          ;
   7, E0527, 001,  7, 61,  ,  ,  ,          ;

  .STATE

  0, 1
  1, 2
  2, 3
  3, 4
  4, 5, 6, 7, #
  5, 6, 7, #
  6, 7, #
  7, #
```

Figure A-17. Entry for 6ALOT KDS to kdss

Entries are now made to **kdss** for the rest of the KDS's in APORALOT, as shown in figure A-18. In the state table for the APPOR KDS, states 3, 4 and 5 are repeatable. However, in the state table, unlike a message state table, state 3 can only go to states 4 or 5, not to itself. Unlike the repeatability of KDS's within a message, the repeatability of fields within a KDS is treated as a group repeatability. Also, according to JINTACCS formatting, if field "n" is repeatable, all fields following field "n" are also repeatable. This is further defined as all fields following field "n" are treated as nested repeatable groups. Therefore, if there are eight fields within a KDS and field 5 is repeatable, fields 5, 6, 7, and 8 are repeatable as a group, fields 6, 7 and 8 are repeatable by themselves.

In building the state table for fields within the KDS, care must be exercised to avoid confusing the state numbers with the field numbers. For example, in the AEW KDS, as shown in figure A-19, states 3 and 4 are alternate contents fields for field 3, states 5 and 6 are alternate contents fields for field 4, and states 7, 8, and 9 are alternate contents fields for field 5. In the AEW state table, state 2 can go to state 3 or state 4. Although state 3 is mandatory, it is only one of the possible alternative fields that may be entered.

Upon completion of the entry to the **kdss** file, the next entry is to be file **dfidui**. Again it is assumed that no entries for this message have been previously entered to this file except for the fields comprising the standard introductory sets. To make entries to **dfidui**, the MED is the necessary documentation.

```
.KDS  ANALOG
.LINEAR
.VERSION 0

.FIELDS
 1, BB319, 003,  1,   . M,  .  .      ;


.STATE

0,  1,
1,  0,


.KDS  APROR
.LINEAR
.VERSION 0

.FIELDS
 1, BB987, 018,  1,   . M,  .  .      ;
 2, EB922, 001,  2,   . M,  .  .      ;
 3, BB107, 004,  3,   . M, R,  .      ;
 4, BB927, 016,  4,   . M, R,  .      ;
 4, BB931, 017,  5,   . M, R,  .      ;


.STATE

0,  1,
1,  2,
2,  3,
3,  4,  5,
4,  3,  4,  5,  0,
5,  3,  4,  5,  0,

.KDS  CANX
.LINEAR
.VERSION 0

.FIELDS
 1, BB050, 001,  1,   . M,  .  .      ;
 2, BB146, 001,  2,   . M,  .  .      ;
 3, CB143, 005,  3,   . M,  .  .      ;
 4, BB580, 001,  4,   . M,  .  .      ;
 5, BB147, 001,  5,   .  .  .  .      ;
 6, BB122, 002,  6,   .  .  .  .      ;
 7, EB042, 001,  7,   .  .  .  .      ;

.STATE

0,  1,
1,  2,
2,  1,
3,  4
4,  5, 6, 7, 0
5,  6, 7, 0
6,  7, 0
7,  0

.KDS  DNGRADE
.LINEAR
.VERSION 0

.FIELDS
 1, BB679, 001,  1,   . M,  .  .

.STATE

0,  1,
1,  0,


.KDS  PERI
.LINEAR
.VERSION 0

.FIELDS
 1, CB143, 006,  1,   . M,  .  .      ;
 2, CB143, 007,  2,   . M,  .  .      ;


.STATE

0,  1,
1,  2,
2,  0,


.KDS  TOTPRI
.LINEAR
.VERSION 0

.FIELDS
 1, BB147, 004,  1,   . M,  . 1,      ;
 2, BB183, 001,  2,   . M, R,  .      ;
 2, BB819, 001,  3,   . M, R,  .      ;
 2, BB988, 002,  4,   . M, R,  .      ;
 2, BB939, 001,  5,   . M, R,  .      ;


.STATE

0,  1,
1,  2,  3,  4,  5,
2,  2,  3,  4,  5,  0,
3,  2,  3,  4,  5,  0,
4,  2,  3,  4,  5,  0,
5,  2,  3,  4,  5,  0,
```

Figure A-18. APORALOT KDS's to be Entered into kdss

```
.KDS  AEW
.LINEAR
.VERSION 0

.FIELDS
 1, E0500, 015,  1,   , M,  ,  ,      ;
 2, E0491, 006,  2,   ,  ,  , 1,      ;
 3, E0651, 002,  3,   , M,  ,  ,      ;
 3, E0681, 002,  4,   , M,  ,  ,      ;
 4, E0651, 003,  5,   , M,  ,  ,      ;
 4, E0681, 003,  6,   , M,  ,  ,      ;
 5, C0469, 008,  7,   , M, R,  ,      ;
 5, C0497, 005,  8,   , M, R,  ,      ;
 5, C0498, 002,  9,   , M, R,  ,      ;


.STATE

0,  1,
1,  2,  3,  4,
2,  3,  4,
3,  5,  6,
4,  5,  6,
5,  7,  8,  9,
6,  7,  8,  9,
7,  7,  8,  9,  #,
8,  7,  8,  9,  #,
9,  7,  8,  9,  #,
```

Figure A-19. AEW Keyword Data Set

Excluding the standard introductory sets, the seven remaining KDS's, CANX, AKNLDG, APPOR, CANX, DWNGRADE, PERID and TGTPRI, break down into 28 DFI/DUI's (26 unique DFI/DUI's). The following DFI/DUI's must be entered into **dfidui**: E0019 001, E0027 016, E0031 017, E0031 022, E0050 001, E0107 004, C0143 005, C0143 006, C0143 007, E0146 001, E0147 001, C0183 001, E0319 003, E0332 002, E0513 001, E0527 001, E0580 001, E0679 001, E0908 002, E0922 002, E0939 001, E0987 018, E0987 023, E0987 031, E0987 032 and E1042 001.

In making an initial entry of a DFI into **dfidui**, the MED must be consulted to determine the DFI type: "com", "dif" or "chn". All the DUI's associated with a DFI must be compared against each other to see if there are any variances of minimum or maximum field size, left/right justification, or in message map characters.

The first entry to be made to **dfidui** is for E0019 001. By consulting the MED, it is determined that all DUI's have the same field size, message map characters, and justification. Therefore, we have a "com"-type DFI. The entry for a "com"-type DFI is ".com", followed by "e" or "c" and the DFI number, left/right justification flag, fixed/variable length flag, minimum field length, maximum field length and the message map codes. Therefore, the first entry is:

```
.com e0019,l,v,2,6,a
```

The next entry is for E0027 016. This is a "dif"-type DFI, since DUI 001 through 004 are codes of one to three numerics, and DUI 007 and 016 are codes of one or two numerics. The entry for a "dif"-type DFI is ".dif", followed by "e" or "c" and the DFI number. This is followed by ".dui", the DUI number, the left/right justification flag, fixed/variable length flag, minimum field length,

maximum field length, and the message map characters, separated by commas. The entry for E0027 016 is the following:

```
.dif e0027
   .dui 016,r,v,1,2,n
```

The third possible type of entry to make to **dfidui** is for a "chn"-type DFI. The entry for C0143 006 is this type. The entries for C0143 006 and C0143 007 are identical to C0143 005. There is no differentiation between DUI's in "chn"-type DFI's. The entry for C0143 005 is in the same format as the ".com", except that it is followed by a list of the component parts of the chain by DFI/DUI. The entry for C0143 005 is the following:

```
.chn c0143,1,f,,7,nnnnna
   .chain
      e0023,001
      e0024,001
      e0025,001
      e0026,001
```

The remaining entries to **dfidui** are shown in figure A-20. The entry for C0183 is not shown. This field, as it appears in the MED, is contrary to JINTACCS rules. To avoid confusion, it was omitted from this tutorial as a future revision to the MED will correct this error. Note that in making the entries of E0031 017 and E0031 022, the entries were simply added to the list already existing in the data base. Also note that the entry of C0143, a chain-type DFI, requires the entry of all the component parts of the chain, thus, E0023 001, E0024 001, E0025 001 and E0026 001 have been added.

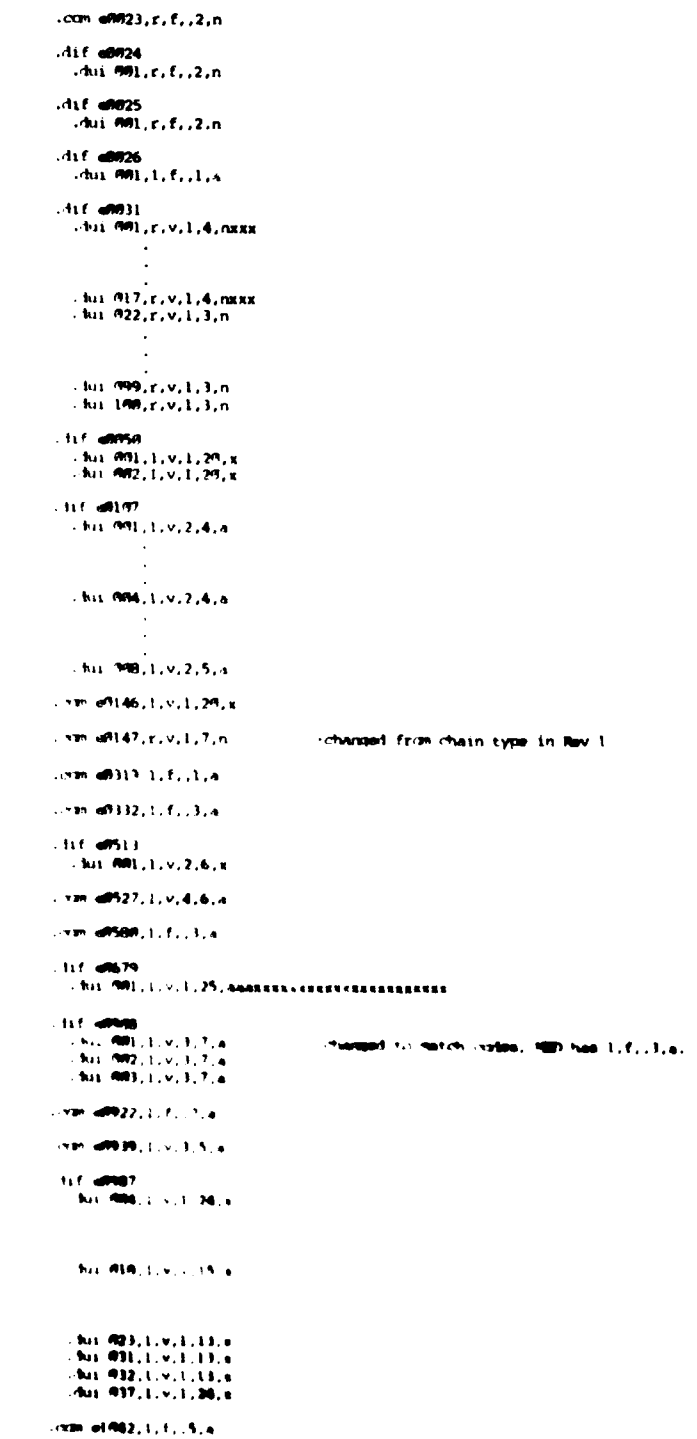


Figure A-20. Entry of APOBALOT DFI's to dfidat

The next entry for these fields is to the **fdfh** file. This file consists of the entry of the DFI and DUI numbers, followed by the field descriptor and field header. The information to be entered can be gathered from the MED and/or the Catalog of Keyword Data Sets. The field descriptor and field header follow one right under the other in the MED under the column "FLD DESC/COL HEADER". The field descriptors and field headers may also be determined from the Catalog of Keyword Data Sets by seeing if they are displayed in the SET MAP area. An entry must be made to this table only if the field can appear by itself; i.e., it can exist outside a chain. The entries to be made to **fdfh** for the APORALOT message appear in figure A-21. A field descriptor must be entered in **dfidui** for each DFI/DUI. A field header need only be entered if the DFI/DUI appears as part of a columnar keyword data set. If there is no entry for field descriptor or field header in the MED, an entry which appropriately captured the essence of the field was made up and entered.

The next entries are to the **codes**, **helptext**, and **itemcodes** files. Since these are almost direct copies from the MED, there is no need to go into great detail.

Entries are made to **codes** and **itemcodes** by entering an "E" or a "C", followed by the DFI number and the DUI number. The DFI number is expected to be a four-digit number, but the first leading zero may be entered as a blank. The DUI number follows immediately, with no separation. The entry for a "C"-type DFI is simply the preceding: the DFI and DUI numbers and DUI name. No additional text follows this entry for a "C"-type DFI. An "E"-type DFI is followed by copying what appears in the DATA ITEM column of the MED, and the entry in the CHAR CODE section of the MED, delimited by asterisks.

```
e0019, 001,tgttyp,tgttyp
e0023, 002,cdday,cdday
e0027, 016,asgn,asgn
e0031, 017,asgn,asgn
e0031, 022,noac,noac
e0050, 001,msg-title                ;ambiguous spec of fd or fh
e0107, 004,amsn,amsn
c0143, 005,date-time-ref            ;no field descriptor in MED
c0143, 006,from,from
c0143, 007,to,to
e0146, 001,msg-orig                 ;no field descriptor in MED
e0147, 001,msg-ser-num              ;no field descriptor in MED
e0319, 003,aknreq,aknreq
e0332, 002,sic                      ;no field descriptor in MED
e0513, 001,actyp,actyp
e0527, 001,icao,icao
e0580, 001,mnthrm                   ;ambiguous spec of fd or fh
e0679, 001,dwngrade                 ;amibguous spec of fd or fh
e0908, 002,gentyp,gentyp
e0922, 001,per-or-num               ;no field descriptor in MED
e0939, 001,msn,msn
e0987, 018,cmpcmd,cmpcmd
e0987, 023,acuntid,acuntid
e0987, 031,lose,lose
e0987, 032,gain,gain
e1042, 001,spec-not                 ;no field descriptor in MED
```

Figure A-21. Entries to **fdfh**

Figure A-22 shows part of the entry for the APORALOT message to the codes file. The identical entry could be made to the itemcodes file, however, some changes may be desired for esthetic purposes. For example, in the entry for E0050 001, the code used for Helicopter Daily Mission Summary is HELODLYMSNSUM. This code spans two lines, and will be concatenated in the processing of the codes file. The codes file is used for internal purposes only; the operator will never see it. The itemcodes file, on the other hand, is displayed to the operator. Therefore, the entry for HELODLYMSNSUM code and similar codes may be displayed as shown in figure A-23, rather than using the same convention as is the codes file.

Figure A-24 shows parts of an entry of E0515 001 to the codes file and the concatenation symbols used. This entry contains a number of examples where codes span more than one line. The sharp sign, "#", is used in processing this file to show that a space should be left between the two sections of the codes being concatenated. The equal sign, "=", is used to represent a hyphen which will be deleted by the processor upon concatenation of the two sections. If there is no symbol between the two sections, upon processing, a simple concatenation of the two sections will take place. These concatenation symbols do not appear in the itemcodes file.

Entries to the helptext file are made as specified in paragraph 2.4.1. A portion of the entry for the APORALOT message is shown in figure A-25. The definition is taken from the MED entry for the DUI explanation. When necessary, the explanation should be rewritten to provide a clear, concise and easily understood definition of the DUI. The instruction portion should provide information to the operator on how information should be entered into a field. For a "C"-type DFI, information for entering each "E"-type DFI which comprises the chain must be embedded in the instructions for the chain.

```
E 019001        TARGET TYPE
*     ASSEMBLY AREAS              * ASSY          *
*     BRIDGE                      * BRIDGE        *
*     BUILDING                    * BLDG          *
*     COMMAND CENTER              * CMDCTR        *
*     COMMUNICATIONS/             * COMELT        *
*       ELECTRONICS               *               *
*     DAM                         * DAM           *
*     DROP ZONE                   * DROPZN        *
*     FORTIFICATION/STRUCTURE     * TFORT         *
                      .
                      .
                      .
*     ANTIRADIATION MISSILE       * ARM           *
*       (ARM)                     *               *
*     ESM AIRCRAFT PERFORMING     * ESMAIR        *
*       ECM DECEPTION             *               *
*     ESM SURFACE JAMMER          * ESMSUR        *
*     ECM SURFACE                 * ECMSUR        *
*     ECM AIRCRAFT JAMMER         * ECMAIR        *
E 023001        DAY
*       (01                       * (01           *
*       THRU                      * THRU          *
*       31)                       * 31)           *
E 026001        TIME ZONE
*     GMT PLUS 1 HOUR             * A             *
*     GMT PLUS 2 HOURS            * B             *
*     GMT PLUS 3 HOURS            * C             *
                      .
                      .
                      .
*     GMT MINUS 3 HOURS           * P             *
*     GMT MINUS 2 HOURS           * O             *
*     GMT MINUS 1 HOUR            * N             *
*     GREENWICH MEAN TIME         * Z             *
*       (GMT)                     *               *
E 050001        MESSAGE TITLE
*     ACKNOWLEDGE MESSAGE         * AKNLDG        *
*     AIR DEFENSE COMMAND         * AIRDEFCOM     *
*       MESSAGE                   *               *
*     AIR DEFENSE WARNING         * AIRDEFWARN    *
*       MESSAGE                   *               *
                      .
                      .
                      .
*     HELICOPTER AVAILABILITY     * HELOAVIL      *
*     HELICOPTER DAILY MISSION    * HELODLYMSN    *
*       SUMMARY                   * SUM           *
                      .
                      .
                      .
*     WARNING ORDER               * WARNORD       *
*     WEATHER FORECAST            * WXFCST        *
*     WEATHER REPORT              * WXRPT         *
C 143006        FROM
C 143007        TO
```

Figure A-22. Partial Entry of DFI's from APORALOT to codes

```
E 050001        MESSAGE TITLE
*     ACKNOWLEDGE MESSAGE          * AKNLDG          *
*     AIR DEFENSE COMMAND          * AIRDEFCOM       *
*       MESSAGE                    *                 *
*     AIR DEFENSE WARNING          * AIRDEFWARN      *
*       MESSAGE                    *                 *
*     AIR EMPLOYMENT/ALLOCA-       * EMPLOYALOC      *
*       TION PLAN                  *                 *
                    .
                    .
                    .
*     HELICOPTER DAILY MISSION     * HELODLYMSNSUM   *
*       SUMMARY                    *                 *
                    .
                    .
                    .
```

Figure A-23. Change in HELODLYMSNSUM Code for itemcode

```
E 515001      POINT TYPE
*    ECM FIX                      * ECM FIX        *
*    HAZARD                       * HAZARD         *
*    HAZARD, NAV                  * NAV            *
*    HAZARD, MINE                 * MINE           *
*    HAZARD, IMPACT POINT         * IMPACT POINT   *
*    HAZARD, GROUND ZERO          * GROUND ZERO    *
*    HAZARD, AIR/WEAPON ENTRY     * AIR WEAPON#    *
*      POINT                      * ENTRY POINT    *
*    HAZARD, MISSILE LAUNCH       * LAUNCH POINT   *
*      POINT                      *                *
*    HAZARD, ECM DECOY            * ECM DECOY      *
*    GENERAL REFERENCE POINT      * GENERAL REF=   *
*                                 * ERENCE POINT   *
*    MARSHAL POINT                * MARSHAL#       *
*                                 * POINT          *
*    WAY POINT                    * WAY POINT      *
                    .
                    .
                    .
*    TOC                          * TOC            *
*    TDS                          * TDS            *
*    STRIKE HOSTILE               * STRIKE HOS=    *
*                                 * TILE           *
*    HOSTILE TROOP CONCEN-        * TROOP CON=     *
*      TRATION                    * CENTRATION     *
*    HOSTILE AIRBASE              * HOSTILE#       *
*                                 * AIRBASE        *
*    HOSTILE SAM SITE             * HOSTILE SAM#   *
*                                 * SITE           *
*    HOSTILE ARTILLERY            * HOSTILE#       *
*                                 * ARTILLERY      *
*    HOSTILE CONVOY               * HOSTILE#       *
*                                 * CONVOY         *
*    HOSTILE RAIL                 * HOSTILE RAIL   *
*    HOSTILE BRIDGE               * HOSTILE#       *
*                                 * BRIDGE         *
```

Figure A-24. E0515 001 Entry to codes Showing Concatenation Symbols

```
.DFI
E 019001
TARGET TYPE

DEFINITION: A DESCRIPTOR USED TO CLASSIFY TARGETS INTO ONE OF SEVERAL
   CLASSES, CATEGORIES, OR GENERIC GROUPINGS.

ENTER ONE OF THE CODES.

.DFI
E 023001
DAY

DEFINITION: ONE OF THE 24 HOUR PERIODS OF A MONTH AS DEFINED BY THE GREGORIAN
   CALENDAR.

ENTER THE DAY OF THE MONTH.

.DFI
E 326001
TIME ZONE

DEFINITION: ONE OF THE 24 TIME ZONES INTO WHICH THE WORLD HAS BEEN DIVIDED
   FOR ESTABLISHING TIME RELATIONSHIPS BETWEEN GEOGRAPHICAL AREAS, BASED
   UPON THE LOCAL STANDARD TIME, GREENWICH, ENGLAND (GMT).

ENTER ONE OF THE CODES.

.DFI
N 031022
NUMBER OF AIRCRAFT

DEFINITION:

ENTER THE NUMBER OF AIRCRAFT.

.DFI
E 350001
MESSAGE TITLE

DEFINITION: THE TITLE OF A MESSAGE/REPORT.

ENTER ONE OF THE CODES.

.DFI
C 143006
FROM

DEFINITION: BEGINNING OF RELEVANT PERIOD.

THIS IS A COMPLETE TIME ENTRY:

     281555T

ENTER THE FOLLOWING SEQUENCE OF CHARACTERS AND NUMBERS:
   1.  DAY OF THE MONTH     (01-31)
   2.  HOUR OF THE DAY      (00-24)
   3.  MINUTE OF THE HOUR   (00-60)
   4.  TIME ZONE            (A-Z OMITTING J)
.DFI
C 143007
TO

DEFINITION: END OF RELEVANT PERIOD.

THIS IS A COMPLETE TIME ENTRY:

     281555T

ENTER THE FOLLOWING SEQUENCE OF CHARACTERS AND NUMBERS:
   1.  DAY OF THE MONTH     (01-31)
   2.  HOUR OF THE DAY      (00-24)
   3.  MINUTE OF THE HOUR   (00-60)
   4.  TIME ZONE            (A-Z OMITTING J)
```

Figure A-25. Entries to helptext File for A650 Message

All actions have now been completed for entering the APORALOT message into the data base. At this point, the cross check program should be run to ensure that nothing has been overlooked and to verify the accuracy of the entries.

GLOSSARY

Terms

Data Field Identifier	A JINTACCS term for what is commonly called a data element. A data element consists of a definition of the concept, a set of data items each identifying a possible status, and the data codes used to represent the data item in JINTACCS messages.
Data Item	A data item identifies a discrete possible status (value) of a DFI (concept).
Data Item Code	A data item code is a set of one or more alphabetic, numeric and/or special symbols taking the form of a word or words, a code, an alphanumeric serial number, etc. A character code may or may not be identical to its data item.
Data Use Identifier	A data use identifier is a subcategory of a data field identifier which specifies a particular usage for the concept defined for the data field identifier.

Field Descriptor	Field descriptors are words, abbreviations or acronyms which describe or amplify the data in formatted fields of linear data sets.
Field Header	Field headers are words, abbreviations or acronyms which describe or amplify the data in the columns of a columnar keyword data set.
Keyword Data Set	A keyword data set is an ordered collection of data presented in either a horizontal form or columnar form. See linear keyword data set, columnar keyword data set and free text keyword data set.
Linear Keyword Data Set	A linear data set is an ordered collection of data fields presented in a horizontal manner.
Columnar Keyword Data Set	A columnar keyword data set is an ordered collection of data fields aligned vertically under a horizontal array of field headers.

Free Text Keyword Data Set	A free text keyword data set is a free text statement of variable length bounded by its keyword data set name and the set terminator mark of '//'.

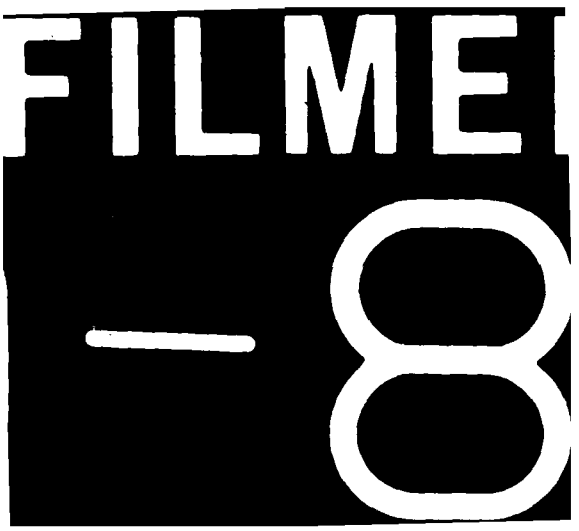
Abbreviations

altchn	A JINTACS DFI that is defined as a concantenation of a variable number of non-CHN DFI's where the first such DFI's value dictates which of several possible combinations of other DFI's will follow it
AUTODIN	Automatic Digital Network
BNF	Backus Naur Format
C	C is the programming language used to develop the ITSME software
CAFMS	Computer Aided Force Management System
Chain DFI	A JINTACCS DFI that is defined as a concatenation of a fixed number of non-CHN DFI's in a fixed order

COM	Character Oriented Message
COMNOTAB	A JINTACCS DFI whose data use identifiers all utilize a common set of data item codes. In addition, no table of values is supplied in the JINTACS MED
COMTAB	A JINTACCS DFI whose data use identifiers all utilize a common set of data item codes and in addition the JINTACCS MED supplied a table of legal values
DICP	Developmental Interface Change Proposal
DIF	A JINTACCS DFI whose data use identifiers use two or more sets of different data item codes some of which have a table of legal values in the JINTACCS MED, others of which do not
DIFTAB	A JINTACCS DFI whose data use identifiers use two or more sets of different data item codes, all of which have a table of legal values specified in the JINTACCS MED

DIFNOTAB	A JINTACCS DFI whose data use identifiers use two or more sets of different data item codes, none of which have a table of legal values specified in the JINTACCS MED
DFI	Data Field Identifier
DUI	Data Use Identifier
FDFH	Field Descriptor and Field Header
IBLD	ITSME data table build program, the executable name of the program described in this specification
ITSME	Interoperability Through Structured Message Exchange
JAMPS	JINTACCS Automated Message Preparation System
JANAP 128(H)	AUTODIN Operating Procedures
JINTACCS	Joint Interoperability of Tactical Command and Control Systems

JOMPSS	JINTACCS Oriented Message Processing Support Software
KDS	Keyword Data Set
MED	Message Element Dictionary
MSG	Message
Opfac	Operational Facility
PLA	Plain Language Address
PTU	Participating Test Unit
PWB/UNIX	PWB/UNIX is the operating system used to develop the ITSME software and it is a Trademark/Service Mark of the Bell System. PWB stands for Programmers Work Bench
TIDP	Technical Interface Design Plan
UNIX	UNIX is a Trademark/Service Mark of the Bell system


FILME
8