NASA Contractor Report 159221

ISSYS - AN INTEGRATED SYNERGISTIC SYNTHESIS SYSTEM

A. R. Dovi

KENTRON INTERNATIONAL, INC.
Hampton Technical Center
an LTV company
Hampton, Virginia 23666

CONTRACT NAS1-13500
February 1980

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

ABSTRACT

A computer aided design system called ISSYS is described. System concepts, usage and major elements of the system are presented. ISSYS controls the sequence of execution of the independent computer codes utilized. A common link between the independent computer codes is an independently constructed data base of information accessed by ISSYS. This data is constructed in a key-name format required by ISSYS. Data blocks are chained (linked) together via executive software. Computer code execution and data management are automatically handled by ISSYS. The system is broken down into major elements and described in their respective sections of this report.

The purpose of the system is threefold:

° The support of structural design studies on advanced and experimental aircraft concepts.

° The investigation of multidisciplinary, computer aided design and analysis methodology.

° The provision of detailed structural data for the evaluation of advanced analytical methods.

This report defines the major functions and rules of the ISSYS system. It is not intended as a stand-alone user's guide. It is assumed that the user is familiar with the Control Data Corporation Network Operating System (NOS) and associated software utilities. External computer codes utilized by the ISSYS system are documented by the developers. A bibliography is contained in this report for further reference by the user on these programs and other topics of interest.

FOREWORD

This document has been prepared for the Structures and Dynamics Division, Design Methods Section, NASA Langley Research Center, by Kentron International, Inc., Hampton Technical Center. The purpose of this document is to present to the user a description and usage of the major subsections of the ISSYS system. Several job control examples as well as a schema for generating data for the system are presented in Appendix A.

Hampton Technical Center provides technical integration assistance to the Structures and Dynamics Division, Aeroelastic Optimization Office under Contract NAS1-16000. This report was prepared by Mr. A. R. Dovi with significant contributions from Messrs. W. H. Greene, D. W. Gross, W. L. Kurtze, E. W. Shields, and G. A. Wrenn. Special acknowledgement is given to the pioneering efforts of Mr. L. A. McCullers in the development of the ISSYS system.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (concluded)

LIST OF FIGURES

1. INTRODUCTION

1.1 PURPOSE

The Integrated Synergistic Synthesis System (ISSYS) is a flexible, user oriented, data independent system of computer codes and executive software procedures associated with the design and analysis of aircraft structures. Current capabilities include: the generation of aeroelastic loads, trimming to a desired fuel mass and center of gravity, static analysis, structural sizing for multiple load conditions, dynamic and flutter calculations, gust response computations and active controls design and analysis.

The purpose of ISSYS is threefold:

° The support of structural design studies on advanced and experimental aircraft concepts.

° The investigation of multidisciplinary, computer aided design and analysis methodology.

° The provision of detailed structural data for the evaluation of advanced analytical methods.

The computer codes perform specific calculations, such as finite element structural analysis or the computation of aerodynamic influence coefficients. Executive software utilizes the Control Data CYBER series computer under the Network Operating System (NOS). Pre- and post-processor codes, which are part of the system, provide data management to integrate computer code input-output data sets. A database complex (LBASE) may be generated by the ISSYS System via executive software data processors. This complex provides required input data files to the various computer codes utilized during an analysis. Data may be input or retrieved from the database during the execution of each task via executive software.

1.2 SYSTEM CONCEPT

The ISSYS concept is to interface separate, stand-alone computer codes; integrating them into procedures which perform significant, independent tasks such as calculating new structural element sizes or performing a flutter analysis. Maximum use is made of existing computer programs (developed outside of the ISSYS team) and the capabilities of Control Data Corporation's Network Operating System (NOS). Relying on external sources in this manner decreases ISSYS development time, increases reliability and flexibility, and facilitates the incorporation of new capabilities. However, responsiveness to changes in the operating system or in externally maintained programs becomes crucial to keeping ISSYS operational.

Utilization of NOS Job Control Language as executive software permits easy, straightforward modification to executive software and allows the execution of user generated procedures and programs intermixed with ISSYS tasks. Using ISSYS Utility Procedures, any part of the system can be modified for a single run (during that run) for a special purpose application, or to checkout a proposed modification. With NOS executive software an existing capability can be modified or a new capability can be developed and checked out without changing the production version of ISSYS. This has produced a flexible, open-ended system which is being improved and expanded almost continuously.

ISSYS control is directed by the user and consists of a sequence of ISSYS and NOS control language commands. Selected examples are presented in the section titled <u>JOB CONTROL</u>.

2. SYSTEM DESCRIPTION

2.1 MAJOR ELEMENTS

The two major elements of ISSYS are the Executive Software and Computer Codes. Executive Software is based on Control Data Network Operating System (NOS) Control Language and NOS Control Statements as the host language. Executive Software consists of the following elements:

- TASK PROCEDURES
- UTILITY PROCEDURES
- AUXILIARY PROCEDURES
- DATA PROCESSORS
- PLOTTING PROCEDURES
- DATA MODIFICATION PROCEDURES

Each of the above elements is described in its respective section of this document.

Computer Codes utilized by, or contained in, the ISSYS System Library for engineering design/analysis are generally existing checked out computer codes, for which no program development is usually required. Each Computer Code performs analysis for an engineering discipline. Computer Codes utilized by ISSYS are described in the section titled PROGRAMS.

Job control of an engineering analysis/design with ISSYS is directed by the user/analyst via the ISSYS Command described in the section titled, THE ISSYS COMMAND. The user must supply a Local Database under the file name LBASE, or information required to create it. The LBASE must be in a text/key-name format as described in NOTE 3 of the Sub-Section titled, ISSYS NOTES AND REGULATIONS. The key-name allows ISSYS Data Processor Procedures to prepare required data for codes to be run.

Relationships between the elements of ISSYS during a job execution as well as the data flow are shown schematically in Figure 2.1-1.

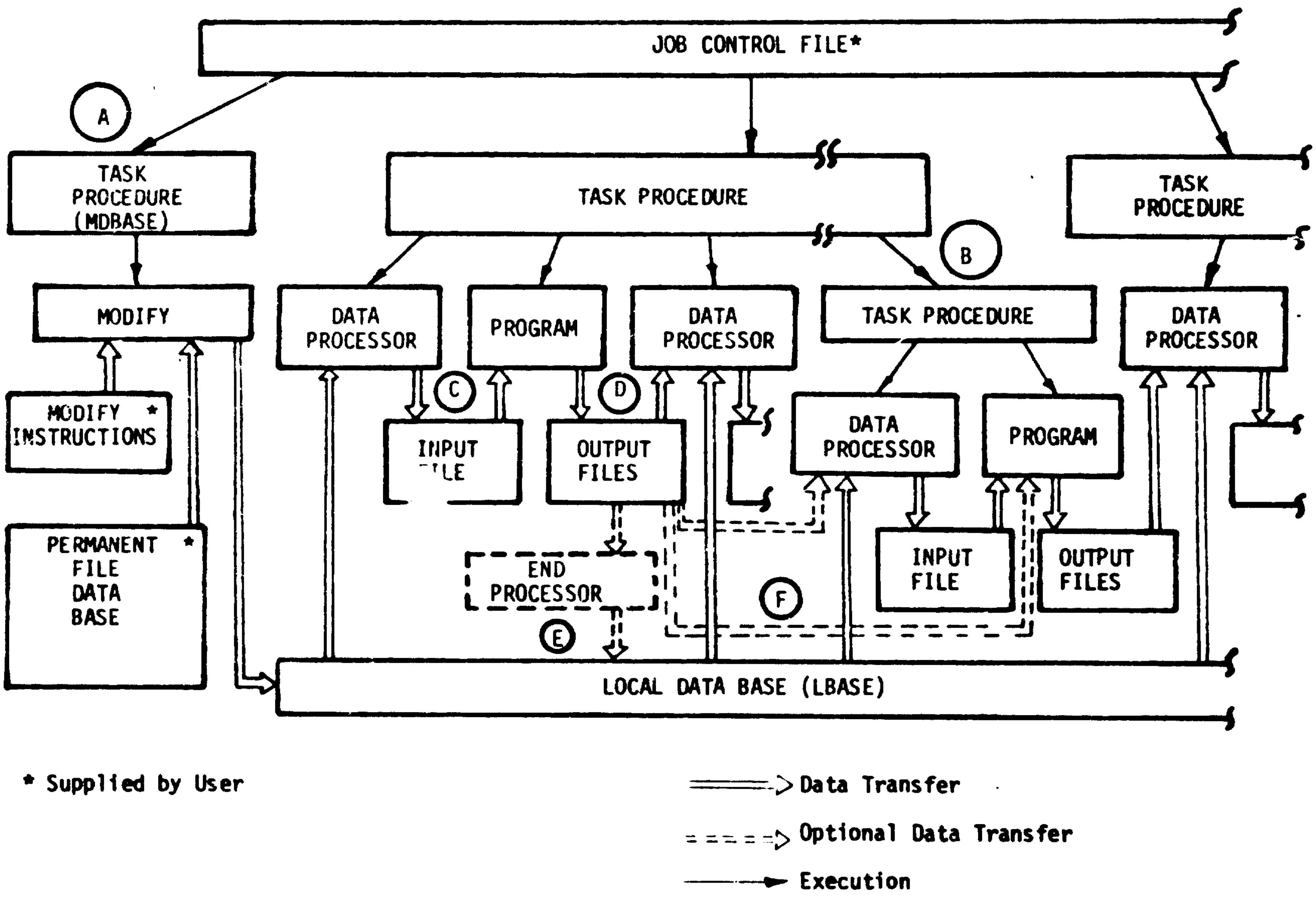

Figure 2.1-1 ISSYS execution and data flow

The execution sequence is specified by a series of ISSYS Commands in the Job Control File. Area (A) in Figure 2.1-1 shows how user-supplied MODIFY instructions and a permanent file data library are used by the CDC MODIFY Utility (called by Auxiliary Procedure MDBASE) to create the Local Database (LBASE). With LBASE established, the Job Control Deck can execute analysis and design Task Procedures. These typically execute Data Processors, Programs, and other Task Procedures (B). A Data Processor uses Data Blocks from LBASE to form an input file for a Program (C). The Program generates output files (D) which are normally used by the next Data Processor, etc.

Some output files, however, are put into LBASE usually after processing (E) by the calling Task Procedure. They can also be used in other Data Processors or directly in other Programs (F). The files used to transfer data from one user-called Task Procedure to another are described in the section on LOCAL FILES.

Executive software elements based on the host computer's operating system provide the capability to model similar techniques on other systems; both large and small computer mainframes. The ISSYS system concept may be hardware or software based. In the open-ended ISSYS system, tasks are linked together by the user/analyst via ISSYS Commands,

i.e., CALL(ISSYS(XQ=CMD NAME,A=arg_1,B=arg_2. . .

ISSYS Commands direct the system to perform specific analysis tasks utilizing data supplied by the user via LBASE in a specified schema compatable to the ISSYS system. A detailed description of the ISSYS Command is presented in the subsection titled THE ISSYS COMMAND.

2.2 SYSTEM LIBRARY

The ISSYS System Library resides as a single permanent file divided into four NOS-LIBEDIT type sub-libraries as shown in Figure 2.2-1. The first two consist of TEXT type records containing the ISSYS Procedures (LIB1) and the Data Processors (LIB2). The third sub-library (LIB3) consists of MODIFY-OPL type records containing input instructions for Programs such as AUTOLAY (used to assemble binary files for programs) and SORTMRG (used in alphabetizing records in a library). The fourth sub-library (ISSLIB) is a ULIB type user library. It is further sub-divided into two sub-libraries containing REL type relocatable binary records for each program and subroutine and TEXT type source decks, respectively.

The ISSYS Procedures in LIB1 can be classified into three categories; Task, Utility, and Auxiliary Procedures. Task Procedures, in general, perform engineering calculations as part of an analysis or design exercise. Utility Procedures are used in the maintenance of a permanent data base or of ISSYS itself--modifying or adding to any part of the system. Auxiliary Procedures are used by ISSYS to perform file manipulation or special output functions. The major elements that comprise the system are described in the following sections of this document.

User interface to the system library is via ISSYS Commands as defined in the section titled, THE ISSYS COMMAND. Optionally, the user may supply to the ISSYS library external computer codes, or other modifications/additions to the system as shown in Figure 2.2-2. As directed by Commands, ISSYS will replace existing elements (Procedures, Codes, OPL's, REL's, TEXT) or add new elements to appropriate libraries. The local data base (LBASE), which is accessed and edited by ISSYS to supply input to various computer codes, is independent of the ISSYS libraries and is supplied by the user or generated at execution time by ISSYS. LBASE is kept in memory as a local text file. After all ISSYS Commands are complete, LBASE is released.

LBASE is accessed by ISSYS via the LBEDT Command described in the section titled THE LBEDT COMMAND. Interface with the host computers operating sysem and Utility Software is via ISSYS Executive Software. Major elements are presented in the section titled Major Elements.

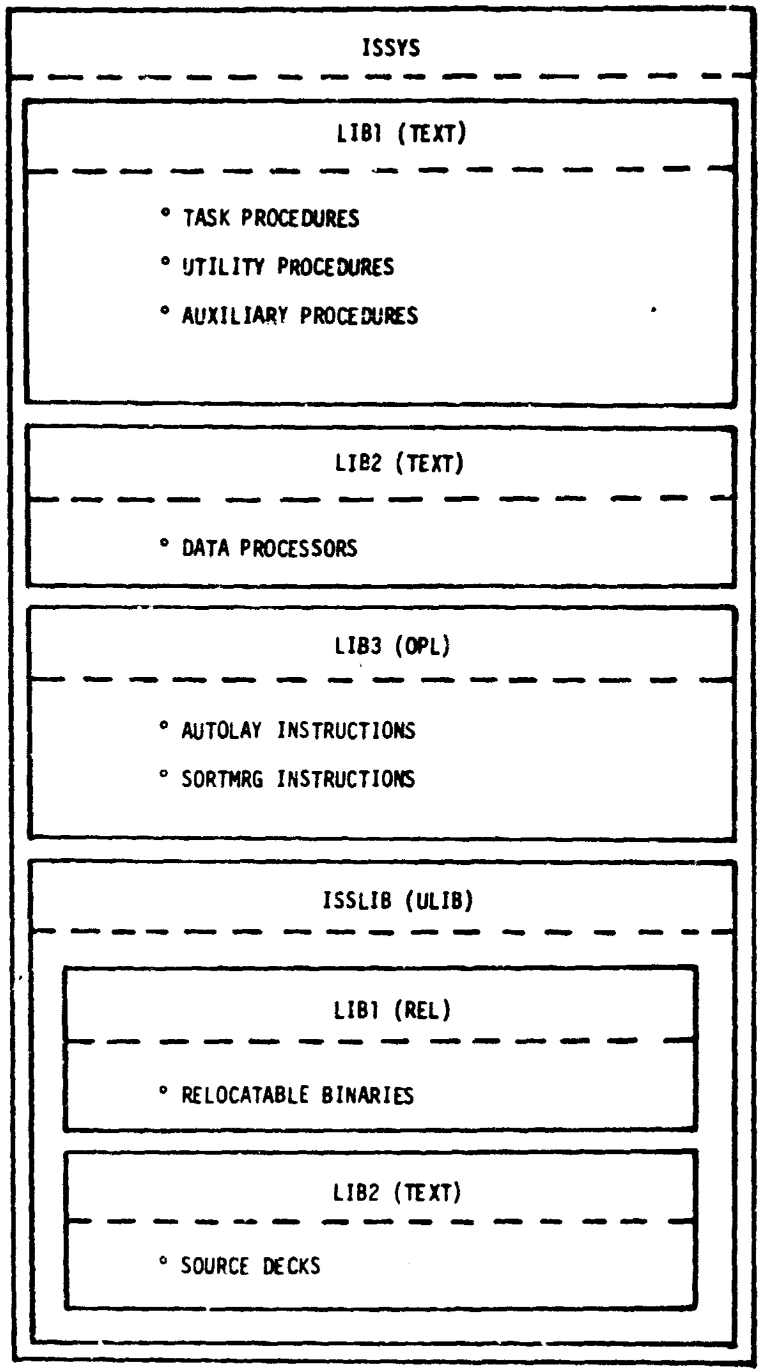

Figure 2.2-1 ISSYS library organization

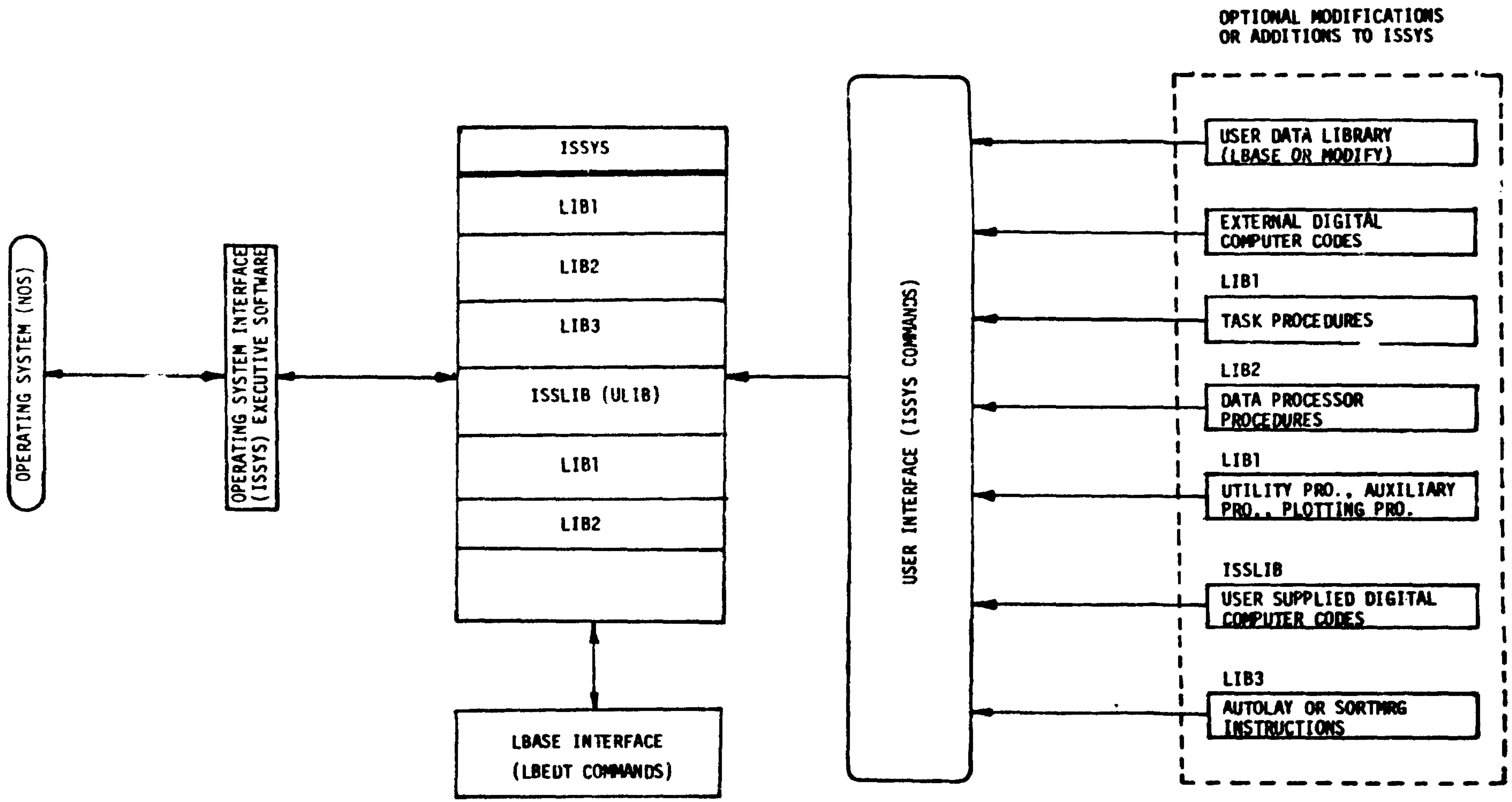

Figure 2.2-2 User modifications or additions.

2.3 THE ISSYS COMMAND

Function: This is the user's control directive of the ISSYS system. The specified ISSYS Command Name (CMD NAME) is retrieved from the ISSYS file and executed with the parameters supplied by the user.

Auxiliary Files Used: ISSYS, ISSLIB, LBEDT, GETLB, COST, XQ, I, L, A, B, C, D, E, R, "CMD NAME", XI, XL, XA, XB, XC, XD, XE, XR, XX, AA, BB, LL, XO, DIN, ISEXO, NOPRINT, ISERR, XEDIT. (See also the specific commands).

Description:

CALL(ISSYS(XQ=CMD NAME,I=INN,L=OUT,A=AAA,B=BBB,C=CCC,D=DDD,E=EEE,R=RFL)

XQ = ISSYS "CMD NAME". If a file exists with the same name as "CMD NAME", the local file will be used in place of the ISSYS "CMD NAME".

I = Input file name. Used for XEDIT, MODIFY, LIBEDIT, etc., instructions only. Not used for source decks or data blocks. Not used by TASK Commands.

L = File for printed output. For TASK Commands, only the "optional" output is on the L file (default: L=ISERR).

A,B,C,D,E = Parameters/file names required by the Command.

R = Optional RFL value. Used to change the field length from the default values set by the Command.

2.4 THE LBEDT COMMAND

Function: This is the link between the ISSYS Commands and the Data Processors. The specified Processor is retrieved from the ISSYS file and executed with the parameters supplied by the Command Procedure.

Auxiliary Files Used: ISAVE, ISDAT, IBASE, ISEDT, NOPRINT, ISEXO, XX, PIN, AA, BB, LL, XL, XO, XA, XB, plus files used by the specific processors.

Description:

CALL(LBEDT(XX=DPROC,LL=OUT,AA=AAA,BB=BBB,DIN=FILE)

XX = Data Processor name.

LL = File for printed output. (Default=ISERR).

AA,BB = Parameters/file names required by the Processor.

DIN = File to contain generated input data.

2.5 ISSYS NOTES AND REGULATIONS

Notes and regulations are presented in this section to provide a reference concerning the disposition of certain files and rules established for the maintenance and operation of the system. A clear understanding of these notes and regulations will avoid minor user errors in the future.

NOTES:

1. All modifications and changes made by ISSYS COMMANDS affect local files only, permanent files must be replaced by the user.

2. If a local Procedure File exists with the same name as an ISSYS COMMAND the local file will be utilized in place of the ISSYS COMMAND version.

3. If a local file exists with the name LBASE, ISSYS COMMAND MDBASE is not required to process data. The local LBASE will be utilized by ISSYS; (see REGULATION 5).

   The LBASE supplied must be in the following key-name format.

```
$$XXXXX
(logical data block 1)
-EOR-
$$XXXXX
(logical data block 2)
-EOR-
   .
   .
   .
-EOR-
$$XXXXX
(logical data block n)
```

4. If a local data file in key-name format exists with the same name as an ISSYS DATA PROCESSOR, the local file will be used in place of the data on LBASE generated by the DATA PROCESSOR. No editing is done by ISSYS to the local file.

5. Relocatable binary programs or subroutines present on file ISREL will be used in place of programs or subroutines with the same names on the ISSYS library.

6. All ISSYS LIST--- type utility commands generate a local, sorted copy of the file being listed. This sorted copy is suitable for replacement of the original file if desired.

2.5 ISSYS NOTES AND REGULATIONS (continued)

7. For all PUT--- and GP--- type ISSYS COMMANDS, if the input (I) file is not local, it will be assigned to the keyboard for TELEX jobs. For all GP--- ISSYS COMMAND, if (I) file is an alternate file supplied by the user, it will be rewound.

REGULATIONS:

1. ISSYS COMMAND Procedures will not use NOS sense switches or R-Registers except EF (Error Flag).
2. Use of GOTO's for control skipping and looping is strongly discouraged.
3. MODIFY OPL's operated on using ISSYS will not use "#" as a prefix character.
4. All data management performed by the ISSYS system will be based on a logical record, TYPE/NAME relationship.
5. The data base complex operated on by ISSYS must have the name LBASE at execution time, see NOTE 3 above.

3. TASK PROCEDURES

3.1 INTRODUCTION

The Task Procedure is the user's execution control of ISSYS for an engineering analysis. Calls to these Procedures provide the user with a Network Operating System (NOS) based executive language. Tasks to be performed and the order in which they are executed are user controlled. Procedure calls may be interspersed with NOS control language supplied by a user.

The modularity of this document is maintained in each Procedure's description. For descriptions of procedures called, refer to that Procedure. Logical data blocks required are documented in the section titled DATA PROCESSORS.

Files required, and files created, are those used only by the Procedure being described. All Task Procedures require the presence of the database library LBASE. LBASE is automatically generated by ISSYS, see Appendix A, or may be supplied by the user, see the section titled ISSYS NOTES AND REGULATIONS.

3.2 ALPHABETICAL INDEX OF TASK PROCEDURES

3.3 TASK PROCEDURE DESCRIPTIONS

Each Task Procedure, including files required for execution and those created, is described in this section. Task procedures typically execute Data Processors, Programs (computer codes), and other Task Procedures required for an engineering analysis.

The Task Procedure functions are presented, whether they are called by the user or by other Task Procedures. The description includes an explanation of the arguments in the ISSYS COMMAND and a step by step indication of the major operations performed.

LBEDT (xxx) indicates a call to the local-database-editor Auxiliary Procedure to use Data Processor (xxx) to prepare an input file for a program execution. ISSYS(XQ=yyy) indicates a call to ISSYS Procedure (yyy) or to an ISSYS Auxiliary Procedure using the ISSYS COMMAND to perform the specified operation.

3.3.1 DLAT - TASK PROCEDURE

Function: Generates generalized aerodynamic force matrices using the DLAT module of the Interaction of Structures, Aerodynamics, and Controls (ISAC), ref. NASA TM-80040.

Procedures Called: LDO

Called By: User

Files Required: LBASE (Tries to get it if not available.)
TAPE5; TAPE9 (data base complex of ISAC).

Files Created: TAPE9 (Add data to data base complex.); DLATPLT

Parameters: C = File name of TAPE5 data (option).

D = User number where the file resides (option).

NOTE: Both C and D must be specified if this option is used.

Ex: CALL(ISSYS(XQ=DLAT,C=SCAR5,D=227940C)

Description:

° Tries to get TAPE5 via parameters C and D if this option is utilized. If TAPE5 is a local file it will be used. If TAPE5 is not assigned via the above methods ISSYS will generate it from LBASE.

° LBEDT(DLTDAT) - Extracts DLAT data block $$DLTFL from LBASE.

° ISSYS(XQ=LDO) - Create DLAT absolute binary file.
DLAT - Generate AIC matrices and gust loads for a non-planar surface.

° If plots are requested a plot vector file DLATPLT is generated.

3.3.2 DLIN - TASK PROCEDURE

Function: Generates structural to aerodynamic transformation matrix utilizing the surface spline interpolation method using the DLIN module of the Interaction of Structures, Aerodynamics, and Controls (ISAC), ref. NASA TM-80040.

Procedures Called: LDO

Called By: User

Files Required: LBASE (tries to get it if not available),
GSTIN (generated by Task Procedure DYNAM)

Files Created: TAPE9 (Data base complex of ISAC), TAPE5, DLINPLT

Parameters: None
Ex: CALL(ISSYS(XQ=DLIN)

Description:

° Prepares input files and executes the ISAC DLIN module

° LBEDT (DLNDAT) - Extracts DLIN data block $$DLNFL from LBASE.

° ISSYS (XQ=LDO) - Create DLIN absolute binary file.
DLIN. - Downwash and integration matrices are stored

° If plots are requested, a plot vector file DLINPLT is generated.

3.3.3 DYNAM - TASK PROCEDURE

Function: Generates natural modes and frequencies using the SPAR structural analysis system and prepares them for gust and flutter analysis.

Procedures Called: LDDYN, MMINRT, SPARPLT

Called By: User

Files Required: LBASE (tries to GET it if not available)

Files Created: FLTIN, GSTIN

Parameters: A = Number of modes processed

B = Turns on gust option

Ex: CALL(ISSYS(XQ=DYNAM,A=12,B=1)

Description:

- LBEDT(CHGSETP) - Extracts Design data blocks $$DSWDM; $$DCVDM; $$DCVPR; $$DSWPR from LBASE.

- Attaches SPAR and DCU programs.

- ISSYS(XQ=LDDYN) - Generates Executable files of CGMASS; MDPROC; MODSEP; STPROC; TRANP.

- LBEDT(CHGDYN1) - Extract data blocks $$CONDM; $$CGRID; $$RODPR; $$CCVPR; $$SWCON; $$CLMSS; $$UPROD; $$STDEF from LBASE.

  SPAR - Generates stiffness and mass matrices.

- ISSYS(XQ=MMINRT) - Task Procedure to calculate fuel, mass, and aircraft inertia data.

- LBEDT(CHGDYN2) - Calculate modes, frequencies, and gust stress SPAR coefficients using data blocks $$STNML; $$DYANL from LBASE.

- ISSYS(XQ=SPARPLT) - If plots are requested a plot vector file SPLT is generated.

- LBEDT(MDIN,AA=A) - Translate SPAR modes and frequencies to FTN MDPROC format using data block $$MDIN from LBASE.

- LBEDT(MSIN,AA=A) - Separate modes into wing, tail, etc. components. MODSEP

- STPROC - Separates stress into shear, moment, and torque components for each mode.

3.3.4 DYNAR - TASK PROCEDURE

Function: Performs a dynamic response flutter and gust analysis using the DYNARES module of the Interaction of Structures, Aerodynamics, and Controls (ISAC), ref. NASA TM-80040.

Procedures Called: LDO

Called By: User

Files Required: LBASE (tries to get it if not available), GSTIN, TAPE9 (Data base complex of ISAC).

Files Created: RMSTR (RMS stress file)

Parameters: C = File name (fn) of TAPE9 data (option).

D = User number where the file (fn) resides (option).

NOTE: Both C and D must be specified if this option is used.

Ex: CALL(ISSYS(XQ=DYNAR,C=SCAR9,D=227940C)

Description:

- LBEDT(DGMDAT) - Extracts DGMBN data block $$DGMFL from LBASE.
  DGMBN. - Generates generalized mass matrix (TAPE7) and damping coefficients (TAPE8).
- LBEDT(DYNDAT) - Extracts DYNARES data block $$DYNAR from LBASE.
- ISSYS(XQ=LDO) - Create DYNARES absolute binary file.
  DYNARES. - Perform flutter and gust analyses.

3.3.5 FINL - TASK PROCEDURE

Function: Perform maneuver loads update and generate stress data.

Procedures Called: LLOOP, GSPIN, LBASE

Called By: User

Files Required: SPARLA, SPDIN, STDIN

Files Created: SPARLN

Parameters: None

Ex: CALL(ISSYS(XQ=FINL)

Description:

- ISSYS(XQ=LLOOP) - Task procedure to update maneuver loads.
- SPAR - Compute maneuver deflections and performs stress analysis.
- LBEDT(GSPIN) - Setup input data for program GSPAR using data block $$STNML from LBASE. Also executes TRANP.
- GSPAR - Reformats stresses and add other necessary element information.

3.3.6 FLTSK - TASK PROCEDURE

Function: Computes generalized unsteady aerodynamic forces (GAF) for flutter using Subsonic Kernel Function.

Procedures Called: SETSK

Called By: FLTWT

Files Required: LBASE

Files Created: ADDOUT

Parameters: A = No. of modes

B = PLOT On/Off switch = NOPLOT turns plots off. Default is plots on.

Ex: CALL(ISSYS(XQ=FLTSK,A=16,B=NOPLOT)

Description:

- ISSYS(XQ=SETSK) - Mode shape processing.
- SUBKRN - Compute GAF for wing.

  Compute GAF for tail (if present).
- ADDSK - Adds tail GAF to wing GAF.

NOTE: All tail computations are triggered by the existence of $$MP2IN in LBASE.

3.3.7 FLTWT - TASK PROCEDURE

Function: Wing plus tail flutter analysis.

Procedures Called: FLTSK, LDO

Called By: User

Files Required: FLTIN (trys to get it from user's account if not available)

Files Created: SPL1, SPL2, VGPLOT (if these exist, they are not recalculated)

Parameters: A = unsteady aerodynamics program

B = number of modes to be used

Ex: CALL(ISSYS(XQ=FLTWT,A=SUBKRN,B=12)

Description:

- Separate FLTIN into FREQS and MSPOUT
- ISSYS(XQ=LDO) - Generate FLTDET and SUBKRN absolute binary files.
- ISSYS(XQ=FLTSK) - Task Procedure to generate generalized unsteady aerodynamics forces for flutter, if file SUBKRN is assigned.
- FLTDET - Perform flutter analysis.
- Copy SAVPLT, from FLTDET, onto VGPLOT.
- Copy TTPE16, from FLTDET, onto VGTEK.

3.3.8 INIT - TASK PROCEDURE

Function: Perform initial SPAR execution.

Procedures Called: SPARI

Called By: User

Files Required: LBASE, (see SPARI, MMINRT, LLOOP)

Files Created: (see SPARI, MMINRT, LLOOP, SPARPLT)

Parameters: None

Ex: CALL(ISSYS(XQ=INIT)

Description:

° ISSYS(XQ=SPARI) - Perform initial SPAR execution for a static analysis and resizing. Secondary procedures perform fuel distribution and rigid loads computation.

° If SPAR plots have been requested via the MODIFY OPL, or SPAR plot directives are supplied on a users LBASE (See section titled ISSYS NOTES AND REGULATIONS), a plot vector file SPLT is created.

3.3.9 INITC - TASK PROCEDURE

Function: Perform initial SPAR execution and create restart tape.

Procedures Called: SPARI

Called By: User

Files Required: LBASE

Files Created: (see SPARI, MMINRT, LLOOP, SPARPLT)

Parameters: A = Volume Serial Number (VSN) of restart tape. The SPARLA file is written to this tape.

Ex: CALL(ISSYS(XQ=INITC,A=MN1150)

Description:

- ISSYS(XQ=SPARI) - Perform initial SPAR execution for a static analysis and resizing. Secondary procedures perform fuel distribution and rigid loads computation.
- If SPAR plots have been requested via the MODIFY OPL, or SPAR plot directives are supplied on a users LBASE (see section titled ISSYS NOTES AND REGULATIONS), a plot vector file SPLT is created.

3.3.10 INITR - TASK PROCEDURE

Function: Reads magnetic restart tape containing initial SPAR execution data.

Procedures Called: SPARR

Called By: User

Files Requested: LBASE, (see SPARR, MMINRT, LLOOP)

Files Created: (see SPARR, MMINRT, LLOOP)

Parameters: A = VSN - Magnetic restart tape number containing SPAR library SPARLA from initial SPAR analysis (see INITC)

Ex: CALL(ISSYS(XQ=INITR,A=VSN)

Description:

° ISSYS(XQ=SPARR) - Reads restart tape to create SPAR library SPARLA. Secondary procedures perform fuel distribution and rigid loads computation.

If SPAR plots are desired, after call to INITR, execute one of the ISSYS Plotting Procedures with A = SPLT, or SPLT may be saved for use with ISSYS plotting procedure TVPLOT. See the database being used for a menu of *DEFINE directives for desired plots. SPAR plot module specifications may also be input via a file called TBASE.

° If SPAR plots have been requested via the MODIFY OPL, or SPAR plot directives are supplied on a users LBASE (see section titled ISSYS NOTES AND REGULATIONS), a plot vector file SPLT is created.

3.3.11 LLOOP - TASK PROCEDURE

Function: Compute new maneuver loads.

Procedures Called: CHGTRIM

Called By: SPARI, SPARR, LOOP

Files Required: LBASE, SPARLG*, SPARLH, ALOAD, OZD, MAS*, SPARLF*, CGDIN

Files Created: OZD, SPDIN, STDIN*, LDPLOT

Parameters: None

Ex: CALL(ISSYS(XQ=LLOOP)

Description:

° LBEDT(CHGTRIM) - Extracts data blocks $$DESGN; $$CGRID from LBASE.

° TRIM - Compute new maneuver loads based on deflections from SPARLH file.

° A plot vector file LDPLOT is generated if file SAVPLT exists after execution of program TRIM.

NOTE: Files marked by a "*" are not always required.

MAS	is used only when delta-alpha gust analysis is performed.
SPARLG SPARLF	are used only in procedures INIT, INITC, INITR.
STDIN	is created only in INIT, INITC, INITR.
OZD	is updated with each call to LLOOP.

3.3.12 LOOP - TASK PROCEDURE

Function: Perform maneuver loads update and compute static deflections due to these loads.

Procedures Called: LLOOP

Called By: User

Files Required: SPARLA (see LLOOP), LBASE

Files Created: SPARLH

Parameters: None

Ex: CALL(ISSYS(XQ=LOOP)

Description:

- ISSYS(XQ=LLOOP) - Perform maneuver load update.
- SPAR - Compute static deflections due to these loads. Deflections written on file SPARLH.

3.3.13 MDBASE - TASK PROCEDURE

Function: Generates a Local Database from a specified MODIFY data library.

Procedures Called: Mode

Called By: User

Files Required: "I" and "A" (see below)

Files Created: LBASE

Parameters: I = file containing MODIFY instructions

L = file for printed output

A = name of MODIFY Old Program Library (OPL)

B = user number for location of "A"

Ex: CALL,ISSYS(XQ=MDBASE,I=INPUT,L=OUTPUT,A=OPL,B=985950C)

Description:

- ISSYS(XQ=MODE) - Task Procedure to perform MODIFY execution.
- Adds Random Access Directory to LBASE, Catalogs LBASE onto "L" file.

3.3.14 MMINRT - TASK PROCEDURE

Function: To compute fuel mass distribution.

Procedures Called: CHGMMI

Called By: SPARI, SPARR

Files Required: SPARLF

Files Created: SYSFUEL
MAS
CGDIN
FUELM

Parameters: L = file for printed output

Ex: CALL(ISSYS(XQ=MMINRT,L=ISERR)

Description:

° LBEDT(CHGMMI) - Extracts data blocks $$CGRID; $$INLST; $$DESGN; $$TANK from LBASE.

° CGMASS - Distributes fuel for each load case.

NOTE: Edits data block $$DESGN on LBASE.

3.3.15 RESE23 - TASK PROCEDURE

Function: Resizes E23 rod elements by fully stressed design.

Procedures Called: None

Called By: User

Files Required: SPARLA data sets BC BTAB, STRS E23, MATC BTAB, DEF E23, E23 EFILE; Optional (MINI GAGE)

Files Created: SPARLQ which contains the new BC BTAB data set produced by RSRD.

Parameters: LL = file for printed output.

RR = field length requirement

Ex: CALL(ISSYS(XQ=RESE23,LL=OUTPUT,RR=RFL)

Description:

- Executes SPAR to calculate and optionally print E23 stresses using the GSPAR processor RSRD.
- LBEDT(XX=PRERR) - Extracts data block $$PRERR from LBASE. SPAR - Calculate and optionally print E23 rod stresses.
- LBEDT(XX=RRDEF) - Extracts data block $$RRDEF from LBASE. GSPAR - E23 rods are resized using processor RSRD. A new SPAR type library, SPARLQ, is produced which contains only the BC BTAB data set. This may be saved for a restart run and is automatically merged with SPARLA by procedure INIT.

3.3.16 RSZG - TASK PROCEDURE

Function: Prepares stress files and resizes shear webs or shear webs and cover panels.

Procedures Called: None

Called By: TRSC, TRSG, TRSM

Files Required: LBASE; SPARLH (containing stress data); TAPE80 (TRSC and TRSM; TAPE60 (TRSG)

Files Created: RMSTR; GAGE34; GAUGE44

Parameters: L = file for printed output

A = TAPE80 (*OPCOM input file from TRSC and TRSM)
TAPE60 (*OPCOM input file from TRSG)

Ex: CALL(ISSYS(XQ=RSZG,A=TAPE80,L=ISERR)

Description:

° LBEDT(OPIN) - Extracts data blocks $$OPIN and $$GSTML from LBASE.

° GUSTB - Combines gust stresses with steady-state stresses for use in resizing structure.

° BRESIZE - Resizes shear webs (and cover panels if called from TRSG) using weight-strength method.

3.3.17 SETPR - TASK PROCEDURE

Function: Sets up Programs and files for a static deflection analysis when rigid load data is available.

Procedures Called: LDSTZ

Called By: User

Files Required: LBASE, CLOAD, DLOAD, OCS, AERO (tries to get them from user's account if not local).

Files Created: DPROPT, DPROPS, DMASST, DMASSS

Parameters: None

Ex: CALL(ISSYS(XQ=SETPR)

Description:

- Acquires other required Programs (SPAR, DCU, BGSPAR)
- LBEDT(CHGSETP) - Extracts Design Data Files from LBASE
- ISSYS(XQ=LDSTZ) - Set up Executable Program Files from ISSYS.

3.3.18 SETSK - TASK PROCEDURE

Function: Controls the mode shape processing.

Procedures Called: LDO, LDR

Called By: FLTSK

Files Required: LBASE, MSPOUT (gets from user's account it if not local)

Files Created: MP1OUT, MP2OUT, MDPLOT, MDTEK

Parameters: None

Ex: CALL(ISSYS(XQ=SETSK)

Description:

- ISSYS(XQ=LDO) - Creates MPROC executable file if it is not already assigned.
- ISSYS(XQ=LDR) - Creates CMACH executable file if it is not already assigned.
- ISSYS(XQ=LDR) - Creates ADDSK executable file if it is not assigned.
- Splits file MSPOUT into files MSWING and MSTAIL.
- LBEDT(SKMACH) - Extracts CMACH data from LBASE.
  CMACH - Change Mach number and prepare input files.
- IF (FILE(MP1,AS)) MPROC(MP1, . . .) - Process wing mode shapes.
- IF (FILE(MP2,AS)) MPROC(MP2, . . .) - Process tail mode shapes.
- Modal plots are created on file MDPLOT and MDTEK.

3.3.19 SETUP - TASK PROCEDURE

Function: Generates steady aerodynamic load random access file (ALOAD) and assembles all relocatable binary program files needed for a static analysis.

Procedures Called: HDR, CHGSETP, LDSTZ, CHGPG1, PG2IN

Called By: User

Files Required: LBASE (tries to get it from user's account if not local)

Files Created: ALOAD
AERPLT*
WASP*

Parameters: A = ONE yields output from program ONE

B = MPR yields output from program MDPROC

C = TWO yields output from program TWO

D = AERPLT yields plot vector file of camber and delta c_p's due to camber and angle of attack.

E = WASP yields aerodynamic input data for the ROT steady aerodynamics program, which generates Program WASP input data.

Ex: CALL(ISSYS(XQ=SETUP,A=ONE*,B=MPR*,C=TWO*,D=AERPLT*,E=WASP*)

NOTE: Output from programs ONE and TWO are put on file ISERR when B parameters are not specified.

* Optional

Description:

° LBEDT(CHGSETP) - Extracts data blocks $$DSWDM; $$DCVDM; $$DCVPR; $$DSWPR from LBASE.

° ISSYS(XQ=LDSTZ) - Set up absolute binary program files from ISSYS.

° LBEDT(CHGPG1) - Extracts $$GEOM; $$CGRID; $$AEROS from LBASE.
PROG1 - Generate Woodward-Carmichael input data.
MPROC - Processes the lifting surface mode shapes.

(continued)

3.3.19 SETUP - TASK PROCEDURE (continued)

- A plot vector file AERPLT is generated (option).
- LBEDT(PG2IN) - Extracts data blocks $$PG2IN; $$DESGN from LBASE.
  PROG2 - Execute Woodward-Carmichael Program.
- More plotting data may be generated on file AERPLT (option).
- MPROC - Processes the lifting surface mode shapes.
- More plotting data may be generated on file AERPLT (option).
- MATRIX - Generates ALOAD file.

3.3.20 SPARI - TASK PROCEDURE

Function: Perform initial SPAR execution, compute rigid loads and rigid displacements.

Procedures Called: CHG59, MMINRT, LLOOP, SPARPLT

Called By: INITC

Files Required: LBASE (see MMINRT, LLOOP)

Files Created: SPARLA, SPARLF (see MMINRT, LLOOP), SPARLG, SPARLH, (see SPARPLT)

Parameters: B = Switch to mount restart tape, i.e. B=1 (mount tape 1).

C = Volume Serial Number (VSN) of restart tape.

Ex: CALL(ISSYS(XQ=SPARI,B=11,C=NN1150)

Description:

° LABEL,SPARLT - Mount restart tape.

° LBEDT(CHG59,AA=NUM) - Set up data for initial SPAR execution. If AA = a number, include SPAR data to create restart tape. Extracts data blocks $$CGRID; $$RODPR.

° SPAR - Perform initial SPAR execution.

° ISSYS(XQ=MMINRT) - Perform fuel distribution.

° ISSYS(XQ=LLOOP) - Perform rigid loads calculations.

° SPAR - Compute rigid static deflections.

° ISSYS(XQ=SPARPLT) - Generates plot vector file SPLT if plot directives are available for SPAR.

3.3.21 SPARPLT - TASK PROCEDURE

Function: Executes SPAR plotting processors

Procedures Called: SPLOT

Called By: DYNAM, SPARR, SPARI

Files Required: SPARLA

Files Created: SPLT (Plot vector file)

Parameters: L = ISSYS error file for abnormal termination

Ex: CALL(ISSYS(XQ=SPARPLT,L=ISERR)

Description:

° LBEDT(SPLOT,LL=ISERR) - Extracts data block $$SPLT from LBASE.
SPAR - Generates plot vector file.

NOTE: The plot vector file SPLT may be saved by the user for later use with ISSYS Task Procedure TVPLOT.

Ex: CALL(ISSYS(XQ=TVPLOT,A=SPLT,L=OUTPUT)

or one of the ISSYS off-line paper plot procedures, i.e., CAL11; VARIAN.

3.3.22 SPARR - TASK PROCEDURE

Function: Read restart tape from initial SPAR execution, compute rigid loads and rigid displacements.

Procedures Called: MMINRT, LLOOP

Called By: INITR

Files Required: LBASE (see MMINRT, LLOOP)

Files Created: SPARLA, SPARLF (see MMINRT, LLOOP), SPARLG, SPARLH, SPARLT (see SPARPLT)

Parameters: B = Volume Serial Number (VSN) of restart tape.

Ex: CALL(ISSYS(XQ=SPARR,B=NN1150)

Description:

- ° MODIFY - Setup input data for SPAR to read restart tape.
- ° SPAR - Read restart tape (SPARLT).
- ° ISSYS(XQ=MMINRT) - Perform fuel distribution.
- ° ISSYS(XQ=LLOOP) - Compute rigid loads.
- ° SPAR - Compute rigid static deflections.
- ° ISSYS(XQ=SPARPLT) - Generates plot vector file SPLT if plot directives are available for SPAR.

3.3.23 TOTL - TASK PROCEDURE

Function: Combines functions of INIT and FINL. Rigid loads only are computed.

Procedures Called: SPARI, GSPIN

Called By: User

Files Required: LBASE, (see SPARI)

Files Created: SPARLN

Parameters: None

Ex: CALL(ISSYS(XQ=TOTL)

Description:

- ISSYS(XQ=SPARI) - Perform initial SPAR execution for a static analysis and resizing. Secondary procedures perform fuel distribution and rigid loads computation.
- SPAR - Performs stress analysis
- LBEDT(GSPIN) - Setup input data for program GSPAR from data block $$STNML on LBASE.
- GSPAR - Reformats stresses and adds other necessary element information.

3.3.24 TOTLC - TASK PROCEDURE

Function: Combines functions of INITC and FINL. Rigid loads only are computed.

Procedures Called: SPARI, GSPIN

Called By: User

Files Required: LBASE, (see SPARI)

Files Created: SPARLN

Parameters: A = Volume Serial Number (VSN) of restart tape.

Ex: CALL(ISSYS(XQ=TOTLC,A=NN1150)

Description:

- ISSYS(XQ=SPARI) - Perform initial SPAR execution for a static analysis and resizing. Secondary procedures perform fuel distribution and rigid loads computation. Creates restart tape.
- SPAR - Performs stress analysis.
- LBEDT(GSPIN) - Setup input data for program GSPAR from data block $$STNML on LBASE.
- GSPAR - Reformats stresses . adds other necessary element information.

3.3.25 TOTLR - TASK PROCEDURE

Function: Combines functions of INITR and FINL. Rigid loads only are computed.

Procedures Called: SPARR, GSPIN

Called By: User

Files Required: LBASE, (see SPARR)

Files Created: SPARLN

Parameters: A = Volume Serial Number (VSN) of restart tape.

Ex: CALL(ISSYS(XQ=TOTLR,A=NN1150)

Description:

- ISSYS(XQ=SPARR) - Reads restart tape containing SPARLA file from initial SPAR execution.
- SPAR - Performs stress analysis.
- LBEDT(GSPIN) - Setup input data for program GSPAR from data block $$STNML on LBASE.

3.3.26 TRSC - TASK PROCEDURE

Function: Controls resizing of composite wing using nonlinear programming techniques for the cover panels.

Procedures Called: RSZG

Called By: User

Files Required: LBASE, SPARLN (containing stress data).

Files Created: DPROPT, DMASST, DPROPS, DMASSS, TAPE40, RZOUT

Parameters: None

Ex: CALL(ISSYS(XQ=TRSC)

Description:

- Sets up design properties and mass files from files created by RSZG and OPCOM (edits the cover properties file).
- ISSYS(XQ=RSZG) - Task Procedure to resize shear webs only using the weight-strength method.
- OPCOM - Resizes cover panels using nonlinear programming techniques.
- File RZOUT contains printable resize information from BRESIZE and OPCOM.

3.3.27 TRSG - TASK PROCEDURE

Function: Controls resizing of metal wing using weight-strength method.

Procedures Called: RSZG

Called By: User

Files Required: LBASE, SPARLN (containing stress data)

Files Created: DPROPT, DMASST, DPROPS, DMASSS, RZOUT

Parameters: None

Ex: CALL(ISSYS(XQ=TRSG)

Description:

- Sets up design properties and mass files from files created by RSZG.
- ISSYS(XQ=RSZG) - Task Procedure to resize shear webs and cover panels using weight-strength method.
- File RZOUT contains printable resize information from BRESIZE.

3.3.28 TRSM - TASK PROCEDURE

Function: Controls resizing of metal wing using nonlinear programming techniques for the cover panels.

Procedures Called: RSZG

Called By: User

Files Required: LBASE, SPARLN (containing stress data).

Files Created: DPROPT, DMASST, DPROPS, DMASSS, TAPE40, RZOUT

Parameters: None

Ex: CALL(ISSYS(XQ=TRSM)

Description:

- Sets up design properties and mass files from files created by RSZG and OPCOM.
- ISSYS(XQ=RSZG) - Task Procedure to resize shear webs only using the weight-strength method.
- OPCOM - Resizes cover panels using nonlinear programming techniques.
- File RZOUT contain printable resize information from BRESIZE and OPCOM.

4.2 FILE DESCRIPTIONS

L - Contains output from various utilities executed for each ISSYS command.

LBASE - Local data base containing all input data required for a given run. Can be generated from a permanent file in MODIFY format using MDBASE, input on cards, or by any other method deemed expedient.

ISERR - Optional output file to be used in debugging. Returned at the end of each user-called Task Procedure that is successfully executed. To print, REWIND and COPYSBF it after the EXIT card.

NOPRINT - Auxiliary ISSYS output file. Returned at the end of each ISSYS Command.

ISLST - Contains a listing of each of the ISSYS procedures used. To print this file:

```
REWIND,ISLST.
COPYSBF,ISLST,OUTPUT.
```

4.2.1 DESIGN DATA FILES

The following four files contain design data for the elements to be resized. If they are not already present, they are extracted from LBASE by CHGSETP called from SETUP, SETPR, or DYNAM. They are generated by calls to TRSC, TRSM, and TRSG.

DPROPT - Cover element properties ($$DCVPR)

DMASST - Cover distributed masses ($$DCVDM)

DPROPS - Shear web element properties ($$DSWPR)

DMASSS - Shear web distributed masses ($$DSWDM)

TAPE40 - Contains design data for composite transfer elements. Used in plotting and in transfer of data to and from the Simplified Model flutter optimization system reference. Generated by a call to TRSC.

4.2.2 LOADS DATA FILES

AERO - Aerodynamic Influence Coefficient (AIC) matrix for maneuver condition. Generated in SETUP; used in LLOOP.

OCS - Old camber slopes used with relaxation factor in recomputing maneuver loads. Generated in SETUP or LLOOP; used in LLOOP.

CLOAD - Cruise load vector. Generated in SETUP; used in SPARI or SPARR.

DLOAD - Maneuver load vector. Generated in SETUP or LLOOP; used in SPARI, SPARR, LOOP, and FINL.

JGRID - Jig shape coordinates. Generated in SPARI or SPARR; used in LLOOP.

4.2.3 DYNAMICS AND FLUTTER DATA FILES

FLTIN - Natural frequencies in NAMELIST format. Generated in DYNAM; used in FLTWT. Plus mode shapes processed into deflections for each lifting surface. Generated in DYNAM; used in SETSK.

GSTIN - Frequencies, full mode shapes, nodal coordinates, and gust stress coefficients (if they exist). Generated in DYNAM; used in DYNAR.

SPL1 - Wing spline coefficients which depend only on geometry. Generated and used in SETSK and MPRSK. May be saved in one run and used in the next to save time.

5. LOCAL DATA BASE

5.1 INTRODUCTION

Primary external input to ISSYS is the Local Data Base (LBASE). LBASE is composed of data blocks of text records (refer to Figure A-3 Appendix A). Each record contains information to build a data model for a computer program. The records are accessed by ISSYS, using a key-name of the form ($$XXXXX) in Columns 1 through 7, where (XXXXX) is an identifier for a data block. The remaining 73 Columns may contain comments. Data input files to target programs are assembled by ISSYS Data Processors. Records are assembled by key-name, then packed and placed on the input file of the target computer program. Individual data blocks, described in this section, are utilized by the ISSYS team for the investigation of analysis methodologies.

5.2 ALPHABETICAL INDEX OF DATA BLOCKS

Section Number	Data Block
5.3.1	$$AEROS
5.3.2	$$CCVPR
5.3.3	$$CGRID
5.3.4	$$CLMSS
5.3.5	$$CONDM
5.3.6	$$DCVDM
5.3.7	$$DCVPR
5.3.8	$$DESGN
5.3.9	$$DGMFL
5.3.10	$$DLNFL
5.3.11	$$DLTFL
5.3.12	$$DSWDM
5.3.13	$$DSWPR
5.3.14	$$DYNAL
5.3.15	$$DYNAR
5.3.16	$$ELDSW
5.3.17	$$ENGDM
5.3.18	$$FDIN
5.3.19	$$FUELM
5.3.20	$$GEOM
5.3.21	$$GSPIN
5.3.22	$$GSTML
5.3.23	$$INLST
5.3.24	$$MACH
5.3.25	$$MDIN
5.3.26	$$MINGB
5.3.27	$$MP1IN
5.3.28	$$MP2IN
5.3.29	$$MSIN
5.3.30	$$OPIN (metal)
5.3.31	$$OPIN (composite)
5.3.32	$$PRERR
5.3.33	$$RODPR
5.3.34	$$RRDEF
5.3.35	$$SK1IN
5.3.36	$$SPLT
5.3.37	$$STANL
5.3.38	$$STCON
5.3.39	$$STDEF
5.3.40	$$SWCON
5.3.41	$$TANK
5.3.42	$$UPROD
5.3.43	$$XQTAB

5.3 DATA BLOCK DESCRIPTIONS

The Data Blocks which build the Local Data Base are individually described in this section. All data blocks are prefixed by a $$XXXXX key brd, where XXXXX is a user specified name, followed by data to form a Data Block. The keyword is required by ISSYS to identify Data Block locations when building the Local Database (LBASE). The description includes the computer program the Data Block is prepared for, type of analysis (static or dynamic), and the Data Block contents along with a description of the data, i.e., FORMAT, type, etc.

5.3.1 $$AEROS - DATA BLOCK

Input for Program(s): *ONE

Type of Analysis: STATIC

Contents: The number of nodes, along with a list of node numbers where deflections are applied, are defined. The panel numbers are also defined, along with the node numbers defining the panel corner points.

Data Description:

Card No.	Format	Type	Description
1	Free	INT	Number of nodes for deflections "NNOD"
2	Free	INT	List of "NNOD" node numbers for deflections (may use as many cards as required)
3	1-5	ALPHA	"PANEL"
	6-11	INT	Panel Number
	12-75	INT	Node numbers of panel corners (up to 8 nodes)

NOTE: Repeat Card 3 for each panel

4	1-6	ALPHA	"END RUN"

5.3.2 $$CCVPR - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static and Dynamic

Contents: Constant Shell Properties - (data only)

Data Description: Input to SPAR Processor TAB, Subprocessor SHELL SECTION PROPERTIES (SA) for all E31, E32, E33, E41, E42, E43 elements that are not to be resized.

Does not contain Subprocessor call statement (the SA card).

5.3.2 $$CCVPR - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static and Dynamic

Contents: Constant Shell Properties - (data only)

Data Description: Input to SPAR Processor TAB, Subprocessor SHELL SECTION PROPERTIES (SA) for all E31, E32, E33, E41, E42, E43 elements that are not to be resized.

Does not contain Subprocessor call statement (the SA card).

5.3.3 $$CGRID - DATA BLOCK

Input for Program(s): SPAR, ONE, THREE, SEVEN, CGMASS

Type of Analysis: Static and Dynamic

Contents: Cruise Grid - Nodal Data for the Cruise Shape (data only)

Data Description: One card per node point

Format: Free Field

Name	Description
K	Entry number
X	X-coordinate
Y	Y-coordinate
Z	Z-coordinate

Cards do not have to be sequential. For nodes used in the aerodynamic simulation, the lower surface node must follow the corresponding upper surface node specified in $$AEROS.

5.3.4 $$CLMSS - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static and Dynamic

Contents: Constant Lumped Masses (or Weights)

Data Description: Contains the following cards

RIGID MASSES or (RMASS)

. (Input to SPAR Processor TAB, Subprocessor
. RMASS for all constant lumped masses -
. does not include fuel mass)
.
.
.
.

5.3.5 $$CONDM - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static and Dynamic

Contents: Constant Distributed Weights - (data only)

Data Description: Nonstructural Weight/Length (2 node elements) or Weight/Area (3 or 4 node elements) for elements that are not resized.

Input to Processor TAB, Subprocessor NSW, but does not contain the NSW card.

5.3.6 $$DCVDM - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static and Dynamic

Contents: Design Cover Distributed Weights - (data only)

Data Description: Nonstructural Weight/Area for cover elements to be resized.

Free Field - Typical Format (I10, F10.5)

Col. 1-10 Entry Number - Referenced by Processor ELD

Col. 11-20 Weight/Area - Units must be consistent with Material Densities

Input to SPAR Processor TAB, Subprocessor MATC.

5.3.7 <u>$$DCVPR</u> - DATA BLOCK

<u>Input for Program(s)</u>: SPAR

<u>Type of Analysis</u>: Static and Dynamic

<u>Contents</u>: Design Cover Properties - (data only)

<u>Data Description</u>: Input to SPAR Processor TAB, Subprocessor SHELL SECTION PROPERTIES (SA) for cover elements to be resized.

Does not contain Subprocessor call statement (the SA card).

DATA BLOCK DESCRIPTIONS

5.3.8 $$DESGN - DATA BLOCK

Input for Program(s): *TWO, *CGMASS, *TRIM

Type of Analysis: Static or Dynamic

Contents: Namelist $DCON

Data Description:

Variable	Type	Description
MACH(14)	REAL	Array of Mach Nos., one for each load case.
ALT(14)	REAL	Array of altitudes, one for each load case.
JNT(2)	INT	Array; JNT(1) is first joint to be printed, JNT(2) is last joint to be printed for displacements.
JRT(2)	INT	Array; JRT(1) is first joint to have static reactions printed. JRT(2) is last joint to have reactions printed.
FLAP(200,5)	REAL	Array of Flap Deflections. Flap (panel no., flap case number) = DEFL.(deg.) + down - up
NFCASE(14)	INT	Array; NFCASE (Load Case NO.) = Flap Case No.
VGUST(14)	REAL	Array; VGUST (Load Case No.) = Vertical Gust (ft/sec).
MFLAG(14)	REAL	Array of Mach Nos. to be re-calculated (if a Mach no. already exists, it will not be re-calculated).
LPLOT(14)	INT	Array; LPLOT (Load Case No.) = 1 for spanwise loading plots from program trim (default is no plotting).
RELAX	REAL	Fraction of new computed loads to be used (default = 1).

5.3.8 $$DESGN - DATA BLOCK (continued)

Variable	Type	Description
LFLAG	INT	Flag for turning off symmetric pull-up loading option (LFLAG = 0, off default = 1, on).
NDOF	INT	No. of degrees of freedom for SPAR (default = 6).
NDTAIL	INT	Node No. where tail force is applied for trim.
SPAN	REAL	Semi-span of wing. If not specified, semi-span = max y-value from grid deck.
SREF	REAL	Reference wing area for lift coefficient.
CBAR	REAL	Reference wing mean aerodynamic chord (defaults are 1.E + 6 and 1.E + 3, respectively).

5.3.9 $$DGMFL - DATA BLOCK

Input for Program(s): DGM

Type of Analysis: Dynamic

Contents: Number of mode shapes desired and damping coefficients, where the number of mode shapes must be less than or equal to the "A" parameter set by the user in Task Procedure DYNAM.

Data Description:

	Description	Format	Type
DATA1	No. of mode shapes	Free Field	INT
	Damping coefficient for each mode (one value will be applied to all)	Free Field	REAL

5.3.10 $$DLNFL - DATA BLOCK

Input for Program(s): DLIN

Type of Analysis: Dynamic

Contents: ISAC TAPE3 data set, ref. NASA TM-80040

Data Description:

Card No.	Description	Format	Starting Column
1	Header	8A10	1
2	HVIB	Free Field	1
3	$DLINPT	NAMELIST	
4	$MODEPLT	NAMELIST (Repeat for each set of mode shapes)	
5	DCM, I/O parameters	Free Field	1
6	I=1,HSECTNS ISS(1,I),ISS(2,I) IPLATE(I),RO(I), XO(I),YO(I)	Free Field	1
7	I=1,NSECTNA IAS(1,I),IAS(2,I), NSS(I),ID(I)	Free Field	1
8	I=1,NNODES TAB(I,1),TAB(I,2), NODE(I),IDF(1,I), IDF(2,I)	Free Field	1

5.3.11 $$DLTFL - DATA BLOCK

Input for Program(s): DLIN; DLAT

Type of Analysis: Dynamic

Contents: ISAC TAPE5 data set, ref. NASA TM-80040.

Data Description:

	Description	Format	Starting Column
Data 1	HEADER	8A10	Column 2
Data 2	Namelist $$DLATINP$	NAMELIST	Column 2
Data 3	DCM input-output parameters	Free Field	Column 1
(Reads MODE shapes from TAPE9 if IDCM=1, or from TAPE9, Ser 1 if IDCM=0)			
Data 4	I=1,NP		
	(SCAP(J),J=1,4), (YCAP(J),J=1,2)	Free Field	Column 1
	(ZCAP(J),J-1,2)NSPAN, NCHORD,COEF	Free Field	Column 1
	(TH(I),I=1,NCHORD)	Free Field	Column 1
	(TAU(I),I=1,NSPAN)	Free Field	Column 1
Data 5 (if NB=0)	I=1,NB		
	ZSC,YSC,NF,NZ,NY,COEFF MRK(1),MRK(2)	Free Field	Column 1
	(F(I),I=1,NF)	Free Field	Column 1
	(RAD(I),I=1,NF)	Free Field	Column 1
Data 6	(LIM(I,1),LIM(I,2), I=1,(NSTRIP)	Free Field	Column 1
(NADD additional MODE data can be read in here instead of or in addition to that on TAPE9 IF NADD = NMOORS - (NELAST + NR + NC) = 0. NADD IS COMPUTED INTERNALLY.			
Data 7 (if NADD=0)	I-1,NADD		
	(H14(J),J=1,NBOX)	(6F10.0)	Column 1
	(H(J),J=1,NBOX)	(6F10.0)	Column 1
	(DH1(J),J=1,NBOX)	(6F10.0)	Column 1
Data 8 (if NC=0)	I=1,NC		
	NBC,NCST,SWHL,ICTYPE	Free Field	Column 1
	(ICBOX(IK),IK=1K,NBC)	Free Field	Column 1
Data 9 (if NBE=0)	I=1,NMODES (include all MODES shapes)		
	BS(I),I=1,NBOXB		REPEAT FOR EACH OF NBE BODIES
	H(I),I=1,NBOXB		
	DH1(I),I=1,NBOXB		
	DH2(I),I=1,NBOXB		

5.3.12 $$DSWDM - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static and Dynamic

Contents: Design Shear Web Distributed Weights - (data only)

Data Description: Nonstructural Weight/Area for shear web elements to be resized.

Free Field - Typical Format (I10, F10.5)

Col. 1-10 Entry Number

Col. 11-21 Weight/Area

Input to SPAR Processor TAB, Subprocessor NSW but does not contain the NSW card.

5.3.13 $$DSWPR - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static and Dynamic

Contents: Design Shear Web Properties - (data only)

Data Description: Input to SPAR Processor TAB, Subprocessor PANEL SECTION PROPERTIES (SB) for shear web elements to be resized.

Does not contain Subprocessor call statement (the SB card).

5.3.14 $$DYNAL - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Dynamic

Contents: Unique Input for Dynamic Analysis

Data Description: Contains the following cards

```
[XQT AUS
  DEFINE DM = 1 RMAS BTAB
  DEFINE CM = 1 DEM DIAG
  PM = SUM(FUEL,DM)
  M+RM = SUM( c CM, d PM)
  K+LM = SUM(K, -e M+RM)
[XQT INV
  RESET K = K+LM
[XQT EIG
  RESET K = K+LM, M = M+RM, INIT = f, NDYN = 9,
                            NREQ = h, SHIFT = e
[XQT DCU
  XCOPY 1,10 VIBR EVAL
  XCOPY 1,11 VIBR MODE
```

where:

Symbol	Description
c	factor to convert the weight to distributed mass.
d	factor to convert the weight to rigid lumped mass.
e	eigenvalue shift if rigid body modes are present (e = 39.5 corresponds to 1 Hertz).
f	number of initial modes (default = 15).
g	max number of iterations (default = 7).
h	number of converged modes required (default = 12).

NOTE: If the structure is statically stable, the 5th and 7th cards and the K=K+LM and SHIFT=e RESET cards in EIG are omitted.

5.3.15 $$DYNAR - DATA BLOCK

Input for Program(s): DYNAR

Type of Analysis: Dynamic

Contents: ISAC TAPE2 data set.

Data Description:

	Description	Format
Data 1	NOCASES	Free Field
Data 2	HEADER	8A10
Data 3	$ INPUT . . . $	NAMELIST
Data 4 (if IDCM=1)	DCM parameters	Free Field
Data 5 (if ISENSE=1)	$ SENLOC . . . $	NAMELIST
Data 6 (if ICSPREAD≠0)	(CS(J,I),J=1,NM-Nc),I=NS)	Free Field
Data 7 (if ICONSYS=1 and if KASE*>1, ICHANG1=1)	$ CONSYM . . . $	NAMELIST
Data 8 (if ICONSYS=1 and if KCASE>1, ICHANG2=1)	I=1,NC J=1,NS $ FILTIN . . . $	NAMELIST
Data 9 (if ICONSYS=1 and if KASE>1, ICHANG3=1)	$ ACTINP . . . $	NAMELIST
Data 10 (if ISELECT=1)	$ SELECT . . . $	NAMELIST
Data 11 (if any PLOT options are on)	$ PLTSEL . . . $	NAMELIST

5.3.16 <u>$$ELDSW</u> - DATA BLOCK

<u>Input for Program(s)</u>: SPAR

<u>Type of Analysis</u>: Static and Dynamic

<u>Contents</u>: Element definition data (continuation of $$STDEF Data Block).

<u>Data Description</u>:

```
E44 $ Main wing shear panels to be resized.
NSECT=1; NMAT=45; NNSW=1; 275  300  302  277
NSECT=2; NMAT=45; NNSW=2; 300  322  324  302
  .        .        .      .    .    .    .
  .        .        .      .    .    .    .
  .        .        .      .    .    .    .

GROUP2 $ Contains elements not to be resized.
$ SHEAR PANELS AT ROOT BC AND WHEEL WELL COUTOUT, NOT RESIZED.
                          .
                          .
                          .
$ LE AND TE SHEAR PANELS
                          .
                          .
                          .
$ FUSELAGE SHEAR PANELS
                          .
                          .
                          .
$ WING TO FUSELAGE TIE SHEAR PANELS
                          .
                          .
                          .
$ HT TO FUSELAGE TIE SHEAR PANELS.
                          .
                          .
                          .
```

5.3.17 $$ENGDM - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static and Dynamic

Contents: Engine Distributed Masses - (Data Only)

Data Description: Nonstructural Weight/Length for engine elements.

Input to SPAR Processor TAB, Subprocessor NSW but does not contain NSW card.

5.3.18 $$FDIN - DATA BLOCK

Input for Program(s): FLTDET, CMACH

Type of Analysis: Flutter

Contents: Flutter analysis and plot parameters

Data Description: Namelist $NAME1

Name	Default	Description
IGAIN	40	Number of interpolated points between reduced frequencies.
NMP	0	Number of modes--if set to "NMOD" will be set to "B" parameter in FLTWT call.
REFMASS	12.	Reference mass.
REFFREQ	10.	Reference frequency.
GMASS(I)	12.	Generalized mass for each mode
G(I)	02.	Damping coefficients for each mode
ITER	2	Number of iterations in solution
NA	3	Number of altitudes
MPS	T	If "T", takes one step toward matched point
CONV(1)	1.	Gen. aero. force conversion factor (to metric)
CONV(2)	1.	Altitude conversion factor (to metric)
CONV(3)	1.	Density conversion factor (to metric)
CONV(4)	1.	Speed of sound conversion factor (to metric)

Plot scale Parameters for V-g, V-ω plots

GMIN	-.12	Minimum damping

5.3.18 $$FDIN - DATA BLOCK (continued)

Name	Default	Description
GMAX	.12	Maximum damping
YSPAN	8.	Length of damping axis in inches
GTK	.1	Damping tick mark increment
FMIN	0.	Minimum Flutter Parameter
FMAX	40.	Maximum Flutter Parameter
XSPAN	10.	Length of flutter parameter axis in inches
FTK	1.	Flutter parameter tick mark increment
FREQMIN	0.	Minimum frequency (ω)
FREQMAX	40.	Maximum frequency
ALT(1)	-8200.	Altitude (meters)
ALT(2)	0.	Altitude (meters)
ALT(3)	14200.	Altitude (meters)

5.3.19 $$FUELM* - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Dynamic (optional for static)

Contents: Fuel Masses (or Weights) - (data only)

Data Description: Input to SPAR Processor TAB, Subprocessor RMASS for fuel masses - Units must be consistent with $$CLMSS.

Does not contain Subprocessor call statement (RMASS card).

Used to compute maneuver design gross weight and center of gravity in CGMASS if present in a static run.

* This data block is generated by CGMASS in Task Procedure MMINRT. Only the data block name is necessary on the local data base.

5.3.20 $$GEOM - DATA BLOCK

Input for Program(s): ONE

Type of Analysis: STATIC

Contents: Description of aerodynamic paneling

Data Description:

Card No.	Field	Type	Description
1	1-5	INT	No. of major wing panels
	6-10	INT	No. of chordwise strips
	11-20	REAL	Fraction of chord for aerodynamic control point on each panel.
	21-50	BLANK FIELD	
	51-60	ALPHA	Type of panel distribution
		OPTIONS ARE:	EVEN -- Even Panel spacing chordwise and spanwise. FREE -- User Supplied spacing.

NOTE: If type of panel distribution=FREE use card 1a also.

Card No.	Field	Type	Description
1a	Free Field	REAL	Fraction of chord for panel spacing (no. of chordwise strips + 1)
2	Free Field	REAL	X - leading edge root
	Free Field	REAL	X - leading edge root
	Free Field	REAL	Y - root
	Free Field	REAL	X - leading edge tip
	Free Field	REAL	X - trailing edge tip
	Free Field	REAL	Y - tip
	Free Field	REAL	Z - root
	Free Field	REAL	Z - tip
	Free	INIT	No. of spanwise strips in this major panel.

5.3.20 $$GEOM - DATA BLOCK (continued)

NOTE: If type of panel distribution=FREE use card 2a also.

2a	Free Field	REAL	Fraction of span for panel spacing (no. of spanwise strips + 1)

NOTE: If not the last major panel and type OF PANEL DISTRIBUTION=FREE, also use card 2b.

Card No.	Field	Type	Description
2b	Free Field	INT	No. of chordwise strips for next major panel
	Free Field	REAL	Fraction of chord for panel spacing (no. of chordwise strips + 1)

NOTE: Repeat card(s) 2. (2a and 2b) for each major panel.

3	Free Field	INT	No. of modes to be plotted. (≤2 x No. of unique Mach Numbers).

5.3.21 $$GSPIN - DATA BLOCK

Input for Program(s): GSPARS

Type of Analysis: Static Resize

Contents: Instructions for SPAR and GSPAR Processors to form a set of data for each E31, E41, or E44 element to be resized. This data set contains geometry and stresses for each load case.

Data Description: This data block is generated by program *TRANP.

5.3.22 $$GSTML - DATA BLOCK

Input for Program(s): GUST

Type of Analysis: Dynamic

Contents: Multiple loads parameter.

Data Description:

	Description	Format
DATA1	$GSTNML	NAMELIST

5.3.23 $$INLST - DATA BLOCK

Input for Program(s): CGMASS

Type of Analysis: Static or Dynamic

Contents: NAMELIST $INPCG (input for *CGMASS)

Data Description:

Vari	Type	Description
DESFL(14)	REAL	ARRAY; DESFL(Load Case No.) = FUEL Wt.
DESCG(14)	REAL	ARRAY; DESCG(Load Case No.) = Desired A/C Center of Gravity
ICAS(14)	INT	ARRAY ; ICAS(Load Case No.) = 1 - Maneuver Case (Flexible loads computed.) 2 - Cruise Case (Case should be set up to correspond to grid deck.) 3 - Taxi Case (No aerodynamic loads.) 4 - Same as 1, only this case generates input file for dynamic analysis. (Use this for only one case.) 5 - Same as 1, only generates file of fuel masses for use in external programs.
GRV(14)	REAL	ARRAY; GRV(Load Case No.) = Load Factor
DISP(14)	INT	ARRAY; DISP(Load Case No.) = Case No. 0 - No output of fuel distribution 1 - Output fuel distribution (default)

5.3.23 $$INLST - DATA BLOCK (continued)

Variable	Type	Description
TNK(14)	INT	ARRAY; TNK(Load Case No.) = Priority Load Case No.
PRI(15,14)	INT	ARRAY; PRI(Tank No., TNK(I)) = Priority
MA(14)	INT	ARRAY; Main Tank Option MA(Load Case No.) = 0 - Main tank option on (fuel routed through main tank) 1 - no main tank option (Default = 0)
IOP(14)	INT	ARRAY; IOP(Load Case No.) = 0 - Tank volume = computed * FUDG(Load Case No.) 1 - Tank volume input with tank data. (Default = 1)
FUDG(14)	REAL	ARRAY; FUDG(Load Case No.) = Factor multiplied to scale tank volume
ICOD(14)	INT	ARRAY; ICOD(Load Case No.) = 0 - Tank priority remains as is until tank is filled 1 - Priority decreases by one every time fuel is put in tank 2 - Calculate range of possible c.g.'s. No fuel distribution done. (Default = 1)

5.3.23 $$INLST - DATA BLOCK (continued)

Variable	Type	Description
KMOD(14)	INT	ARRAY; KMOD(Load Case No.) = 1 - Fuel distributed by minimum distances from fuel only c.g. (c.g. location with fuel) 0 - Fuel distributed by similar distances from fuel only c.g. (Default = 1)
GFMASS	REAL	Conversion factor from fuel units to force units (Default = 1.0)
NFSURF	INT	Fuel mass surface location = 1 - Upper surface only 2 - Mass split between upper and lower surfaces.

5.3.24 $$MACH - DATA BLOCK

Input for Program(s): CMACH

Type of Analysis: Flutter

Contents: Mach number data

Data Description: One card, free field

MIN, MAX, DELT

Name	Description
MIN	Minimum Mach number
MAX	Maximum Mach number
DELT	Incremental Mach number

Successive flutter analyses are performed starting at MIN, incrementing by DELT, until MAX is reached.

5.3.25 $$MDIN - DATA BLOCK

Input for Program(s): MDPROC

Type of Analysis: Dynamic

Contents: Mode shape data reformatting instructions.

Data Description: Three cards, free field format

Card 1: N, M
Card 2: NDOF, (IDOF(I),I=1,NDOF)
Card 3: MDOF,(KDOF(K)=1,MDOF)

Name	Description
N	Largest node number in SPAR analysis
M	Number of mode shapes
NDOF	Number of degrees of freedom per node present in SPAR generated file
IDOF(I)	Ith degree of freedom (see SPAR START Card in Processor TAB)
MDOF	Number of degrees of freedom to be output
KDOF(I)	Ith degree of freedom to be output

5.3.26 $$MINGB - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static with rod resizing.

Contents: SPAR commands to define the minimum gage values for the rod elements.

Data Description:

```
[XQT AUS
 TABLE(NI=6,NJ=a)
 MINI GAGE
 TRAN(SOURCE=BC)
```

where: a = Number of rod properties defined in TAB subprogram BC.

5.3.27 $$MP1IN - DATA BLOCK

Input for Program(s): MPROC, CMACH

Type of Analysis: Flutter

Contents: Planform and mode processing data for the wing.

Data Description: Formatted cards as follows:

Contents	Format	Conditions
XLR,XTR,XLT,XTT	(4E20.12)	Required
YROOT,YTIP,NL,NT	(2E20.12,2I4)	Required
YL(I),XL(I)	(2E20.12)	NL Cards
YT(I),XT(I)	(2E20.12)	NT Cards
MS,NC,LP,LSYM	(4I4)	Required
NI,ISYM	(2I4)	Required
IPLOT	(I4)	Required
CMIN,CINT,NCON,GRID, PAPER,HN,SPACE	(2F10.2,I10, 4F10,2)	≠ 0
MODES	(I4)	Required

Name	Description
XLR	X-coordinate of leading edge root
XTR	X-coordinate of trailing edge root
XLT	X-coordinate of leading edge tip
XTT	X-coordinate of trailing edge tip
YROOT	Y-coordinate of root
YTIP	Y-coordinate of tip
NL	Number of leading edge breaks
NT	Number of trailing edge breaks
YL	Y-coordinate of leading edge breaks
XL	X-coordinate of leading edge breaks
YT	Y-coordinate of trailing edge breaks
XT	X-ccordinate of trailing edge breaks

5.3.27 $$MP1IN - DATA BLOCK (continued)

Name	Description
MS	Number of spanwise pressure functions
NC	Number of chordwise pressure functions
LP	Number of downwash collocation points If = 0, MS*NC Gaussian points are used.
LSYM	If = 1, force quadrature coefficients are computed for antisymmetric loading
NI	Number of input points
ISYM	= 1, antisymmetric spline = 2, symmetric spline = 3, nonsymmetric spline
IPLOT	= 1, modes will be plotted = 0, no mode will be plotted.

Plot Parameters

CMIN	Minimum contour in normalized z units
CINT	Contour interval in normalized z units
NCON	Number of contours
GRID	Grid size in plotter inches
PAPER	Semispan length in plotter inches
HN	Number height in plotter inches
SPACE	Frame Advance in plotter inches
MODES	Number of modes, if set to "NMOD" will be replaced by "B" parameter from FLTWT call.

5.3.28 $$MP2IN - DATA BLOCK

Input for Program(s): MPROC, CMACH

Type of Analysis: Flutter

Contents: Planform and mode processing data for the tail

Data Description: Same as $$MP1IN.

This block should be present in LBASE only if the tail aerodynamics are to be included in the flutter analysis.

5.3.29 $$MSIN - DATA BLOCK

Input for Program(s): MODSEP

Type of Analysis: Flutter

Contents: Node selection data for spline interpolation

Data Description:

Card(s)	Contents	Format
1	NDEFL,NMOD	Free Field
2	NW,ICODE	(I5,R1)
3	JGP(I),I=1,NW	Free Field (as many cards as required)
4	MOJOID	(8A10)

Cards 2-4 are repeated for each component (wing, horizontal tail, wing fin, etc.)

Name	Description
NDEFL	Number of nodes in SPAR output
NMOD	Number of modes, if set to "NMOD", it will be set to the "A" parameter in the DYNAM call
NW	Number of points to be output for that component
ICODE	Deflection component desired - x, y, or z
JGP(I)	Ith joint number to be output
MOJOID	Component Identification (title)

5.3.30 $$OPIN - DATA BLOCK (metal)

Input for Program(s): RESIZE, OPCOM (METAL)

Type of Analysis: Metal strength sizing

Contents: Program control parameters and mechanical properties

Data Description: Namelist $CONIN

Name	Description
ITYOCO	Type of construction (= 1 for metal sandwich skin)
NDV	No. of design variables (= 1→2 for face sheet thickness (t) and core depth (h))
NCON	No. of constraints
NCASE	No. of load cases
NSTRES	No. of stresses input per load case (= 3 for σx, σy, and σxy)
SF	Safety factor applied to stresses

Data Description: Namelist $OPIN

Name	Description
XRHO	Shear web element density
RHOS	Skin face sheet density
RHOC	Sandwich core density
CR	Geometry constraint factor (CR = h/t) ratio (0 → 1)
XE	Young's modulus
XMU	Poisson's ratio
SALLOW	Fatigue limit stress allowable
SIGULT	Ultimate stress allowable
SIGYLD	Yield stress allowable
TAUYLD	Shear stress allowable

5.3.31 $$OPIN - DATA BLOCK (composite)

Input for Program(s): OPCOM (COMPOSITE)

Type of Analysis: Composite strength sizing

Contents: Program control parameters and mechanical properties

Data Description: Namelist $CONMIN

Name	Description
ITYOCO	Type of construction (= 2 for composite sandwich skin)
NDV	No. of design variables (= 3→6 for 0° ply thickness (t_0), ±ϕ° ply thickness ($t_{\pm\phi}$), 90° ply thickness (t_{90}), core depth (h), orientation of axes of orthotropy (γ), and ply orientation angle (ϕ).
NCON	No. of constraints
NCASE	No. of load cases
NLOADS	No. of loads per load case (= 3 for Nx, Ny, and Nxy)
SF	Safety factor applied to loads

Data Description: Namelist $OPIN

Name	Description
XRHO	Shear web element density
RHOS	Skin face sheet density
RHOC	Sandwich core density
CR	Geometry constraint factor (CR = h/t) ratio (0→1)
XNU12	Poisson's ratio for face sheets
SIGALLC	Longitudinal compression allowable stress
SIGALLT	Longitudinal tension allowable stress
SIGALTC	Transverse compression allowable stress
SIGALTT	Transverse tension allowable stress
SIGALTS	Inplane shear allowable stress
NLAM	No. of ply orientations
STALLC(1)	Longitudinal compression allowable strain - Cruise
STALLC(2)	Longitudinal compression allowable strain - Maneuver
STALLC(3)	Longitudinal compression allowable strain - Taxi
STALLT(1)	Longitudinal tension allowable strain - Cruise
STALLT(2)	Longitudinal tension allowable strain - Maneuver
STALLT(3)	Longitudinal tension allowable strain - Taxi
STALS(1)	Inplane shear allowable strain - Cruise
STALS(2)	Inplane shear allowable strain - Maneuver
STALS(3)	Inplane shear allowable strain - taxi
E11	Young's modulus in fiber direction
E22	Young's modulus in tranverse direction
G12	Shear modulus

5.3.31 $$OPIN - DATA BLOCK (composite) (continued)

Name	Description
XE	Effective Young's modulus for shear webs
XMU	Poisson's ratio for shear webs
SIGYLD	Yield stress allowable for shear webs
TAUYLD	Shear stress allowable for shear webs
SALLOW	Fatigue stress allowable
IJD	Integer controlling computation of bending stiffness for buckling (= 1 to compute)
CC(1)	Knockdown factor for compression stress - Cruise
CC(2)	Knockdown factor for compression stress - Maneuver
CC(3)	Knockdown factor for compression stress - Taxi
CCT(1)	Knockdown factor for tension stress - Cruise
CCT(2)	Knockdown factor for tension stress - Maneuver
CCT(3)	Knockdown factor for tension stress - Taxi

C-2

5.3.32 $$PRERR - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static

Contents: SPAR commands for calculation and optional printing of rod (E23) stresses.

Data Description:

```
              [XQT GSF
                RESET SET = 1
                 E23
              [XQT GSF
                RESET SET = 2
                 E23
                 .
                 .
                 .
              [XQT PSF
OPTIONAL        RESET SET = 1
                 E23
                 .
                 .
                 .
```

5.3.33 $$RODPR - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static and Dynamic

Contents: Reference frame and 2 node element property TAB subprocessor input data as required for the following subprocessors.

ALTERNATE REFERENCE FRAMES (ALTREF)
JOINT REFERENCE FRAMES (JREF)
BEAM ORIENTATION (MREF)
BEAM RIGID LINKS (BRL)
E21 SECTION PROPERTIES (BA)
BEAM S6x6 (BB)
E23 SECTION PROPERTIES (BC)
E24 SECTION PROPERTIES (BD)

5.3.34 $$RRDEF - DATA BLOCK

Input for Program(s): GSPAR

Type of Analysis: Static with rod resizing

Contents: Commands for selection and resizing of rods; also SPAR Commands for writing BC BTAB to library SPARLQ.

Data Description:

```
[XQT RSRD
    .
    .
    .
(See TASK PROCESSOR RSRD)

[XQT DCU
  TOC  1
 COPY 1,17 BC BTAB
 TOC 17
```

5.3.35 $$SK1IN - DATA BLOCK

Input for Program(s): SUBKRN, CMACH

Type of Analysis: Flutter

Contents: Reduced frequency data for Subsonic Kernel Function unsteady aerodynamics analysis

Data Description: Cards 1 and 2: Header information

Namelist $NAM1

Name	Description
MACH	Mach number
NK	Number of reduced frequencies
KMAX	Maximum reduced frequency
KMED	Median reduced frequency
KMIN	Minimum reduced frequency
NM	Number of modes, if set to "NMOD" it will be set to the "B" parameter in the FLTWT call

5.2.36 $$SPLT - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static or Dynamic

Contents: Plot specifications and control statements

Data Description: Input for SPAR processors PLTA and PLTB, to generate finite element model plots. Specifications and control statements are generated.

```
[XQT PLTA
   .
   .
   .
   .
[XQT PLTB
   .
   .
   .
```

5.3.37 $$STANL - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static

Contents: Information Unique to the Initial Static Analysis

Data Description: Contains the following cards

```
   RIGD = RIGID
   M+RM = SUM(a CM, b DM)
[XQT DCU
   XCOPY 1,6 M+RM
[XQT INV
```

a and b are factors to convert the distributed mass and the rigid lumped mass respectively to weights. If either set of mass data was input as weight, the corresponding factor may be omitted.

5.3.38 $$STCON - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static

Contents: Constraint Data Unique to Static Analysis

Data Description: Input data for Processor TAB, Subprocessor CONSTRAINT DEFINITION for static constraints required to stabilize free body analysis.

Does not contain Subprocessor call statement (CON).

5.3.39 $$STDEF - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static and Dynamic

Contents: Element Definition Data and Processor Calls Common to Static and Dynamic Analyses

Data Description:

```
[XQT ELD
  .
  . (Input to Processor ELD*)
  .
  .
[XQT TOPO
[XQT E
  RESET MWARP=1.0 (if necessary)
[XQT EKS
[XQT K
[XQT AUS
  DEFINE DM= 1 RMAS BTAB
  DEFINE CM= 1 DEM DIAG
```

* For E31, E41, and E44 elements, the elements to be resized should be in GROUP 1 and defined first within each type. The elements in each type not to be resized should be in GROUP 2 or above.

5.3.40 $$SWCON - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static and Dynamic

Contents: Constant shear web properties and joint elimination and constraint data

Data Description: Contains the following cards

.
.
. (Input to SPAR Processor TAB, Subprocessor PANEL SECTION PROPERTIES (SB)
. for shear webs not to be resized)
.
.

JOINT ELIMINATION SEQUENCE (JSEQ)*

.
. (JSEQ Data - If desired)
.
.

CONSTRAINT DEFINITION 1 (CON=1)

.
.
. (Constraint data common to both static and dynamic analysis)
.
.

* Optional

5.3.41 $$TANK - DATA BLOCK

Input for Program(s): CGMASS

Type of Analysis: Static or Dynamic

Contents: Fuel tank definitions, volumes, and maximum fuel masses per tank

Data Description: Fuel tank definitions

FORMAT (A8, 16X, 4I8)

Col	Name	Description
1-8	WORD	= CTRMEM, triangular element = CQDMEM, quadrilateral element = ENDTANK, terminates tank = ENDRUN, terminates data
25-32	N1	Grid point numbers (see $$CGRID)
33-40	N2	for element corners, numbered
41-48	N3	clockwise
49-56	N4	Not needed for triangles

NOTE: Each lower surface element must follow its upper surface element.

For ENDTANK cards: FORMAT (20X, 3A10, 2F10.2)

Col	Name	Description
21-50	TITLE	Title defining the preceding tank definitions (i.e., INBOARD MAIN TANK).
51-60	INPVTOT	fuel tank volume in IN^3.
61-70	FUELTOT	maximum fuel mass for this tank in lb mass.

For ENDRUN cards: Columns 73-80 are blank.

5.3.42 <u>$$UPROD</u> - DATA BLOCK

<u>Input for Program(s)</u>: SPAR

<u>Type of Analysis</u>: Static

<u>Contents</u>: SPAR Commands which update the rod element properties (BC BTAB) by reading a new BC BTAB from SPARLQ to library 1, SPARLA.

<u>Data Description</u>:

```
[XQT DCU                       data set is copied
  COPY 17, 1 BC BTAB           from SPARLQ to SPARLA
  TOC 1         lists all data sets on SPARLA including
                the new, updated BC BTAB.
```

5.3.43 $$XQTAB - DATA BLOCK

Input for Program(s): SPAR

Type of Analysis: Static and Dynamic

Contents: First Block of Data for TAB Processor with Material Constants

Data Description: Contains the following cards

[XQT TAB
ONLINE = 0 (if desired)
START . . .
MATERIAL CONSTANTS (or MATC)
.
. (Material Data)
.
.

6. DATA PROCESSORS

6.1 DESCRIPTION OF THE DATA PROCESSORS

The Data Processors are NOS Control Card Procedure Files used primarily to generate input data decks for the individual Programs. These files are called by an ISSYS Utility Procedure (LBEDT) which, in turn, is called by Task Procedures.

The Data Processors use the GTR and PACK NOS Control Statements and the XEDIT Text Editor to manipulate input Data Blocks (those starting with $$) and Local Data Files, adding actual card images where required, to form data decks. The make-up of the resulting data decks, the Program which uses each of them, and the calling Task Procedure are indicated in the following pages for each Data Processor.

The Input Data Blocks, Local Data Files, Utility Procedures Programs, and Task Procedures are discussed in separate sections.

6.2 ALPHABETICAL INDEX OF DATA PROCESSORS

6.3 DATA PROCESSOR DESCRIPTION

ISSYS Data Processor functions are described. The computer program(s), for which data are prepared, along with the Task Procedure which calls the Data Processor are given for cross reference. The type of analysis (static or dynamic) followed by the makeup of the resulting data deck are also presented.

Data Processors are typically called from Task Procedures to build Data Blocks to perform an engineering analysis.

6.3.1 CHGDYN1 - DATA PROCESSOR

Prepares Input for Program: SPAR

Called from Procedure File(s): DYNAM

Type of Analysis: Dynamic

Makeup of Resulting Data Deck:

```
  $$XQTAB
* NSW
  DMASSS
  DMASST
  $$ENGDM
  $$CONDM
* JLOC
  $$CGRID
  $$RODPR
* SA(1)
  DPROPT
  $$CCVPR
* SB
  DPROPS
  $$SWCON
  $$CLMSS
  $$STDEF
* GMAS = SUM(CM,DM)
* [XQT DCU
* XCOPY 1,6 GMAS
```

* Actual Card Image Added.

6.3.2 CHGDYN2 - DATA PROCESSOR

Prepares Input for Program: SPAR

Called from Procedure File(s): DYNAM

Type of Analysis: Dynamic

Makeup of Resulting Data Deck:

$$DYANL
SYSFUEL

#File SYSFUEL read in middle of $$DYANL file.

6.3.3 CHGMMI - DATA PROCESSOR

Prepares Input for Program: CGMASS

Called from Procedure File(s): MMINRT

Type of Analysis: Static or dynamic

Makeup of Resulting Data Deck:

File DIN	File DINT
$$CGRID	$$TANK
* 0, 0, 0, 0	
$$INLST	
$$DESGN	

* Actual Card Image Added.

6.3.4 CHGPG1 - DATA PRCCESSOR

Prepares Input for Program: PROG1

Called from Procedure File(s): SETUP

Type of Analysis: Static and Dynamic

Makeup of Resulting Data Deck: $$GEOM
$$CGRID
$$AEROS

6.3.5 CHGSETP - DATA PROCESSOR

Prepares Input for Program: SPAR

Called from Procedure File(s): SETUP, SETPR

Type of Analysis: Static and Dynamic

Makeup of Resulting Data Deck: $$DSWDM
$$DCVDM
$$DCVPR
$$DWSPR

6.3.6 CHGTRIM - DATA PROCESSOR

Prepares Input for Program: TRIM

Called from Procedure File(s): LLOOP

Type of Analysis: Static

Makeup of Resulting Data Deck:

File DIN	File DINT
$$DESGN	$$CGRID

6.3.7 CHG59 - DATA PROCESSOR

Prepares Input for Program: SPAR

Called from Procedure File(s): SPARI (from INIT, INITC, TOTL, TOTLC)

Type of Analysis: Static

Makeup of Resulting Data Deck:

```
  $$XQTAB
* NSW
  DMASSS
  $$ENGDM
  $$CONDM
* JLOC
  $$CGRID
  $$RODPR
* SA(1)
  DPROPT
  $$CCVPR
* SB
  DPROPS
  $$SWCON
  $$STCON
  $$CLMSS
  $$STDEF
  $$STANL
```

* Actual Card Image Added.

6.3.8 DGMDAT - DATA PROCESSOR

Prepares Input for Program: DGM

Called from Procedure File(s): DYNAR

Type of Analysis: Dynamic

Makeup of Resulting Data Deck: $$DGMFL

6.3.9 DLNDAT - DATA PROCESSOR

Prepares Input for Program: DLIN

Called from Procedure File(s): DLIN

Type of Analysis: Dynamic

Makeup of Resulting Data Deck: $$DLNFL
$$DLTFL

6.3.10 DLTDAT - DATA PROCESSOR

Prepares Input for Program: DLAT

Called from Procedure File(s): DLAT

Type of Analysis: Dynamic

Makeup of Resulting Data Deck: $$DLTFL

6.3.11 DYNDAT - DATA PROCESSOR

Prepares Input for Program: DYNARES

Called from Procedure File(s): DYNAR

Type of Analysis: Dynamic

Makeup of Resulting Data Deck: $$DYNAR

6.3.12 FDIN - DATA PROCESSOR

Prepares Input for Program: FLTDET, CMACH

Called from Procedure File(s): None (see Data Processor SKMACH)

Type of Analysis: Flutter

Makeup of Resulting Data Deck:

```
  $$FDIN
* FREQS (inserted into $$FDIN)
```

All "NMOD" strings are XEDITed into the "A" parameter, where "A" is the number of modes to be used in the analysis.

XEDIT is used to change all "NMOD" strings to the value of the A parameter:

```
C/NMOD/A/
```

* Actual Card Image Added.

6.3.13 GSPIN - DATA PROCESSOR

Prepares Input for Program: GSPAR

Called from Procedure File(s): FINL

Type of Analysis: Static

Makeup of Resulting Data Deck:

$$GSPIN (generated by Program *TRANP)

6.3.14 MDIN - DATA PROCESSOR

Prepares Input for Program: MDPROC

Called from Procedure File(s): DYNAM

Type of Analysis: Dynamic (flutter or gust)

Makeup of Resulting Data Deck:

$$MDIN

All "NMOD" strings are XEDITed into the "A" parameter from the DYNAM call, where "A" is the number of modes to be read from the SPAR file. "A" must be .LE. the number of rigid plus flexible SPAR modes.

6.3.15 MP1IN - DATA PROCESSOR

Prepares Input for Program: MPROC

Called from Procedure File(s): None (see Data Processor SKMACH)

Type of Analysis: Flutter

Makeup of Resulting Data Deck:

$$MP1IN

All "NMOD" strings are XEDITed into the "A" parameter from the SETSK call (in FLTWT), where "A" is the number of modes to be interpolated and plotted.

6.3.16 MP2IN - DATA PROCESSOR

Prepares Input for Program: MPROC

Called from Procedure File(s): None (see Data Processor SKMACH)

Type of Analysis: Flutter

Makeup of Resulting Data Deck: $$MP2IN

(See MP1IN)

6.3.17 MSIN - DATA PROCESSOR

Prepares Input for Program: MODSEP

Called from Procedure File(s): DYNAM

Type of Analysis: Dynamic (flutter)

Makeup of Resulting Data Deck:

```
  $$MSIN
  $$CGRID
* 0,0,0,0          (inserted after 2nd $$MSIN Card)
```

"All "NMOD" strings are XEDITed into the "A" parameter from the DYNAM call, where "A" is the number of modes to be used.

* Actual Card Image Added.

6.3.18 OPIN - DATA PROCESSOR

Prepares Input for Program: GUSTB, OPCOM, RESIZE

Called from Procedure File(s): RSZG

Type of Analysis: Static

Makeup of Resulting Data Deck: $$OPIN
$$GSTML

6.3.19 PG2IN - DATA PROCESSOR

Prepares Input for Program: TWO

Called from Procedure File(s): SETUP

Type of Analysis: Static

Makeup of Resulting Data Deck: $$PG2IN
-EOF-
$$DESGN

6.3.20 PRERR - DATA PROCESSOR

Prepares Input for Program: SPAR

Called from Procedure File(s): RESE23

Type of Analysis: Static with rod resizing.

Makeup of Resulting Data Deck: $$PRERR

6.3.21 RRDEF - DATA PROCESSOR

Prepares Input for Program: GSPAR

Called from Procedure File(s): RESE23

Type of Analysis: Static with rod resizing

Makeup of Resulting Data Deck: $$RRDEF

6.3.22 SKMACH - DATA PROCESSOR

Prepares Input for Program: CMACH

Called from Procedure File(s): FLTSK

Type of Analysis: Flutter

Makeup of Resulting Data Deck:

```
  $$MACH
  $$SK1IN
  $$MP1IN
  $$FDIN
* FREQS    (inserted into $$FDIN)
  $$MP2IN
```

All "NMOD" strings are XEDITed into the "A" parameter from the FLTSK call, where "A" is the number of modes to be used.

* Actual Card Image Added.

6.3.23 SK1IN - DATA PROCESSOR

Prepares Input for Program: SUBKRN

Called from Procedure File(s): None (see Data Processor SKMACH)

Type of Analysis: Unsteady aerodynamics

Makeup of Resulting Data Deck:

$$SK1IN

All "NMOD" strings are XEDITed into the "A" parameter from the DYNAM call, where "A" is the number of modes to be used.

6.3.24 SPLOT - DATA PROCESSOR

Prepares Input for Program: SPAR

Called from Procedure File(s): SPARPLT

Type of Analysis: Static or dynamic

Makeup of Resulting Data Deck:

```
[XQT PLTA
 (Specifications for plotting)

[XQT PLTB
 (Plot and control statements)

[XQT EXIT
```

7. PROGRAMS

7.1 INTRODUCTION

Calculations are performed by Programs written primarily in FORTRAN. Some of these Programs were developed specifically for use in ISSYS. Others were written elsewhere but are maintained as part of the ISSYS System. Some Programs, SPAR for example, were written and are maintained outside of the ISSYS System. The source program names prefixed by a * are records in the system and can be accessed using ISSYS Utility Procedures.

7.2 ALPHABETICAL INDEX OF PROGRAMS

7.3 PROGRAM DESCRIPTIONS

ISSYS Programs (computer codes) are described in this section. The function of each program is given. The Task Procedure which executes it, and the Data Processor, or other Program which builds the input file(s) for it are also presented. A brief description of calculations performed and files created or used by the Program is included.

7.3.1 *ADDSK - SOURCE DECK, ADDSK - BINARY

Function: Combine wing and tail generalized aerodynamic forces.

Executed By: FLTSK

Input Prepared By: Program CMACH and SUBKRN

Description: Adds the tail generalized aerodynamic forces, multiplied by the ratio of the tail reference length to the wing reference length (usually one-half the root chord), to the wing generalized aerodynamic forces. The reduced frequencies in the SUBKRN input have already been ratioed by this factor in CMACH.

Files:

ADDIN - Input data
SK1OUT - Wing generalized aerodynamic forces
SK2OUT - Tail generalized aerodynamic forces
ADDOUT - Superimposed generalized aerodynamic forces (Identical to SK1OUT if tail data is not included in the analysis)

NOTE: An ISSYS format source deck may contain the source code for any number of programs and/or subprograms.

7.3.2 *CGMASS - SOURCE DECK, CGMASS - BINARY

Function: To distribute fuel to specified nodes and generate a SPAR type input file for each load case.

Executed by: MMINRT

Input Prepared by: Processor CHGMMI

Description: Uses tank definition to compute center of gravity for each tank. Distributes fuel such that desired c.g. (or as close to desired c.g. as possible) is obtained using all fuel specified.

Files:

- DIN - Input file prepared by Processor CHGMMI.
- SPARLF - SPAR mass file.
- ISX - Data block $$DESGN updated.
- ISY - Scratch file.
- CGDIN - Input file created for Program TRIM.
- SYSFUEL - Fuel file in SPAR format for dynamic analysis.
- DINT - Fuel tank simulation.
- MAS - File of total masses per node for each load case (for Program TRIM).
- FUELM - File containing fuel masses for specified load case (intended for use by external programs).

7.3.3 *CMACH - SOURCE DECK, CMACH - BINARY

Function: Change Mach number and prepare input files.

Executed by: FLTSK

Input Prepared by: SKMACH (which uses FDIN)

Description: Changes the flutter analysis Mach number and geherates input data files for SUBKRN, ADD, and FLTDET.

Files:

- CMIN - Input data.
- SK1 - SUBKRN input data for wing.
- SK2 - SUBKRN input data for tail.
- ADDIN - ADD input data.
- FDIN - FLTDET input data.

7.3.4 *CONMIN - SOURCE DECK, CONMIN - BINARY

Function: A FORTRAN Program for constrained function minimization.

Executed By: Program OPCOM from Procedures TRSC, TRSM

Input Prepared By: Program OPCOM

Description: CONMIN is a FORTRAN program, in subroutine form, for the solution of linear or nonlinear constrained optimization problems.

Reference: CONMIN - A FORTRAN Program for constrained function minimization, Users Manual, NASA TM X-62,282, Ames Research Center.

7.3.5 *COST - SOURCE DECK, COST - BINARY

Function: Generates a dayfile summary of the computer resources used plus the total cost in dollars.

Executed By: User

Input Prepared By: None

Files Required: None

Files Created: None

Description: This routine is always available to ISSYS users. It displays the resources used since the last CHARGE card.

This is a COMPASS routine which uses the GETJA macro from COMCMAC to obtain the job accounting information.

7.3.6 *DYCORE - SOURCE DECK, DYCORE - BINARY

Function: Dynamic core allocation

Called By: PROG2, MPROC, MATRIX

Entry Points: LOCMM

Externals: Misc. Common Memory Manager entry points

Usage:
```
IADDR = LOCMM (NWORDS)
IDELT = LOCMM (NWORDS, ARRAY(1))
    L = LOCMM (0)
```

Description: Uses Common Memory Manager (CMM) to allocate and de-allocate storage dynamically given the number of words (NWORDS > 0) and the starting address for the allocated storage. Optionally, if a second argument is provided LOCMM will return an index-increment. Finally, if NWORDS > 0 then all storage will be released.

Note that LOCMM with NWORDS > 0 can be called as often as needed:

```
e.g.,   IDELTA = LOCMM (NWA, A(1))
        IDELTB = LOCMM (NWB, B(1))
```

7.3.7 DGM - SOURCE DECK, DGMBN - BINARY

Function: Prepares generalized mass matrix and damping coefficients for Program DYNARES (ISAC).

Executed by: Task Procedure DYNAR

Input Prepared By: Data Processor DGMDAT

Description: Matrices are generated based on number of degrees of freedom.

Files:
DIN - Input - prepared by Processor DGMDAT
TAPE7 - Output, damping coefficients
TAPE8 - Output, generalized mass matrix

7.3.8 DLAT - SOURCE DECK, DLATBN - BINARY

Function: Using aerodynamic lifting surface theory known as Doublet Lattice method, computes the generalized aerodynamic forces, ref. NASA TM-80040, ISAC.

Executed By: Task Procedure DLAT

Input Prepared By: Data Processor DLTDAT

Description: Generalized aerodynamic force matrices are generated by DLAT for use in the equations of motion.

Files:

- DIN - Input, prepared by Processor DLTDAT if TAPE5 not assigned.
- TAPE5 - Input, FORTRAN NAMELIST Data (see ref.).
- TAPE9 - Input/Output, ISAC data-complex.
- DLATPLT - Output, plot vector file.

7.3.9 DLIN - SOURCE DECK, DLINBN - BINARY

Function: Aero/Structure interface program module of ISAC, ref. NASA TM-80040.

Executed By: Task Procedure DLIN.

Input Prepared By: Data Processor DLNDAT Task Procedure DYNAR.

Description: Finds a transformation [T], utilizing the surface spline interpolation technique between deflections at structural nodes and deflections and slopes on aerodynamic boxes.

Files:

GSTIN - Input, frequency and loads data.
DIN - Input, file prepared by Processor DLNDAT.
TAPE3 - Input (see ref.)
TAPE5 - Input (see ref.)
TAPE9 - Output, ISAC data-complex.
DLINPLT - Output, plot vector file.

7.3.10 DYNARES - SOURCE DECK, DYNARBN - BINARY

Function: Performs flutter and gust analyses, ref. NSA TM-80040.

Executed By: Task Procedure DYNAR.

Input Prepared By: Data Processor DYNDAT, Program DGM, Task Procedure DYNAR.

Description: Control laws may be treated by direct input of a matrix or transfer functions or by utilizing combinations of built-in filters. Roots are determined by P-R method.

Files:

DIN - Input, prepared by Processor DYNDAT.
TAPE5 - Input; loads, damping coefficients, generalized masses, nodal identification.
TAPE9 - ISAC data-complex.
RMSTR - Output, RMS gust stress file.
DYNPLT - Output, plot vector file.

7.3.11 *FLTDET - SOURCE DECK, FLUTDET - BINARY

Function: Solution of flutter equations.

Executed by: FLTWT

Input Prepared by: FDIN or Program CMACH

Description: Solves the flutter eigenvalue problem for flutter speeds, frequencies, and mode shapes using the generalized aerodynamic force file formed by ADD. Additional program capability includes a matched point search and V-g and V-ω plots.

Files:

DIN - Input data.
ADDOUT - Generalized aerodynamic forces.

7.3.12 GSPARS - SOURCE DECK, GSPAR - BINARY

Function: SPAR stress post-processor

Executed by: RSZG

Input Prepared by: GSPIN

Description: Processes stress data from main SPAR library file SPARLA and puts the stresses on files to be read by OPCOM or RESIZE along with other element data required by these programs.

Files:

DIN - Input data.
SPARLA - Main SPAR library file.
SPARLN - Stress data for triangular elements.
SPARLO - Stress data for quadrilateral elements.
SPARLP - Stress data for shear webs.
TAPE60 - SPARLN for RESIZE.
TAPE63 - SPARLO for RESIZE.
TAPE66 - SPARLP for RESIZE.
TAPE80 - SPARLN for OPCOM.
TAPE83 - SPARLO for OPCOM.

7.3.13 *GETFIT - SOURCE DECK, GETFIT - BINARY

Function: Determines the File Information Table (FIT) address of a given file.

Called By: LOCBA

Entry Point(s): GETFIT

External Ref(s): SYSERR

ENTRY (X1) - FIT pointer word in the form
VFD 1/VAR, 1/FP, 58/ADDRESS

EXIT (X1) - FIT address, if found

Description: Uses the list of file names/addresses located at RA+2 to obtain the FIT address for the desired file. The job is aborted with error number 62_{10} if the file name cannot be found.

This routine is the same as routine GETFIT= on the FORTRAN library.

This routine is not callable from a FORTRAN progran. The entry point and the calling sequences are not compatible.

7.3.14 *GUST - SOURCE DECK, GUSTB - BINARY

Function: Combines gust stresses with steady-state stresses for use in resizing the structure.

Executed by: RSZG

Input Prepared by: GSPAR

Files Required: TAPE50 - steady-state stresses for triangular and quadrilateral elements plus shear webs.
TAPE25 - gust RMS stress files from DYNARES (ISAC).

Files Created: TAPE60 - combined gust and steady-state stresses for triangular and quadrilateral elements.
TAPE66 - combined gust and steady-state stresses for shear webs.

Description: Combines stress file (RMSTR) created by Task Procedure DYNAR with the stress data file (SPARLN) produced by GSPAR to create the stress data file (TAPE66) for resizing.

7.3.15 *LDBUG - SOURCE DECK, LDBUG - BINARY

Function: Set loader parameters for the ISSYS debug mode.

Executed By: The ISSYS LDBUG Procedure causes this routine to be used in place of the LDSET routine.

Externals: None

Entry Points: LDBUG

Files Created:

Description: This routine is currently identical with *LDSET except that CMMSAFE is used instead of CMMFAST and the preset is: PRESETA = NGINF.

7.3.16 *LDSET - SOURCE DECK, LDSET - BINARY

Function: Set loader parameters for ISSYS programs.

Executed By: Loaded with each Mainline Program.

External References: None

Entry Points: LDSET

Files Created:

Description: Uses the COMPASS LDSET instruction to generate internal, binary loader instructions similar to those produced by the NOS LDSET control card.

7.3.17 *LDTEK - SOURCE DECK, LDTEK - BINARY

Function: Set loader parameters for ISSYS Programs for interactive graphics. The LDTEK Procedure causes this routine to be used in place of the LDSET routine.

Externals: None

Entry Points: LDTEK

Files Created:

Description: This routine is indentical with the LDSET routine except that LIBFTEK is used instead of LRCGOSF for graphics subroutines.

7.3.18 *LOCBA - SOURCE DECK, LOCBA - BINARY

Function: Returns the FIT starting address and execution time name for indicated file.

Called By: *READU, *ONE, *MDPROC

Entry Points: LOCBA, XLOCBA

External Refs: GETFIT.

Files Created: None

Description: This is a FORTRAN (FTN) callable link to the system GETFIT= routine. Given the 'logical-unit-number' or the 'left-justified-zero-filled-name', LOCBA will return the corresponding FIT address. Optionally, if LOCBA is called with two arguments, the second argument will be set to the execution time L-J-Z name on return.

```
e.g., L = LOCBA (3)
      L = LOCBA (SLTAPE3, NAMEZ)
```

where L = FIT address and NAMEZ = 'left-justified-zero-filled-name'.

7.3.19 *LOOK - SOURCE DECK, LOOK - BINARY

Function: Plot SAVPLT files on TEKTRONICS 4014 Screen

Executed By: TVPLOT, USER

Input Prepared By: *MPROC, *FLTDET, SPAR, *TRIM

Files Required: INPUT, OUTPUT, SAVPLT

Files Created: TTPE16

Description: Interactively reads and scales SAVPLT files for TEKTRONICS 4014 terminals. User inputs can be provided for baud rate, magnification, and/or X/Y origin shifts. For multi-frame plot files, a frame skipping option is available.

7.3.20 *MATRIX - SOURCE DECK, MATRIX - BINARY

Function: To generate a random access file containing aerodynamic matrices for each Mach number.

Executed by: Task Procedure SETUP

Input Prepared by: Program *TWO and *MPROC

Description:

A. Multiplies spline matrix of structural nodes to aerodynamic panel control points (SPLINE1) times aerodynamic influence coefficients (AIC) times spline matrix of aerodynamic panel centroids to structural nodes (SPLINE2).

B. Multiplies AIC by SPLINE2.

(i.e.) -

A. SPLINE1 x AIC x SPLINE2

B. AIC x SPLINE2

Files:

- ISV - First set of structural nodes.
- ISR - Aerodynamic influence coefficient matrices from program TWO.
- ISX - First set of structural node coordinates.
- ALOAD - Random access file of aerodynamic matrices.
- ISS - First set of SPLINE coefficients.
- ISO - Second set of SPLINE coefficients.

7.3.21 *MDPROC - SOURCE DECK, MDPROC - BINARY

Function: Post-processor for SPAR mode shapes and frequencies.

Executed by: DYNAM

Input Prepared by: MDIN

Description: Reads natural frequencies from a SPAR generated file and writes them out on a separate file in Namelist format. Reads mode shapes from a SPAR generated file, selects the desired degrees of freedom, and puts these degrees of freedom (for each grid point for each mode) on a file.

Files:

- DIN - Input data.
- SPARLJ - Frequencies from SPAR.
- FREQS - Frequencies in Namelist format.
- SPARLK - Mode shapes from SPAR.
- MODES - Mode shapes used by MODSEP.
- GSTIN - Frequencies plus MODES plus ISX plus SPARLL (if it exists).

7.3.22 *MODSEP - SOURCE DECK, MODSEP - BINARY

Function: Separate mode shapes into desired groups.

Executed by: DYNAM

Input Prepared by: MSIN

Description: Separates the modal displacements of the entire structural model into displacements per lifting surface as defined in the input data. Displacements normal to the lifting surface and the location of the points are written on a file to be used by MPROC.

Files:

DIN - Input data.
MODES - Mode shapes from MDPROC.
MSPOUT - Modal displacements per lifting surface.
ISX - Coordinates of grid points used in MSPOUT.

7.3.23 *MPROC - SOURCE DECK, MPROC - BINARY

Function: Modal interpolation.

Executed by: MPRSK

Input Prepared by: MP1IN (wing), MP2IN (tail).

Description: Processes the lifting surface mode shapes by spline interpolation between the structural nodes and the aerodynamic collocation points for use in SUBKRN. If the spline coefficient matrix does not exist, it is generated. Mode shape plots are generated if desired.

Files:

- DIN - Input data.
- MSPOUT - Modal displacements, all lifting surfaces.
- MSWING - Modal displacements for wing.
- MSTAIL - Modal displacements for tail.
- SPL1 - Spline coefficients for wing.
- SPL2 - Spline coefficients for tail.
- MPOUT - Processed modes for SUBKRN.
- MP1OUT - MPOUT for wing.
- MP2OUT - MPOUT for tail.

7.3.24 *ONE - SOURCE DECK, PROG1 - BINARY

Function: To generate input files for programs TWO and MPROC.

Executed by: Task Procedure SETUP.

Input Prepared by: CHGPG1

Description: Generates Woodward-Carmichael input data deck for aerodynamic panel geometry and input decks for both executions of MPROC in Task Procedure SETUP. Generates binary input deck for first MPROC execution in SETUP.

Files:

- DIN - Input file.
- ISZ - Second MPROC execution input file.
- PG2IN - Program two input file.
- ISX - First set of structural node coordinates.
- ISI - First MPROC execution input file.
- ISV - First set of structural nodes.
- AERPLT - Plot vector file.

7.3.25 *OPCOM - SOURCE DECK, OPCOM - BINARY

Function: Resize of composite or metal cover panels using nonlinear programming techniques.

Executed by: TRSC, TRSM

Input Prepared by: OPIN

Description: Uses cover element stress data files from GSPAR to resize composite (executed by TRSC) or metal (executed by TRSM) elements considering stress, minimum gage, and stability constraints and using nonlinear programming techniques. For metal, the design variables are the cover thickness (t) and the core depth (h). For a 0/±α/90-degree composite laminate oriented at γ-degrees, the possible design variables are t_0, t_α, t_{90}, α, γ, h.

Files:

DIN - Input data.
TAPE80 - Stress data for triangular elements.
TAPE83 - Stress data for quadrilateral elements.
TAPE71 - Property data for cover elements.
TAPE72 - Mass data for cover elements.
DPROPT - TAPE71
DMASST - TAPE72
TAPE20 - Output file.
TAPE35 - Output file.
TAPE40 - Design variable values.
TAPE30 - TAPE40 internal to OPCOM.

7.3.26 *READU - SOURCE DECK, READU - BINARY

Function: Reads unformatted SPAR sequential files.

Executed By: *TRIM, *CGMASS, *MDPROC

Entry Points:
```
CALL OPENLL(LFN,NAME)
CALL REWNU(LFN,NAME)
CALL READU(LFN,WSA,NWORDS)
CALL CLOSEU(LFN)
```

External Refs: None

Files Created: None

Description: READU is a FORTRAN callable utility to read a SPAR sequential file where:

- LFN = Logical unit number.
- NAME = Execution tie file name from the execution control card.
- WSA = Output array containing NWORDS.
- NWORDS = Number of words to be read.

NOTES: The second argument (NAME) on the OPENU/REWNU call is optional.

READU does not reposition the file.

Hitting the End Of Information (EOI) is fatal.

7.3.27 *RESIZE - SOURCE DECK, BRESIZE - BINARY

Function: Resize of cover panels and shear webs using simultaneous failure mode techniques.

Executed by: RSZG

Input Prepared by: OPIN

Description: Uses stress data files produced by GSPAR to resize elements considering stress allowable, minimum gage, and stability constraints using simultaneous failure mode techniques. A property file and a non-structural mass file is generated for each type of element. When RSZG is called by TRSC or TRSM, only the shear webs are resized.

Files:

DIN - Input data.
TAPE60 - Stress data for triangular elements.
TAPE63 - Stress data for quadrilateral elements.
TAPE66 - Stress data for shear webs.
TAPE61 - Property data for triangular elements.
TAPE64 - Property data for quadrilateral elements.
TAPE62 - Mass data for triangular elements.
TAPE65 - Mass data for quadrilateral elements.
TAPE67 - Property data for shear webs.
TAPE68 - Mass data for shear webs.
DPROPT - TAPE61 plus TAPE64.
DMASST - TAPE62 plus TAPE65.
DPROPS - TAPE67.
DMASSS - TAPE68.

7.3.28 SPAR - SOURCE DECK, SPAR - BINARY

Function: Finite element structural analysis.

Executed by: SPARI (twice), SPARR (twice), LOOP, FINL, DYNAM (twice).

Input Prepared by: CHG59, CHF1, CHF2, CHF3, CHF4, CHGDYN1, CHGDYN2.

Description: SPAR is a general purpose finite element structural analysis program. It is used to compute deflections, stresses, natural modes and frequencies of vibration, gust stress coefficients, and the structural mass matrix. See the Task Procedure File descriptions for its function at any point in the system.

Files:

SPARLA - Main library.
SPARLF - Mass matrix.
SPARLG - Cruise deflections.
SPARLH - Maneuver deflections.
SPARLJ - Frequencies.
SPARLK - Mode shapes.
SPARLN - Stress data for triangular elements.
SPARLC - Stress data for quadrilateral elements.
SPARLP - Stress data for shear webs.
SPARLT - Restart tape containing SPARLA.
DIN - Input data.
SPARLL - Gust stress coefficients.
DINW - Tape write instructions for restart.

Reference: SPAR Structural Analysis System Reference Manual, NASA CR-145098-1, Engineering Information Systems, Inc., February 1977.

7.3.29 *SUBKRN - SOURCE DECK, SUBKRN - BINARY

Function: Calculates unsteady aerodynamic forces.

Executed by: FLTSK (twice).

Input Prepared by: Program CMACH.

Description: Calculates the subsonic kernel function aerodynamic forces for a set of lifting surface modes that have been processed by MPROC. The forces are calculated at several values of reduced frequencies at a particular Mach number.

Files:
- SK1 - Input data for wing.
- SK2 - Input data for tail.
- MP1OUT - Processed modes for wing.
- MP2OUT - Processed modes for tail.
- SK1OUT - Generalized aerodynamic forces for wing.
- SK2OUT - Generalized aerodynamic forces for tail.

7.3.30 *TRANP - SOURCE DECK, TRANP - BINARY

Function: Generates input data for SPAR and GSPAR processors used in structural resizing and gust analysis.

Executed By: Data processors GSPIN (structural resizing) and CHGDYN2 (gust analysis).

Input Prepared By: GSPIN and CHGDYN2

Description: Depending on the variable ITYPE, either of the following are generated: (1) SPAR commands to calculate modal stresses for a set of vibration modes, extract the appropriate element stresses for resizing and write those to file SPARLL, or 2) GSPAR commands to assemble element geometry data and stresses for each load case used in resizing. These element information sets are written to files SPARLN, SPARLO, SPARLP for the E31, E41, and E44 elements, respectively.

Files: TAPE7 - Contains the SPAR/GSPAR input instructions produced by TRANP.

7.3.31 *TRIM - SOURCE DECK, TRIM - BINARY

Function: Computes aerodynamic load vectors and merges data with output from *CGMASS to form a SPAR input file.

Executed by: LLOOP

Input Prepared by: Processor CHGTRIM and Program *CGMASS.

Description: TRIM uses the steady aerodynamics random access file, ALOAD, and a vector of Z-displacements to compute an aerodynamic load vector for each load case. It also trims the aircraft, can add flaps to the loads or perform a delta-alpha steady-state gust analysis for each load case. This program also creates the jig shape (JGRID) data set.

FILES:

TAPE1 - JGRID (jig shape)
TAPE2 - CGRID (cruise shape)
TAPE3 - SPARLG (cruise deflections)
TAPE4 - SPARLH (maneuver deflections)
TAPE5 - INPUT (NAMELIST DCON)
TAPE6 - OUTPUT
TAPE7 - SPDIN (output SPAR file)
TAPE8 - ALOAD (aerodynamic pressure data)
TAPE9 - CGDIN (SPAR data from CGMASS)
TAPE10 - STDIN (SPAR stress data)
TAPE11 - OZD (old z-displacements)
TAPE12 - NOZD (newly computed OZD file)
TAPE13 - MASS (masses from CGMASS)
TAPE14 - SPARLF (SPAR mass matrix)

7.3.32 *TWO - SOURCE DECK, PROG2 - BINARY

Function: Computes steady aerodynamic data (Woodward-Carmichael program).

Executed by: Task Procedure SETUP

Input Prepared by: Processor PG2IN and Program ONE.

Description: The Woodward-Carmichael Program is a finite element, small perturbation (linear), potential flow aerodynamics Program. It is used to compute steady subsonic or supersonic aerodynamic data.

The aerodynamic influence coefficient matrix is generated for program MATRIX.

Files:

- DIN - Input file.
- ISR - Aerodynamic influence coefficients.
- IST - File containing aerodynamic panel slopes from first MPROC execution.
- ISQ - Delta C_ps' as input for second MPROC execution (for plotting purposes).
- ALOAD - Random access file for aerodynamic matrices.
- AERPLT - Plot vector file (contains aero. panel plots).

References: Woodward, F. A.: "An Improved Method for the Aerodynamic Analysis of Wing-Body-Tail Configurations in Subsonic and Supersonic Flow", Aerophysics Research Corporation, Bellevue, Washington, May 1973.

8. AUXILIARY PROCEDURES

8.1 DESCRIPTION OF ISSYS AUXILIARY PROCEDURES

There are several Procedures within ISSYS that most users will never use directly. These are the Auxiliary Procedures which are used within the ISSYS Task and Utility Procedures to set up required files and programs, print headers, provide alternate data modification capability, and dispose of plot files. They are documented in this section.

8.2 ALPHABETICAL INDEX OF AUXILIARY PROCEDURES

8.3 AUXILIARY PROCEDURE DESCRIPTIONS

Auxiliary procedures are described in this section. Auxiliary Procedures typically perform file manipulation or special output functions. The function of each Procedure is given along with Procedures it is called by and calls made to Procedures. A brief description is also presented. Several Auxiliary Procedure Descriptions are additionally described in the ISSHLP file, Appendix B of this document.

8.3.1 GETLB - AUXILIARY PROCEDURE

Function: Sets up libraries, title page, and headings.

Called By: Procedure ISSYS

Calls Made: TITLE, HDR

Descriptions: CALL(ISSYS(XQ=GETLB)

8.3.2 HDR - AUXILIARY PROCEDURE

Function: Prints a one page ISSYS header for each user called Task Procedure.

Called By: User called Task Procedures.

Calls Made: None

Description: CALL(ISSYS(XQ=HDR,L=XL,A=XA)

where:

XL = File to receive output.

XA = Header for title page.

8.3.3 LBEDT - AUXILIARY PROCEDURE

Function: Calls ISSYS Data Processors. If a local file exists with the same name as an ISSYS Data Processor the local file will be used in place of the data file normally created by the Data Processor.

Called By: ISSYS Task Procedures

Calls Made: None

Description: CALL(LBEDT(XX=DRRPC,LL=OUT,AA=AAA,BB=BBB,DIN=FILE)

where:

XX = Data Processor name
LL = File for printed output. (Default=ISERR)
AA,BB = Parameters/file names required by the Data PRocessor.
DIN = File to contain generated input data. (Default=DIN)

NOTE: See also Section 2.4, the LBEDT Command.

8.3.4 LDBUG - AUXILIARY PROCEDURE

Function: To checkout the loading of an AUTOLAY'd Computer Code. This code must exist in the ISSYS system in its relocatable binary form. A core image module is generated.

Called By: User

Calls Made: None

Description: CALL(ISSYS(XQ=LDBUG,A=XA,B=XB,R=XR,L=ISERR)

where:

XA = Overlayed output program file name.
XB = Overlayed input program file name.
XR = IF(XR>70100)RFL,XR. (Increase field length to XR)

L = ISSYS error file.

Libraries are load set to satisfy externals and memory is preset to negative infinity using program LDBUG. The AUTOLAY'd code is loaded followed by a NOGO control card from which a core image module is generated.

NOTE: XA may have same name as XB.

8.3.5 LDDYN - AUXILIARY PROC DURE

Function: Generates an AUTOLAY of programs CGMASS, MDPROC, MDSEP, STPROC TRANP and prepares libraries for loading.

Called By: DYNAM

Calls Made: None

Description: CALL(ISSYS(XQ=LDDYN,L=ISERR,R=XR)

where:

L = ISSYS error file.
R = Field length requirement. (Default=70100)

Libraries are load set to satisfy externals and memory is preset to zero using program *LDSET.

8.3.6 LDO - AUXILIARY PROCEDURE

Function: Loads overlayed Programs to create an absolute binary file for execution using Program AUTOLAY.

Called By: FLTWT, SFTSK

Calls Made: None

Description: CALL(ISSYS(XQ=LDO,L=ISERR,A=XA,B=XB,R=XR)

where:

XA = Overlayed OUTPUT program file name.
XB = Overlayed input program file name.
XR = IF(XR>70100)RFL,XR. (Increase field length to XR).
L = ISSYS error file.

NOTE: XA may have same name as XB.

Libraries are load set to satisfy externals and memory is preset to zero using program LDSET.

8.3.7 LDR - AUXILIARY PROCEDURE

Function: Creates a relocatable binary file for execution of non-overlayed programs.

Called By: SETSK

Calls Made: None

Description: CALL(ISSYS(XQ=LDR,A=XA,B=XB,L=ISERR,R=XR)

where:

XA = Output relocatable file name.
XB = Input non-overlayed program file name.
ISERR = ISSYS error file.
XR = IF (XR>70100)RFL,XR. (Increase field length to XR)

NOTE: XQ and XB may have same file name.

Libraries are load set to satisfy externals and memory is preset to zero using program LDSET.

8.3.8 LDSTZ - AUXILIARY PROCEDURE

Function: Generates an AUTOLAY of programs *ONE, *CGMASS, *TRANP, *TRIM, *MATRIX, *GUST, *MPROC, and prepares libraries for loading. In addition, a core image module is generated for MPROC.

Called By: SETPR, SETUP

Calls Made: None

Description: CALL(ISSYS(XQ=LDSTZ,L=ISERR,R=XR)

where:

ISERR = ISSYS error file.
XR = IF(XR>70100)RFL,XR. (Increase field length to XR)

Libraries are load set to satisfy externals and memory is preset to zero using program LDSET.

8.3.9 LDTEK - AUXILIARY PROCEDURE

Function: Generates an AUTOLAY of program LOOK and prepares libraries for loading. A core image module is generated.

Called By: TVPLOT

Calls Made: None

Description: CALL(ISSYS(XQ=LDTEK,A=LOOK,B=LOOK,L=ISERR)

where:

A = Output program file name.
B = Input program file name.
L = ISSYS error file.

Libraries are load set to satisfy externals and memory is preset to zero using program LDSET.

8.3.10 TITLE - AUXILIARY PROCEDURE

Function: Prints a two page ISSYS title.

Called By: GETLB

Calls Made: None

Description: CALL(ISSYS(XQ=TITLE,L=OUTPUT)

where:

L = Output file name to print TITLE.

9. PLOTTING PROCEDURES

9.1 INTRODUCTION

Plotting Procedures give the user the capability to utilize various plotting devices available to the computer operating system. These procedures accept plot vector files for paper or Tektronix terminal plot display. Various Task Procedures in the ISSYS system generate plot vector files, i.e., DYNAM, SPARI, SPARR, FLTWT . . . These files are acceptable to Plotting Procedures.

9.2 ALPHABETICAL INDEX OF PLOTTING PROCEDURES

9.3 PLOTTING PROCEDURE DESCRIPTIONS

Plotting Procedures are described in this section with procedures called by and calls made by the Procedure. Files required and created are briefly described. Plotting procedures are designed to use the current plotting devices available at LaRC.

9.3.1 CAL11 - PLOTTING PROCEDURE

Function: Plot output will be on the CALCOMP 11 inch plotter

Called By: User

Calls Made: None

Files Required: XA (Plot vector file)

Files Created: SAVPLT

Description: CALL(ISSYS(XQ=CAL11,A=XA,B=XB)

where:

XA = plot vector file.

XB = plot control card label.

9.3.2 CAL12 - PLOTTING PROCEDURE

Function: Plot output will be on the CALCOMP 12 inch plotter

Called By: User

Calls Made: None

Files Required: XA (Plot vector file)

Files Created: SAVPLT

Description: CALL(ISSYS(XQ=CAL12,X=XA,B=XB)

where:

XA = plot vector file.

XB = plot control card label.

9.3.3 CAL30 - PLOTTING PROCEDURE

Function: Plot output will be on the CALCOMP 30 inch plotter

Called By: User

Calls Made: None

Files Required: XA (Plot vector file)

Files Created: SAVPLT

Description: CALL(ISSYS(XQ=CAL30,A=XA,B=XB)

where:

XA = Plot vector file.

XB = Plot control card label.

9.3.4 CAL33 - PLOTTING PROCEDURE

Function: Plot output will be on the CALCOMP 33 inch plotter

Called By: User

Calls Made: None

Files Required: XA (Plot vector file)

Files Created: SAVPLT

Description: CALL(ISSYS(XQ=CAL33,A=XA,B=XB)

where:

XA = Plot vector file.

XB = Plot control card label.

9.3.5 TVPLOT - PLOTTING PROCEDURE

Function: To generate TEKTRON plots using the PLOT10 graphics package.

Called By: User

Files Required: Plot vector file. . . i.e., VGPLOT; MDPLOT

Files Created: Plot output to TEKTRON 4014 screen.

Description: CALL(ISSYS(XQ=TVPLOT,A=XA)

where:

XA = Plot vector file.

See Auxiliary Procedure LDTEK.

9.3.6 VARIAN - PLOTTING PROCEDURE

Function: Plot output will be on the VARIAN plotter.

Called By: User

Calls Made: None

Files Required: XA (Plot vector file)

Files Created: SAVPLT

Description: CALL(ISSYS(XQ=VARIAN,A=XA,B=XB)

where:

XA = Plot vector file.
XB = Plot control card label.

9.3.7 VRSTEC - PLOTTING PROCEDURE

Function: Plot output on the VERSATEC 35 inch plotter

Called By: User

Calls Made: None

Files Required: XA (Plot vector file)

Files Created: SAVPLT

Description: CALL(ISSYS(XQ=VRSTEC,A=XA,B=XB)

where:

XA = Plot vector file.
XB = Plot control card label.

C-3

10. DATA MODIFICATION PROCEDURES

10.1 INTRODUCTION

The purpose of the Data Modification Procedure is to access a user's database. The database may be structured in MODIFY or UPDATE Utility Library format, i.e., DECK(s) and COMDECK(s). (See APPENDIX A for an explanation of a MODIFY DATA LIBRARY.) From the user's external database the Local Data Base (LBASE) is generated.

10.2 ALPHABETICAL INDEX OF DATA MODIFICATION PROCEDURES

Section Number	Data Modification Procedures
10.3.1	MODE
10.3.2	MODF
10.3.3	MODU
10.3.4	UDATF
10.3.5	UDATQ
10.3.6	UDATW
10.3.7	UMODF

10.3 DATA MODIFICATION PROCEDURE DESCRIPTIONS

Data Modification Procedures are described in this section. Data Modification Procedures typically operate on an external database (not an integral part of the ISSYS system). The external database may be in UPDATE or MODIFY utility format. The function of each Procedure is described along with calls made to other Procedures and Procedures called. Files required and created are also described.

10.3.1 MODE - DATA MODIFICATION PROCEDURE

Function: Performs a MODIFY on selected decks in the quick mode.

Called By: MDBASE, User

Calls Made: None

Files Required: XA (OPL)

Files Created: XC (NPL), XB (COMPILE)

Description: CALL(ISSYS(XQ=MODE,A=XA)

where:

XA = MODIFY Old Program Library (OPL).

NOTE: (see NOTE for MODF)

10.3.2 MODF - DATA MODIFICATION PROCEDURE

Function: Perform a full MODIFY on specified OPL.

Called By: User

Calls Made: None

Files Required: XA (OPL)

Files Created: XC (NPL), XB (COMPILE)

Description: CALL(ISSYS(XQ=MODF,A=XA)

where:

XA = MODIFY Old Program Library (OPL)

NOTE: File XC contains MODIFY OPL's plus OPLC's followed by a zero length record all followed by the OPLD.

```
OPL
 .
 .
 .
OPLC
 .
 .
 .
(00)
OPLD
```

10.3.3 MODU - DATA MODIFICATION PROCEDURE

Function: Peforms a MODIFY in UPDATE EDIT mode. Only decks named on DECK directives are edited and written to compile file.

Called By: User

Calls Made: None

Files Required: XA (OPL)

Files Created: XC (NPL), XB (COMPILE)

Description: CALL(ISSYS(XQ=MODU,A=XA)

where:

XA = MODIFY Old Program Library (OPL)

NOTE: (See NOTE for MODF).

10.3.4 UDATF - DATA MODIFICATION PROCEDURE

Function: Performs a full UPDATE and creates a random New Program Library.

Called By: User

Calls Made: None

Files Required: XA (OPL)

Files Created: XB (COMPILE), XC (NEWPL)

Description: CALL(ISSYS(XQ=UDATF,A=XA)

where:

XA = Old Program Library (OPL)

10.3.5 UDATQ = DATA MODIFICATION PROCEDURE

Function: Performs a quick UPDATE of only the decks specified on the COMPILE card.

Called By: User

Calls Made: None

Files Required: XA (OPL)

Files Created: XB (COMPILE), XC (NEWPL)

Description: CALL(ISSYS(XQ=UDATQ,A=XA)

where:

XA = UPDATE Old Program Library (OPL)

10.3.6 UDATW - DATA MODIFICATION PROCEDURE

Function: Creates an UPDATE sequential New Program Library.

Called By: User

Calls Made: None

Files Required: XA (OPL)

Files Created: XB (compile), XC (NEWPL)

Description: CALL(ISSYS(XQ=UDATW,A=XA)

where:

XA = UPDATE Old Program Library (OPL)

10.3.7 UMODF - DATA MODIFICATION PROCEDURE

Function: Converts an UPDATE Old Program Library to a MODIFY Old Program Library.

Called By: User

Calls Made: SRTPL

Files Required: XA (UPDATE OPL)

Files Created: XA (MODIFY OPL), XC (MODIFY NPL), XB (MODIFY compile)

Description: CALL(ISSYS(XQ=UMODF,A=XA)

where:

XA = UPDATE Old Program Library (OPL).

11. UTILITY PROCEDURES

11.1 INTRODUCTION

The Utility Procedure provides additional capability to the ISSYS System user. Utilities allow the user to interface new computer codes, Task Procedures, or Data Processors to the system. Other Utilities allow the user to get copies of Task Procedures, Data Processors, or Computer Codes from the system. Capabilities for listing files are also available. Documentation for Utility procedures exists on the ISSHLP file under UN=497950C.

11.2 ALPHABETICAL INDEX OF UTILITY PROCEDURES

Section Number	ISSYS UTILITY PROCEDURES
ISSHLP FILE	ISSYS
ISSHLP FILE	CMPF
ISSHLP FILE	CMPT
ISSHLP FILE	GETDA
ISSHLP FILE	GETPL
ISSHLP FILE	GETSC
ISSHLP FILE	GETX
ISSHLP FILE	GPDA
ISSHLP FILE	GPPL
ISSHLP FILE	GPSCF
ISSHLP FILE	GPSCT
ISSHLP FILE	GPX
11.3.1	HELP
ISSHLP FILE	LDO
ISSHLP FILE	LDR
ISSHLP FILE	LIST
ISSHLP FILE	LISTDA
ISSHLP FILE	LISTH
ISSHLP FILE	LISTSC
ISSHLP FILE	LISTX
ISSHLP FILE	PUTDA
ISSHLP FILE	PUTPL
ISSHLP FILE	PUTSF
ISSHLP FILE	PUTST
ISSHLP FILE	PUTUL
ISSHLP FILE	PUTX
ISSHLP FILE	SORTDA
ISSHLP FILE	SORTH
ISSHLP FILE	SORTX

* The ISSHLP File is presented in Appendix B.

11.3 UTILITY PROCEDURE DESCRIPTIONS

Utility procedures are either described in this section or on the ISSHLP file under UN=497950C. A copy of the ISSHLP file is presented in APPENDIX B. User modifications/additions to the ISSYS System as described in the subsection titled SYSTEM LIBRARY are generated utilizing Utility Procedures. These procedures are also utilized for sorting, listing, and obtaining copies of various elements of the ISSYS library.

11.3.1 HELP - UTILITY PROCEDURE

Function: Copies a specified Task Procedure documentation from the ISSHLP file to output.

Called By: User

Calls Made: None

Description: CALL(ISSYS(XQ=HELP,A=XA)

where:

XA = Task Procedure name

NOTE: If Task Procedure name = * the complete ISSHLP file will be copied to output.

12. JOB CONTROL

12.1 JOB CONTROL DESCRIPTION

The Job Control is directed by the user and consists of a sequence of ISSYS and NOS control language commands. The NOS control commands are typically used to SAVE or REPLACE files on mass storage or for file manipulation. Most of the files of interest are described in the section on Local Files. These include the key data files which may be saved to provide restart capability and the ISSYS output files.

For Task Procedures, the available output has been classified into "required", "not required", and "optional" categories. The normal, "required" output is put on the OUTPUT file, the "not required" is put on the NOPRINT file, and the "optional" is put on the ISERR file by default. The NOPRINT file is always returned at the end of each ISSYS Command. The ISERR file is returned after the successful completion of each of the main, user-called commands. It contains additional output which may be useful in diagnosis and should be copied to OUTPUT if an abnormal termination occurs. Note that all required local file manipulation is handled by the system itself and is transparent to the user. A partial list of file names used by ISSYS is presented in the Section titled <u>LOCAL FILES</u>.

12.2 SAMPLE EXECUTION DECKS

The following are Job Control examples for typical ISSYS runs. They do not include the JOB, USER, or CHARGE cards. Also, it is necessary to (ATTACH,ISSYS) from account (227940C). A 7/8/9 card is indicated by -EOR-, and a 6/7/8/9 card is indicated by -EOF-. In the strength sizing examples, the number of aeroelastic maneuver load recalculations (updates) were arbitrarily selected.

12.2 SAMPLE EXECUTION DECKS (continued)

Example 1. Perform a strength resizing for a metal structure using the non-linear programming based optimizer. (For composite wing skins substitute TRSC for TRSM.)

```
CALL(ISSYS(XQ=MDBASE,I=INPUT,L=OUTPUT,A=ASTPL,B=227940C) - Building LBASE from
                                                            MODIFY data library
                                                            ASTPL under user
                                                            number 227940C.

CALL(ISSYS(XQ=SETUP) - Assemble programs for static analysis.
SAVE,ALOAD.          - Save load data.

CALL(ISSYS(XQ=INIT)  - Perform initial SPAR execution.
CALL(ISSYS(XQ=FINL)  - Resizing iteration with two aeroelastic load updates.
CALL(ISSYS(XQ=TRSM)
*CALL(ISSYS(XQ=RESE23,L=OUTPUT) - Resize E23 elements.

CALL(ISSYS(XQ=INIT)
CALL(ISSYS(XQ=LOOP)
CALL(ISSYS(XQ=LOOP)  - Resizing iteration with four load updates.
CALL(ISSYS(XQ=FINL)
CALL(ISSYS(XQ=TRSM)
*CALL(ISSYS(XQ=RESE23,L=OUTPUT)

CALL(ISSYS(XQ=TOTL)
CALL(ISSYS(XQ=TRSM)  - Resizing iteration with one load update.
*CALL(ISSYS(XQ=RESE23,L=OUTPUT)

SAVE,DPROPT,DPROPS,DMASST,DMASSS.
SAVE,DLOAD=DLOADF,OCS=OCSF.        - Save design data and flexible loads.

EXIT.

REWIND,ISERR. - Copy optional output to OUTPUT file for abnormal termination.

COPYSBF,ISERR,OUTPUT.

-EOR-

          (Modify instructions to create LBASE)

-EOF-
```

* Optional for the resizing of E23 (rod) elements.

12.2 SAMPLE EXECUTION DECKS (continued)

Example 2. Perform one additional iteration, creating a SPAR restart tape.

```
CALL(ISSYS(XQ=MDBASE . . .)
GET,ALOAD/UN= . . . - Get and setup current Local Files.
GET,DPROPT,DMASST,DPROPS,DMASSS.
CALL,(ISSYS(XQ=SETPR) - Setup programs and files when load data (ALOAD) is
                        available.

CALL(ISSYS(XQ=INITC,A=NA1010) - Initial SPAR execution and create restart tape.
CALL(ISSYS(XQ=LOOP)  Resizing iteration with three load updates.
CALL(ISSYS(XQ=FINL)  Creating restart tape number NA1010
CALL(ISSYS(XQ=TRSM)

EXIT.

REWIND,ISERR.
COPYSBF,ISERR,OUTPUT.
```

12.2 SAMPLE EXECUTION DECKS (continued)

Example 3. Repeat the iteration using the weight-strength resizer with the restart tape.

```
CALL(ISSYS(XQ=MDBASE . . .)
GET,ALOAD/UN= . . .
GET,DRPROPT,DMASST,
CALL(ISSYS(XQ=SETPR)
CALL(ISSYSXQ=INITR(B=NA1010) - Resizing iteration without load update using
CALL(ISSYS(XQ=TRSG)             Restart tape.

EXIT.

REWIND,ISERR.
COPYSBF,ISERR,OUTPUT.
```

12.2 SAMPLE EXECUTION DECKS (continued)

Example 4. Perform dynamic and flutter analyses with 12 natural modes and Varian plots.

```
CALL(ISSYS(XQ=MDBASE . . .)
GET,DPROPT,DPROPS,DMASST,DMASSS.

CALL(ISSYS(XQ=DYNAM,A=12) - Generate natural modes and frequencies.
SAVE,FLTIN.

CALL(ISSYS(XQ=FLTWT,A=SUBKRN,B=12) - Perform wing + tail flutter analysis.
SAVE,SPL1. - Save spline coefficients for wing.

EXIT.

REWIND,ISERR.
COPYSBF,ISERR,OUTPUT.

-EOR-
          (Modify instructions)
-EOF-
```

12.2 SAMPLE EXECUTION DECKS (continued)

Example 5. Perform a flutter analysis with required Data Blocks in card form.

```
COPYBF,INPUT,LBASE.
GET,FLTIN,SPL1.
CALL(ISSYS(XQ=FLTWT,A=SUBKRN,B=12)

EXIT.

REWIND,ISERR.
COPYSBF,ISERR,OUTPUT.

-EOR-

$$ . . .
.
.
.

-EOR-

                    (Data blocks separated by -EOR on one file)

$$ . . .
.
.
.

-EOR-

$$ . . .
.
.
.
.

-EOR-

-EOF-
```

12.2 SAMPLE EXECUTION DECKS (continued)

Example 6. Perform a dynamic analysis with a temporary change to Program *CGMASS, a supplied input data deck (normally generated by Data Processor CHGMMI), and a change in execution sequence in Task Procedure MMINRT (called from DYNAM).

```
CALL(ISSYS(XQ=GPSCF,I=INPUT,L=ISERR,A=*CGMASS) - Retrieve, edit and replace *CGMASS.
COPYBR,INPUT,CHGMMI.
CALL(ISSYS(XQ=GPX,I=INPUT,L=ISERR,A=MMINRT) - Retrieve, edit and replace MMINRT.

CALL(ISSYS(XQ=MDBASE . . .)
CALL(ISSYS(XQ=DYNAM,A=12)
.
.
.

-EOR-

(XEDIT instructions to change CGMASS, used by Utility Procedure GPSCF.

-EOR-
        *

-EOR-

(Input data for CGMASS)

-EOR-

(XEDIT instructions to change MMINRT)

-EOR-
        *

-EOR-

(Modify instructions)

-EOF-
```

* NOTE: The null record is a LIBEDIT instruction indicating simple replacement of the affected elements of ISSYS. Note that only the local file ISSYS is modified.

12.2 SAMPLE EXECUTION DECKS (continued)

Example 7. Generation of RMS stresses using gust loading with the ISAC system.

CALL(ISSYS(XQ=MDBASE,I=INPUT,L=OUTPUT,A=DSTPL,B=227940C) - Build LBASE from MODIFY data library DSTPL under User Number 227940C.

CALL(ISSYS(XQ=DYNAM,A=12,B=1) - Generate natural modes and frequencies.

CALL(ISSYS(XQ=DLIN) - Generate transformation matrix utilizing surface spline interpolation.

CALL(ISSYS(XQ=DLAT) - Generate generalized aerodynamic force matrices.

CALL(ISSYS(XQ=DYNAR) - Dynamic response flutter and gust analysis.

SAVE,RMSTR. - Save RMS stresses for strength sizing.

EXIT.

REWIND,ISERR.
COPYSBF,ISERR,OUTPUT. - Copy optional output to OUTPUT file for abnormal termination.

-EOR-
-EOF-

NOTE: File GSTIN is utilized by ISAC modules DLIN and DYNARES. This file is generated by Task Procedure DYNAM and may be saved by the user.

TAPE9 contains the ISAC database complex, and may be saved by the user for restarts between ISAC modules.

13. BIBLIOGRAPHY

Carmichael, Ralph L. and Woodward, Frank A.: An Integrated Approach to the Analysis and Design of Wings and Wing-Body Combinations in Supersonic Flow. NASA TN D-3685, 1966.

Control Data Corporation: NOS VERSION 1, MODIFY Reference Manual (60450100). March 1978.

Control Data Corporation: NOS VERSION 1 Reference Manual (60445300) NOS 1.2. Vol. 1 and 2. March 1978.

Desmarais, Robert N. and Bennett, Robert M.: An Automated Procedure for Computing Flutter Eigenvalues. J. Aircraft, Vol. II, No. 2, February 1974, pp. 75-80.

Donato, Vincent W. and Huhn, Charles R., R.: Supersonic Unsteady Aerodynamics for Wings with Trailing Edge Control Surfaces and Folded Tips. AFFDL TR 68-30, 1978.

Giesing, J. P.; Kaman, T. P.; and Rodden, W. P.: Subsonic Unsteady Aerodynamics for General Configurations. Part I: Direct Application of the Nonplanar Doublet-Lattice Method. AFFDL TR 71-5, 1971.

Giles, G. L.; Blackburn, C. L.; and Dixon, S. C.: Automated Procedures for Sizing Aerospace Vehicle Structures (SAVES). Journal of Aircraft, Vol. 9, No. 12, December 1972, pp. 812-819.

Giles, Gary L.: Computer-Aided Methods for Analysis and Synthesis of Supersonic Cruise Aircraft Structures. NASA CP-001, Proceedings of the SCAR Conference (Part 2), Held at Langley Research Center, Hampton, Virginia, November 9-12, 1976.

Gross, David W.: A Multi-Disciplinary Approach to Structural Design for Stochastic Loads. Presented at the AIAA 17th Aerospace Sciences Meeting, New Orleans, Louisiana, January 15-17, 1979. AIAA Paper No. 79-0238.

Harder, Robert T.; and Desmarias, Robert N.: Interpolation Using Surface Splines. J. Aircraft, Vol. 9, No. 2, February 1972, pp. 189-191.

LTV Aerospace Corporation, Hampton Technical Center: Computer Aided Structural Methods with Applications to a Supersonic Arrow-Wing Configuration. NASA CR-132551, 1974.

NASA Langley Research Center: NOS User's Guide for LaRC Computer Complex, NOS 1.2. March 1978.

NASA Langley Research Center: XEDIT User's Guide. Level-A, Version 2.1.7, July 1978.

Peele, L. P. and Adams, W. M., Jr.: A Digital Program for Calculating the Interaction Between Flexible Structures, Unsteady Aerodynamics, and Active Controls (ISAC). NASA TM-80040, January 1979.

Sobieszczanski, Jaroslaw: Building a Computer-Aided Design Capability Using a Standard Time Share Operating System. Presented at the Winter Annual Meeting of the ASME, Houston, Texas, November 30-December 5, 1975; the meeting proceedings "Integrated Design and Analysis of Aerospace Structures", ASME 1975, pp. 93-112.

Vanderplaats, G. N.: CONMIN-A FORTRAN Program for Constrained Function Minimization User's Manual. NASA TM X-62282, August 1973.

Watkins, Charles E.; Woolston, Donald S.; and Cunningham, Herbert J.: A Systematic Kernel Function Procedure for Determining Aerodynamic Forces on Oscillating or Steady Finite Wings at Subsonic Speeds. NASA TR R-48, 1959.

Whetstone, W. D.: "Computer Analysis of Large Linear Frames", Journal of the Structural Division ASCE, November 1969, pp. 2401-2417.

Whetstone, W. D.: SPAR Structural Analysis System Reference Manual. NASA CR-158970-1, December 1978.

Woodward, Frank A.: Analysis and Design of Wing-Body Combinations at Subsonic and Supersonic Speeds. J. Aircraft, Vol. 5, No. 6, November-December 1968, pp. 5258-534.

Wrenn, G. A.; McCullers, L. A.; and Newsom, J. R. Structural and Aeroelastic Studies of a Supersonic Arrow-Wing Configuration. NASA CR-145325, July 1978.

APPENDIX A

MODIFY DATA LIBRARY REQUIRED BY ISSYS TO BUILD LBASE

MODIFY DATA LIBRARY REQUIRED BY ISSYS TO BUILD LBASE

A.1 SCHEMA FOR GENERATING A MODIFY LIBRARY

ISSYS uses a MODIFY Data Library to build a Local Data Base (LBASE). LBASE contains data model information to be accessed by ISSYS and supplied to the various computer codes required for analysis. The following is a schema for the MODIFY Data Library required by ISSYS to build LBASE; refer to Figure A-1.

1. Each deck containing fields of data model information is designated a MODIFY COMMON DECK (OPLC).

2. The parent of each OPLC or group of OPLC's is a MODIFY DECK (OPL). This OPL contains *CALL (Call OPLC) and *IFCALL (conditionally Call OPLC) directives to appropriate OPLC's to build data blocks.

3. The appropriate *WEOR (write end of record) and *CWEOR (conditionally write end of record) must be placed after each set of data OPLC's, which may be generated by several *CALL or *IFCALL directives.

4. Data should appear in 80 column widths, therefore, a *WIDTH, 80 (defines number of columns preceding sequence information on Compile File) and *NOSEQ (no sequence information on Compile File) directives should be used at the beginning of each parent OPL.

5. A data OPLC, or set of data OPLC's, called from the parent OPL and terminated with a *WEOR or *CWEOR, define an ISSYS data block as described in the section titled, LOCAL DATA BASE, of this document.

A.2 LBASE GENERATION BY ISSYS

A MODIFY DATA LIBRARY showing the use of directives in an OPL to generate the ISSYS Local Data Base (LBASE) is presented in Figure A-1. Data Blocks are structured by MODIFY *CALL and *IFCALL directives to OPLC's. The first Data Block of a record must contain a $$key-name in columns one through seven of the first field, see OPLC-DESGN10 in Figure A-1 for example.

Before ISSYS generates an LBASE, the user must declare which OPL should be written to LBASE, for the example of Figure A-1, *EDIT AST10 will write this Deck to LBASE. The user also has the responsibility to define which conditionally called Common Decks to place on LBASE by using the DEFINE directive. Data Blocks may be serially written to one record, as shown in Figure A-3, where OPLC-STDEF; OPLC-QD102; OPLC-ELDSW are written to the fourth record. ISSYS processors must be designed to strip off $$key-words on all but the first OPLC written to a record. Thus each record begins with a $$key-word followed by the data model and terminated with an end of record (--EOR--) marker.

The $$DEFINE fields of Figure A-1 are comments describing which DEFINE name to use for the type of structure to analyze. Data Blocks will be generated via *CALL and *IFCALL directives terminated with appropriately placed *CWEOR and *WEOR directives. Several COMDECK's follow, each beginning with a $$key-name, followed by data required for a particular computer code to be executed by ISSYS. A typical Batch Job Control sequence is shown in Figure A-2 to create a LBASE. The required JOB, USER, and CHARGE control cards required by the local operating system are shown. ISSYS is then ATTACH'ed to the job. ISSYS is called to execute task procedure MDBASE. This call generates an LBASE for subsequent use by other ISSYS procedures during execution. The *DEFINE BPLATE directive builds the LBASE structure of Figure A-3. *EDIT AST10 directive will write this DECK to LBASE. Utilizing a MODIFY DATA library format to construct LBASE, allows the user to take advantage of MODIFY maintenance capabilities, and compatability with other NOS Control Statements.

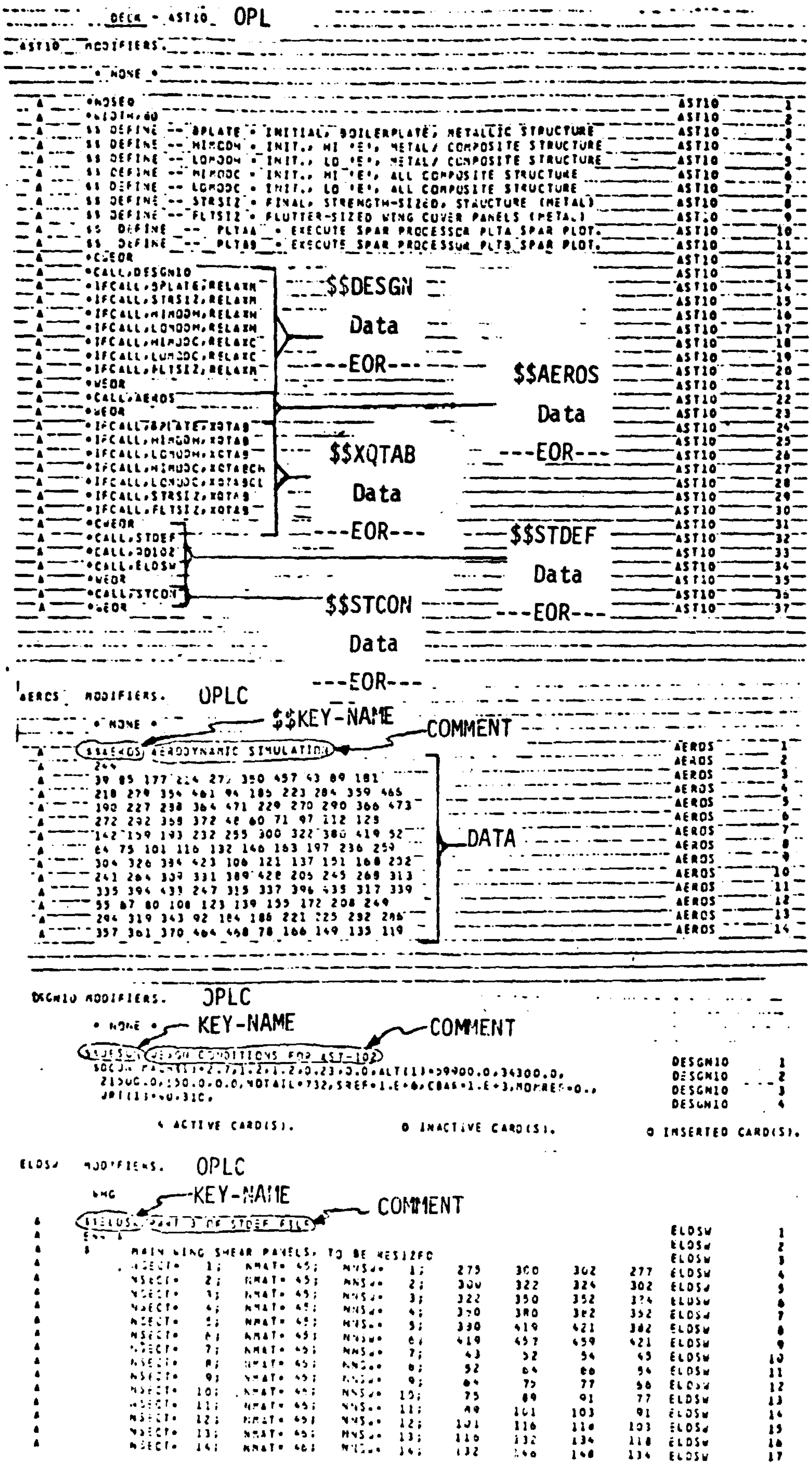

Figure A-1 Modify data library.

ORIGINAL PAGE IS OF POOR QUALITY

```
00102    MODIFIERS.       OPLC

      * NONE *   KEY-NAME                      COMMENT

▲    ([illegible]) QUADS FOR AST-102:STOEF PART 2)                                    00102     1
▲    $     BEEFED UP WHEEL WELL RIBS                                                  00102     2
▲          NSECT=25;   NMAT= 3;    283    319 $                                       00102     3
▲          NSECT=26;   NMAT= 3;    310    330 $                                       00102     4
▲    E21 $                                                                            00102     5
▲    $     TRIANGULAR MEMBRANES                                                       00102     6
▲    $     WING TRIANGULAR MEMBRANE PANELS                                            00102     7
▲          NSECT=  1;   NMAT= 19;   NNSW= 387;    32    43    55                      00102     8
▲          NSECT=  2;   NMAT= 19;   NNSW= 388;    54    45    56                      00102     9
▲          NSECT=  3;   NMAT= 19;   NNSW= 389;    78    67    80                      00102    10
▲          NSECT=  4;   NMAT= 19;   NNSW= 390;   106    94   108                      00102    11
▲          NSECT=  5;   NMAT= 19;   NNSW= 391;   107    95   109                      00102    12
▲          NSECT=  6;   NMAT= 19;   NNSW= 392;   153   139   155                      00102    13
▲          NSECT=  7;   NMAT= 19;   NNSW= 393;   154   140   156                      00102    14
▲          NSECT=  8;   NMAT= 19;   NNSW= 394;   206   190   208                      00102    15
▲          NSECT=  9;   NMAT= 19;   NNSW= 395;   207   191   209                      00102    16
▲          NSECT= 10;   NMAT= 19;   NNSW= 396;   247   229   249                      00102    17
▲          NSECT= 11;   NMAT= 19;   NNSW= 397;   248   230   250                      00102    18
▲          NSECT= 12;   NMAT= 24;   NNSW= 398;   292   272   294                      00102    19
▲          NSECT= 13;   NMAT= 24;   NNSW= 399;   293   273   295                      00102    20
▲          NSECT= 14;   NMAT= 24;   NNSW= 400;   341   319   343                      00102    21
▲          NSECT= 15;   NMAT= 24;   NNSW= 401;   342   320   344                      00102    22
▲          NSECT= 16;   NMAT= 29;   NNSW= 402;   374   401   375                      00102    23
▲          NSECT= 17;   NMAT= 29;   NNSW= 403;   375   402   371                      00102    24
▲          NSECT= 18;   NMAT= 29;   NNSW= 404;   372   374   370                      00102    25
▲          NSECT= 19;   NMAT= 29;   NNSW= 405;   373   375   371                      00102    26
▲          NSECT= 20;   NMAT= 29;   NNSW= 406;   409   437   398                      00102    27

STCON    MODIFIERS.       OPLC

      * NONE *   KEY-NAME   COMMENT

▲    ($$STCON) STATIC CONSTRAINTS)                                                    STCON     1
▲       ZERO 3; 40$                                                                   STCON     2
▲       ZERO 1,3; 310$                                                                STCON     3


STOEF    MODIFIERS.       OPLC

      * NONE *   KEY-NAME                      COMMENT

▲    ($$STOEF) STIFFNESS DEF FOR THRU AUS DEFINE, QUADRILATERALS)                     STOEF     1
▲    EXQT ELD                                                                         STOEF     2
▲    ONLINE=0 $                                                                       STOEF     3
▲    E21 $                                                                            STOEF     4
▲    $      BEAM ELEMENTS                                                             STOEF     5
▲           MAXFWL $                                                                  STOEF     6
▲           MFAT... $                                                                 STOEF     7
▲    $      ENGINE ELEMENTS                                                           STOEF     8
▲           NSECT= 1; NNSW=725; 349 352 $                                             STOEF     9
▲           NSECT= 2; NNSW=726; 362 392 $                                             STOEF    10
▲           NSECT= 3; NNSW=727; 392 431 $                                             STOEF    11
▲           NSECT= 4; NNSW=728; 431 469 $                                             STOEF    12
▲           NSECT= 5; NNSW=729; 469 519 $                                             STOEF    13
▲           NSECT= 6; NNSW=730; 519 557 $                                             STOEF    14
▲           NSECT= 7; NNSW=731; 557 591 $                                             STOEF    15
▲           NSECT= 8; NNSW=732; 591 647 $                                             STOEF    16
▲           NSECT= 9; NNSW=733; 378 399 $                                             STOEF    17
▲           NSECT=10; NNSW=734; 399 438 $                                             STOEF    18
XQTAB    MODIFIERS.       OPLC

      * NONE *   KEY-NAME
                                        COMMENT
▲    ($$XQTAB) TAB EXECUTION WITH MATERIALS)                                          XQTAB     1
▲    EXQT TAB                                                                         XQTAB     2
▲    ONLINE=0 $                                                                       XQTAB     3
▲    START 7536                                                                       XQTAB     4
▲    TITLE" SCAR MODEL BY GILES                                                       XQTAB     5
▲    TEXT$                                                                            XQTAB     6
▲    "      THIS MODEL WAS CONVERTED TO SPAR BY GARY L. GILES.                        XQTAB     7
▲    "      WORK WAS STARTED ON 12-23-74.                                             XQTAB     8
▲    "      THIS CONVERSION WILL BE SAME AS NASTRAN MODEL,                            XQTAB     9
▲    "      THAT IS, IT WILL CONTAIN BENDING MEMBERS.                                 XQTAB    10
▲    MATERIAL CONSTANTS$                                                              XQTAB    11
▲                1   16.0+6    0.3    0.100    4.9-6                                  XQTAB    12
▲                2   16.0+6    0.3    0.349    4.9-6                                  XQTAB    13
▲                3   16.0+6    0.3    0.210    4.9-6                                  XQTAB    14
▲                4   16.0+6    0.3    0.210    4.9-6                                  XQTAB    15
▲                5    3.2+6    0.3    0.042    4.9-6                                  XQTAB    16
▲                6    3.2+6    0.3    0.042    4.9-6                                  XQTAB    17
▲                7    3.2+6    0.3    0.042    4.9-6                                  XQTAB    18
▲                8    3.2+6    0.3    0.042    4.9-6                                  XQTAB    19
▲                9    3.2+6    0.3    0.042    4.9-6                                  XQTAB    20
▲               10    3.2+6    0.3    0.042    4.9-6                                  XQTAB    21
▲               11   16.0+6    0.3    0.233    4.9-6                                  XQTAB    22
```

Figure A-1 Concluded.

```
JOB - - -
USER - - -
CHARGE - - -
ATTACH,ISSYS=ISSYSnn*/UN=227940C,NA.
CALL(ISSYS(XQ=MDBASE,A=ASTPL nn*,B=227940C.
          .
          .
          .
          .
          .

(Other ISSYS Commands and NOS Control Cards)

          .
          .
          .
          .
          .

- - - EOR - - -
*DEFINE BPLATE
*EDIT AST10
```

* nn Indicates the level number.

FIGURE A-2 - Batch Job Control Sequence to Create LBASE

```
$$DESGN                        ($$Key-name)

OPLC-DESGN10                   (DATA     )      TEXT RECORDS
OPLC-RELAXM

- - - EOR - - -                (End of record)

$$AEROS

OPLC-AEROS

- - - EOR - - -

$$XQTAB

OPLC-XQTAB

- - - EOR - - -

$$STDEF

OPLC-STDEF
OPLC-QD102
OPLC-ELDSW

- - - EOR - - -

$$STCON

OPLC-STCON

- - - EOR - - -
- - - EOF - - -
```

FIGURE A-3 - LBASE Structure for *DEFINE BPLATE;
*EDIT AST10 MODIFY DIRECTIVES

APPENDIX B

ISSHLP FILE

```
CATALOG OF ISSHLP        FILE  1       79/11/23. 15.49.34.
REC    NAME       TYPE           LENGTH       CKSUM       DATE

  1    ISSYS      TEXT              211        6363
  2    NOTES      TEXT              240        7361
  3    RULES      TEXT              165        3537
  4    CMPF       TEXT              143        0645
  5    CMPT       TEXT              165        3136
  6    GETDA      TEXT              121        0345
  7    GETPL      TEXT              146        5541
  8    GETSC      TEXT              166        3203
  9    GETX       TEXT              133        1007
 10    GPDA       TEXT              164        6440
 11    GPPL       TEXT              150        3655
 12    GPSCF      TEXT              225        6212
 13    GPSCT      TEXT              223        3301
 14    GPX        TEXT              144        1777
 15    LDO        TEXT              263        0763
 16    LDR        TEXT              202        7365
 17    LIST       TEXT               66        4714
 18    LISTDA     TEXT               75        6437
 19    LISTH      TEXT               66        6106
 20    LISTSC     TEXT               65        7630
 21    LISTX      TEXT               72        1545
 22    PUTDA      TEXT              205        4214
 23    PUTPL      TEXT              162        1120
 24    PUTSF      TEXT              263        3462
 25    PUTST      TEXT              252        4577
 26    PUTUL      TEXT              214        3772
 27    PUTX       TEXT              214        1341
 28    SORTH      TEXT               70        4740
 29    SORTSC     TEXT               71        1107
 30    SORTX      TEXT               75        1447
 31    SRTPL      TEXT              124        2171
 32    SRTXT      TEXT              115        3471
 33    (00)           SUM =        6556    LIBRARY =          1

 34    ISSHLP     OPLD              103        2141   79/11/23.

 35    * EOF *        SUM =        6661
```

```
ISSYS
CALL:ISSYS(XQ=COMMAND,I=INN,L=OUT,A=AAA,B=BBB,C=CCC,D=DDD,E=EEE,R=RFL)
************************************************************************
    ISSYS       ---- THE ISSYS PROCEDURE IS THE DRIVER FOR THE REST OF
                     THE SYSTEM --- THE SPECIFIED COMMAND IS RETRIEVED
                     FROM THE FILE AND EXECUTED WITH THE PARAMETERS
                     SUPPLIED BY THE USER.
    XQ          ---- ISSYS COMMAND NAME.
    I           ---- INPUT FILE NAME (USED FOR EDIT, MODIFY, ETC.
                     INSTRUCTIONS; NOT USED FOR SOURCE DECK TYPE DATA)
    L           ---- FILE FOR PRINTED OUTPUT.  FOR TELEX JOBS, PORTIONS
                     OF THIS FILE ARE COPIED TO THE SCREEN TO INDICATE
                     THE STEPWISE RESULTS OF THE COMMAND.
    A,B,C,D,E   ---- PARAMETERS/ FILE NAMES REQUIRED BY THE COMMAND.
    R           ---- OPTIONAL RFL VALUE.  USED TO CHANGE THE FL FROM THE
                     DEFAULT VALUES SET BY THE COMMAND.
    AUX. FILES (NOT AVAILABLE FOR OTHER USES)
                ---- ISSYS,ISSLIB,LBEDT,I,L,A,B,C,D,E,R
                     XI,XL,XA,XB,XC,XD,XE,XR,COST
                     'COMMAND',XQ,XX,AA,BB,XD,DIN,ISEXQ,ISERR,NOPRINT
                     (SEE ALSO THE INDIVIDUAL COMMANDS)
```

NOTES
ISSHLP FILE NOTATION

```
********************************************************************************
     NOTE G      ---- IF THE INDICATED FILE IS NOT ALREADY 'ASSIGNED', AN
                      ATTEMPT WILL BE MADE TO 'GET' OR 'ATTACH' IT.
     NOTE LB     ---- IF THE INDICATED FILE IS NOT ALREADY 'LOCAL', IT
                      WILL BE GENERATED USING DATA FROM THE 'LBASE' FILE.
     NOTE PR     ---- FOR TASK COMMANDS, THE AVAILABLE OUTPUT HAS BEEN
                      CLASSIFIED INTO "REQUIRED", "OPTIONAL", AND "NOT
                      REQUIRED" CATAGORIES.  THE NORMAL, "REQUIRED" OUTPUT
                      IS ON THE 'OUTPUT' FILE, THE SECOND TYPE GOES ONTO
                      FILE 'ISERR', AND THE LAST TYPE IS ON 'NOPRINT'.
                      THE 'ISERR' FILE, WHICH CONTAINS INFORMATION FOR
                      DEBUGGING, IS RETURNED AT THE END OF EACH OF THE
                      MAIN, USER-CALLED, TASK COMMANDS; WHEREAS, THE
                      'NOPRINT' FILE IS ALWAYS RETURNED AT THE END OF EACH
                      'ISSYS' COMMAND.
     NOTE R      ---- IF THE INDICATED FILE IS NOT 'IN' (INPUT) TYPE, IT
                      WILL BE REWOUND BEFORE USE.
     NOTE TT     ---- FOR TELEX JOBS, IF THE 'I' FILE IS NOT ALREADY
                      'LOCAL', IT WILL BE 'ASSIGNED' TO TYPE 'TT' (THE
                      KEYBOARD).  IN ADDITION, PORTIONS OF THE 'L' FILE
                      ARE COPIED TO THE SCREEN TO INDICATE THE STEPWISE
                      PROGRESS OF THE COMMAND.
```

```
RULES
ISSYS RULES & REGULATIONS
*****************************************************************************
     RULE A      ---- ALL CHANGES MADE BY 'ISSYS' COMMANDS AFFECT 'LOCAL'
                      FILES ONLY -- PERMANENT FILES MUST BE 'REPLACE'ED BY
                      THE USER.
     RULE B      ---- FOR ALL 'PUT**' AND 'GP***' TYPE COMMANDS, IF THE
                      'I' FILE IS NOT 'LOCAL' IT WILL BE 'ASSIGNED' TO
                      'TT' (THE KEYBOARD), FOR TELEX JOBS.  FOR ALL
                      'GP***' COMMANDS, IF 'I' IS NEITHER 'TT' NOR 'IN'
                      (INPUT) TYPE, IT WILL BE REWOUND.
     RULE C      ---- IF A 'LOCAL' FILE EXISTS WITH THE SAME NAME AS AN
                      'ISSYS' COMMAND, THE 'LOCAL' FILE WILL BE USED IN
                      PLACE OF THE 'ISSYS' VERSION.
     RULE D      ---- NONE OF THE 'ISSYS' COMMANDS USE ANY SENSE
                      SWITCHES OR R-REGISTERS (EXCEPT 'EF').
     RULE E      ---- USE OF INTERNAL 'GOTO'S FOR CONTROL CARD SKIPPING OR
                      LOOPING HAS BEEN STRONGLY DISCOURAGED.
```

```
CMPF
CALL(ISSYS(XQ=CMPF,L=OUT,A=SCDECK,B=OPT,R=RFL)
*****************************************************************************
     CMPF        ---- PERFORMS AN 'FTN' COMPILATION ON THE GIVEN SOURCE
                      DECK(S).
     L      (TT) ---- FILE FOR PRINTED OUTPUT.
     A       (R) ---- FILE CONTAINING THE SOURCE DECK(S) TO BE COMPILED.
                      NOTE THAT IF 'A' IS TYPE 'IN', ONLY ONE RECORD
                      IS USED, OTHERWISE THE INTIRE FILE WILL BE
                      COMPILED.
     B           ---- OPTIONAL 'FTN' OPTIMIZATION LEVEL.  THE DEFAULT IS
                      OPT=2 WITH "UO" (UNSAFE OPTIMIZATION).
     R           ---- THE DEFAULT FL USED IS 54100B.
     ISREL (R) ---- FILE CONTAINING THE RELOCATABLE BINARY OUTPUT.
     OTHER CALLS -- NONE
     CALLED BY ---- PLTSF
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                 ---- ISX
```

```
CMPT
CALL(ISSYS(XO=CMPT,L=OUT,A=SCDECK,B=OPT,R=RFL)
**********************************************************************
     CMPT        ---- PERFORMS AN 'FTN' COMPILATION ON THE GIVEN SOURCE
                      DECK(S).  THE FOLLOWING 'NOS' SYSTEM TEXT FILES ARE
                      PROVIDED FOR COMPASS ROUTINES -- ICTEXT, NOSTEXT,
                      SYSTEXT, AND OPL.
     L      (TT) ---- FILE FOR PRINTED OUTPUT.
     A       (R) ---- FILE CONTAINING THE SOURCE DECK(S) TO BE COMPILED.
                      NOTE THAT IF 'A' IS TYPE 'IN', ONLY ONE RECORD
                      IS USED, OTHERWISE THE INTIRE FILE WILL BE
                      COMPILED.
     B           ---- OPTIONAL 'FTN' OPTIMIZATION LEVEL.  THE DEFAULT IS
                      OPT=2 WITH "UO" (UNSAFE OPTIMIZATION).
     R           ---- THE DEFAULT FL USED IS 741008.
     ISREL (R) ---- FILE CONTAINING THE RELOCATABLE BINARY OUTPUT.
     OTHER CALLS -- NONE
     CALLED BY ---- PUTST
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                 ---- ISCRO,ISCR1 ISX.
```

```
GETDA
CALL(ISSYS(XQ=GETDA,L= UT,A=OPL,B=NAME)
***************************************************************************
     GETDA      ---- THIS COMMAND RETRIEVES A GIVEN DATA BLOCK FROM A
                     'MODIFY' (OPL) FORMAT LIBRARY FILE.
     L    (TT)  ---- FIL  FOR PRINTED OUTPUT (= CATALOG OF 'ISRCE' FILE)
     A     (G)  ---- 'MOLIFY' (OPL) FORMAT LIBRARY FILE. IF THIS FILE IS
                     NOT  OCAL THIS COMMAND WILL TRY TO 'GET' IT.
     B          ---- 'MOD FY' DECK NAME FOR DATA BLOCK REQUESTED.
     ISRCE  (R) ---- FILE CONTAINING DESIRED DATA BLOCK IN CODED FORMAT.
     OTHER CALLS -- NONE
     CALLED BY ---- GPDA
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                ---- NONE
```

```
GETPL
CALL(ISSYS(XO=GETPL,L=OUT,A=NAME)
**********************************************************************
     GETPL      ---- THIS COMMAND RETRIEVES A GIVEN 'ISSYS' DATA DECK
                     FROM THE 'ISSYS' LIBRARY.
     L     (TT) ---- FILE FOR PRINTED OUTPUT (=CATALOG OF 'ISRCE' FILE)
     A          ---- 'ISSYS' DATA DECK NAME.  NOTE THAT EACH OF THESE
                     DECK NAMES IS UNIQUE.  EACH DECK MAY PRODUCE MORE
                     THAN ONE RECORD AT EXECUTION TIME.  ALSO, NOTE THAT
                     ALL OF THE 'ISSYS' DATA DECKS MAY BE OBTAINED WITH
                     ONE CALL BY USING ONLY AN ASTERISK (*).
     ISRCE (R) ---- FILE CONTAINING THE DESIRED DATA DECK(S) IN CODED
                     FORMAT.  EACH DECK WILL BE A SEPARATE RECORD.
     OTHER CALLS -- NONE
     CALLED BY ---- GPPL
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                ---- ISX
```

```
GETSC
CALL(ISSYS(XQ=GETSC,L=OUT,A=*NAME)
****************************************************************************
     GETSC      ---- THIS COMMAND RETRIEVES A GIVEN SOURCE DECK  FROM THE
                     'ISSYS' LIBRARY.
     L    (TT) ---- FILE FOR PRINTED OUTPUT (=CATALOG OF 'ISPCE' FILE)
     A          ---- 'ISSYS' SOURCE DECK NAME.  NOTE THAT EACH OF THESE
                     DECK NAMES IS UNIQUE AND IS THE SAME AS THE FIRST
                     PROGRAM OR ROUTINE IN THE DECK, PREFIXED BY AN
                     ASTERISK (*).  EACH DECK MAY CONTAIN MORE THAN ONE
                     PROGRAM OR ROUTINE.  ALSO NOTE THAT ALL OF THE
                     'ISSYS' SOURCE DECKS MAY BE OBTAINED WITH ONE CALL
                     BY USING ONLY AN ASTERISK.
     ISRCE (R) ---- FILE CONTAINING THE DESIRED SOURCE DECK(S) IN
                     CODED FORMAT.  EACH DECK WILL BE A SEPARATE
                     RECORD.
     OTHER CALLS -- NONE
     CALLED BY ---- GPSCF,GPSCT
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                ---- ISCRO
```

```
GETX
CALL(ISSYS(XQ=GETX,L=OUT,A=NAME)
========================================================================
     GETX        ---- THIS COMMAND RETRIEVES A GIVEN COMMAND OR DATA
                      PROCESSOR FROM THE 'ISSYS' LIBRARY.
     L     (TT)  ---- FILE FOR PRINTED OUTPUT (=CATALOG OF 'ISRCE' FILE)
     A           ---- 'ISSYS' COMMAND NAME.  NOTE THAT EACH OF THESE
                      COMMAND NAMES IS UNIQUE.  NOTE THAT ALL OF THE
                      'ISSYS' COMMANDS MAY BE OBTAINED WITH ONE CALL BY
                      USING ONLY AN ASTERISK.
     ISRCE (R)   ---- FILE CONTAINING THE DESIRED COMMAND(S) IN CODED
                      FORMAT.  EACH COMMAND WILL BE A SEPARATE RECORD.
     OTHER CALLS -- NONE
     CALLED BY ---- GPX
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                 ---- NONE
```

```
GPDA
CALL(ISSYS(XO=GPDA,I=INN,L=OUT,A=OPL,B=NAME)
**************************************************************************
     GPDA        ---- THIS COMMAND IS THE SUM OF THE 'GETDA' AND 'PUTDA'
                      COMMANDS, WITH AN 'XEDIT' EXECUTION IN THE MIDDLE.
                      (SEE NOTE A)
     I  (TT,R) ---- INPUT FILE WITH ONE RECORD OF 'XEDIT' INSTRUCTIONS
                      AND A SECOND OF 'LIBEDIT' DIRECTIVES.  (SEE NOTE B)
                      'LIBEDIT' EXAMPLES -- (SEE PUTDA)
     L     (TT) ---- 'MODIFY' AND 'LIBEDIT' PRINTED OUTPUT FILE.  NOTE
                      THAT 'XEDIT' OUTPUT ALWAYS GOES TO FILE 'OUTPUT'
                      (=THE SCREEN, FOR TELEX JOBS).
     A      (G) ---- 'MODIFY' (OPL) FORMAT LIBRARY FILE.  THE UPDATED
                      LIBRARY WILL BE ON THIS SAME FILE UPON COMPLETION.
     B           ---- 'MODIFY' DECK NAME FOR DATA BLOCK DESIRED.
     OTHER CALLS -- GETDA,PUTDA
     CALLED BY ---- NONE
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                 ---- ISEXO
```

```
GPPL
CALL(ISSYS(XQ=GPPL,I=INN,L=OUT,A=NAME)
**********************************************************************
     GPPL      ---- THIS COMMAND IS THE SUM OF THE 'GETPL' AND 'PUTPL'
                    COMMANDS, WITH AN 'XEDIT' EXECUTION IN THE MIDDLE.
                    (SEE NOTE A)
     I  (TT,R) ---- INPUT FILE WITH ONE RECORD OF 'XEDIT' INSTRUCTIONS
                    AND A SECOND OF 'LIBEDIT' DIRECTIVES.  (SEE NOTE B)
                    'LIBEDIT' EXAMPLES -- (SEE PUTPL)
     L    (TT) ---- 'MODIFY' AND 'LIBEDIT' PRINTED OUTPUT FILE.  NOTE
                    THAT 'XEDIT' OUTPUT ALWAYS GOES TO FILE 'OUTPUT'
                    (=THE SCREEN, FOR TELEX JOBS).
     A         ---- 'ISSYS' DATA DECK NAME FOR DATA BLOCK DESIRED.  (SEE
                    GETPL)
     OTHER CALLS -- GETPL,PUTPL
     CALLED BY ---- NONE
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
               ---- ISEXO
```

```
GPSCF
CALL(ISSYS(XQ=GPSCF,I=INN,L=OUT,A=*NAME,B=OPT,R=RFL)
**************************************************************************
     GPSCF       ---- THIS COMMAND IS THE SUM  OF THE 'GETSC' AND 'PUTSF'
                      COMMANDS WITH AN 'XEDIT' EXECUTION IN THE MIDDLE.
                      THE 'PUTSF' COMMAND CALLS 'CMPF' FOR COMPILATION
                      AND THEN 'PUTUL' TO UPDATE THE 'ISSYS' 'USER
                      LIBRARY'.  (SEE NOTE A)
     I  (TT,R) ---- INPUT FILE WITH ONE RECORD OF 'XEDIT' INSTRUCTIONS
                      AND A SECOND OF 'LIBEDIT' INSTRUCTIONS.
                      (SEE NOTE B)
                      'LIBEDIT' EXAMPLES -- (SEE PUTUL)
     L     (TT) ---- FILE FOR PRINTED OUTPUT FROM 'FTN' AND 'LIBEDIT'.
                      NOTE THAT 'XEDIT' OUTPUT ALWAYS GOES TO FILE 'OUTPUT
                      (=THE SCREEN, FOR TELEX JOBS)
     A           ---- 'ISSYS' SOURCE DECK NAME.  (SEE GETSC)
     B           ---- OPTIONAL 'FTN' OPTIMIZATION LEVEL.  THE DEFAULT IS
                      OPT=2 WITH "UO" (UNSAFE OPTIMIZATION).
     R           ---- THE DEFAULT FL USED IS 54100B, FOR COMPILATION
                      AND 50100B FOR LIBRARY MANIPULATION.
     OTHER CALLS -- GETSC,PUTSF(INCL. CMPF & PUTUL)
     CALLED BY ---- NONE
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                 ---- ISEXO
```

```
GPSCT
CALL(ISSYS(XQ=GPSCT,I=INN,L=OUT,A=*NAME,B=OPT,R=RFL)
==========================================================================
     GPSCT      ---- THIS COMMAND IS INTENDED FOR USE WITH 'COMPASS'
                     ROUTINES WHICH REQUIRE ACCESS TO SYSTEM TEXT FILES.
                     THIS COMMAND IS IDENTICAL TO 'GPSCF' EXCEPT THAT
                     'ISSYS' COMMAND 'CMPT' IS USED FOR COMPILATION.
                     (SEE NOTE A)
     I  (TT,R)  ---- INPUT FILE WITH ONE RECORD OF 'XEDIT' INSTRUCTIONS
                     AND A SECOND OF 'LIBEDIT' INSTRUCTIONS.
                     (SEE NOTE B)
                     'LIBEDIT' EXAMPLES -- (SEE PUTUL)
     L     (TT) ---- FILE FOR PRINTED OUTPUT FROM 'FTN' AND 'LIBEDIT'.
                     NOTE THAT 'XEDIT' OUTPUT ALWAYS GOES TO FILE 'OUTPUT
                     (=THE SCREEN, FOR TELEX JOBS)
     A          ---- 'ISSYS' SOURCE DECK NAME.  (SEE GETSC)
     B          ---- OPTIONAL 'FTN' OPTIMIZATION LEVEL.  THE DEFAULT IS
                     OPT=2 WITH "UO" (UNSAFE OPTIMIZATION).
     R          ---- THE DEFAULT FL USED IS 74100B, FOR COMPILATION
                     AND 50100B FOR LIBRARY MANIPULATION.
     OTHER CALLS -- GETSC,PUTST(INCL. CMPT & PUTUL)
     CALLED BY ---- NONE
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                ---- ISEXO
```

```
GPX
CALL(ISSYS(XQ=GPX,I=INN,L=OUT,A=NAME)
**********************************************************************
     GPX          ---- THIS COMMAND IS THE SUM OF THE 'GETX' AND 'PUTX'
                       COMMANDS WITH AN 'XEDIT' EXECUTION IN THE MIDDLE.
                       (SEE NOTE A)
     I  (TT,R) ---- INPUT FILE WITH ONE RECORD OF 'XEDIT' INSTRUCTIONS
                       AND A SECOND OF 'LIBEDIT' INSTRUCTIONS.
                       (SEE NOTE B)
                       'LIBEDIT' EXAMPLES -- (SEE PUTX)
     L     (TT) ---- FILE FOR PRINTED OUTPUT.  NOTE THAT 'XEDIT' OUTPUT
                       ALWAYS GOES TO THE 'OUTPUT' FILE (=THE SCREEN, FOR
                       TELEX JOBS).
     A            ---- 'ISSYS' COMMAND NAME.  (SEE GETX)
     OTHER CALLS -- GETX,PUTX
     CALLED BY ---- NONE
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                  ---- ISEXO
```

```
LDO
CALL(ISSYS(XQ=LDO,L=OUT,A=OVLAY,B=ISPROG,R=RFL)
=========================================================================
     LDO          ---- THIS COMMAND USES THE 'AUTOLAY' PROGRAM TO EXTRACT
                       THE DESIRED 'ISSYS' PROGRAM AND ITS SUBROUTINES FROM
                       THE 'ISSYS' USER LIBRARY.  THE PROGRAM IS THEN
                       'LOADED' TO GENERATE ABSOLUTE BINARY OVERLAYS.
     L     (TT)   ---- FILE FOR PRINTED OUTPUT FROM 'AUTOLAY' PLUS A
                       CATALOG OF FILE 'A' PLUS A 'FULL' LOAD MAP.
     A      (R)   ---- FILE FOR THE ABSOLUTE BINARY OUTPUT.  NOTE THAT THE
                       FILE NAME USED IN 'ISSYS' OVERLAY CALLS IS THE SAME
                       AS THE MAIN (0,0) PROGRAM NAME.
     B            ---- NAME OF THE 'ISSYS' DATA DECK, WITH PRESET 'AUTOLAY'
                       INSTRUCTIONS, TO BE USED.  THIS NAME IS THE SAME AS
                       THE NAME OF THE PROGRAM TO BE EXTRACTED FROM THE
                       USER LIBRARY.  NOTE THAT THE 'OVERLAY' CARDS MUST BE
                       INCLUDED IN THIS DATA DECK.
     R            ---- THE DEFAULT FL USED IS 70100B.
     ISREL        ---- OPTIONAL BINARY INPUT FILE.  ANY RELOCATABLE BINARY
                       ROUTINES PRESENT ON THIS FILE WILL BE USED IN PLACE
                       OF THOSE 'ISSYS' ROUTINES WITH THE SAME NAME.  ALSO,
                       THE FOLLOWING USER LIBRARIES ARE AVAILABLE --
                       FTNMLIB, LRCGOSF, AND NMACFTN.
     OTHER CALLS -- NONE
     CALLED BY ---- FLTWT,SETSK
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                  ---- ISX,ISY,ISCR0,ISCR1,ISCR2,ISCR3,ISCR4
```

```
LDR
CALL(ISSYS(XQ=LDR,L=OUT,A=RELOC,B=ISPROG,R=RFL)
********************************************************************************
     LDR         ---- THIS COMMAND USES THE 'AUTOLAY' PROGRAM TO EXTRACT
                      THE DESIRED 'ISSYS' PROGRAM AND ITS SUBROUTINES FROM
                      THE 'ISSYS' USER LIBRARY.
     L     (TT)  ---- FILE FOR PRINTED OUTPUT FROM 'AUTOLAY' PLUS A
                      CATALOG OF FILE 'A'.
     A      (R)  ---- FILE FOR RELOCATABLE BINARY OUTPUT FROM 'AUTOLAY'.
     B           ---- NAME OF THE 'ISSYS' DATA DECK, WITH PRESET 'AUTOLAY'
                      INSTRUCTIONS, TO BE USED.  THIS NAME IS THE SAME AS
                      THE NAME OF THE PROGRAM TO BE EXTRACTED FROM THE
                      USER LIBRARY.
     R           ---- THE DEFAULT FL USED IS 70100B.
     ISREL       ---- OPTIONAL BINARY INPUT FILE.  ANY RELOCATABLE BINARY
                      ROUTINES PRESENT ON THIS FILE WILL BE USED IN PLACE
                      OF THOSE 'ISSYS' ROUTINES WITH THE SAME NAME.
     OTHER CALLS -- NONE
     CALLED BY ---- SETSK
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                 ---- ISY,ISCR0,ISCR1,ISCR2,ISCR3,ISCR4
```

```
LIST
CALL(ISSYS(XQ=LIST,L=OUT,R=RFL)
*******************************************************************************
     LIST        ---- THIS COMMAND IS THE SUM OF THE 'LISTX' AND 'LISTSC'
                      COMMANDS.
     L           ---- FILE FOR PRINTED OUTPUT.
     R           ---- THE DEFAULT FL, FOR SORTING, IS 40100B.
     OTHER CALLS -- LISTX,LISTSC
     CALLED BY ---- NONE
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                 ---- NONE
```

```
LISTDA
CALL(ISSYS(XQ=LISTDA,L=OUT,A=OPL,R=RFL)
***********************************************************************
     LISTDA    ---- THIS COMMAND SORTS AND LISTS THE GIVEN 'MODIFY'
                    (OPL) FORMAT LIBRARY FILE.
     L         ---- FILE FOR PRINTED OUTPUT FROM 'MODIFY'.
     A         ---- 'MODIFY' OPL FILE NAME.
     R         ---- THE DEFAULT FL, FOR SORTING, IS 40100B.
     OTHER CALLS -- HDPL,SRTPL
     CALLED BY ---- NONE
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
               ---- ISERR
```

```
LISTH
CALL(ISSYS(XO=LISTH,L=OUT,R=RFL)
***************************************************************************
     LISTH     ---- THIS COMMAND SORTS AND LISTS THE CONTENTS OF THE
                    'ISSHLP' FILE.
     L         ---- FILE FOR PRINTED OUTPUT.
     R         ---- THE DEFAULT FL, FOR SORTING, IS 401000.
     OTHER CALLS -- HDPL,SORTH
     CALLED BY ---- NONE
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
               ---- ISX,ISY,ISZ,ISERR
```

```
LISTSC
CALL(ISSYS(XQ=LISTSC,L=OUT,R=RFL)
*********************************************************************
     LISTSC     ---- THIS COMMAND SORTS AND LISTS THE 'ISSYS' SOURCE
                     DECKS.
     L          ---- FILE FOR PRINTED OUTPUT.
     R          ---- THE DEFAULT FL, FOR SORTING, IS 40100B.
     OTHER CALLS -- HDUL,SORTSC
     CALLED BY ---- LIST
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                ---- ISX,ISCRO,ISERR
```

```
LISTX
CALL(ISSYS(XQ=LISTX,L=OUT,R=RFL)
****************************************************************************
     LISTX        ---- THIS COMMAND SORTS AND LISTS THE 'ISSYS' COMMAND,
                       DATA PROCESSOR, AND DATA BLOCK SUB-LIBRARIES.
     L            ---- FILE FOR PRINTED OUTPUT.
     R            ---- THE DEFAULT FL, FOR SORTING, IS 40100B.
     OTHER CALLS -- HDR,SORTX
     CALLED BY ---- LIST
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                  ---- ISX,ISERR
```

```
PUTDA
CALL(ISSYS(XQ=PUTDA,I=INN,L=OUT,A=OPL,B=SFILE)
**************************************************************************
     PUTDA       ---- THIS COMMAND REPLACES OR ADDS A GIVEN DATA BLOCK
                      ONTO A 'MODIFY' (OPL) FORMAT LIBRARY FILE.
                      (SEE NOTE A)
     I      (TT) ---- INPUT FILE CONTAINING 'LIBEDIT' INSTRUCTIONS.  IF
                      THIS FILE IS ALREADY LOCAL IT MUST BE POSITIONED
                      BY THE USER.  (SEE NOTE B)
                      EXAMPLES --
                         (CR)                   -- REPLACEMENT OF OLD DATA
                      OR,
                         *ADD,LIB1,OPLC/*(CR) -- ADDITION OF NEW DATA
                         (CR)
     L      (TT) ---- FILE FOR PRINTED OUTPUT FROM 'MODIFY' AND 'LIBEDIT'.
     A       (G) ---- 'MODIFY' (OPL) FORMAT LIBRARY FILE.  THIS FILE WILL
                      CONTAIN THE UPDATED LIBRARY UPON COMPLETION.
     B       (R) ---- FILE CONTAINING INPUT DATA DECKS.  NOTE THAT 'MODIFY
                      REQUIRES EACH DECK TO BE A SEPARATE RECORD.
     OTHER CALLS -- NONE
     CALLED BY ---- GPDA
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
             ---- ISI,ISN,IST,ISX,ISY,ISCRO
```

```
PUTPL
CALL(ISSYS(XO=PUTPL,I=INN,L=OUT,A=FILE)
******************************************************************************
     PUTPL      ---- THIS COMMAND REPLACES OR ADDS THE GIVEN 'ISSYS'
                     FORMAT DATA BLOCKS ONTO THE 'ISSYS' LIBRARY.
                     (SEE NOTE A)
     I     (TT) ---- INPUT FILE CONTAINING 'LIBEDIT' INSTRUCTIONS.  IF
                     THIS FILE IS ALREADY LOCAL IT MUST BE POSITIONED
                     BY THE USER.  (SEE NOTE B)
                     'LIBEDIT' EXAMPLES --
                        (CR)                    -- REPLACE OLD DATA BLOCKS
                     OR,
                        *ADD,LIB3,OPL/* (CR) -- ADD NEW DATA BLOCKS
                        (CR)
     L     (TT) ---- FILE FOR PRINTED OUTPUT.
     A      (R) ---- FILE CONTAINING 'ISSYS' DATA BLOCKS.  NOTE THAT EACH
                     DATA BLOCK MUST BE A SEPARATE RECORD.
     OTHER CALLS -- NONE
     CALLED BY ---- GPPL
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                ---- ISN,ISCRO
```

```
PUTSF
CALL(ISSYS(XQ=PUTSF,I=INN,L=OUT,A=SDECK,B=OPT,R=RFL)
*****************************************************************************
     PUTSF      ---- THIS COMMAND PERFORMS AN 'FTN' COMPILATION ON THE
                     GIVEN SOURCE DECK(S) AND REPLACES OR ADDS THE SOURCE
                     DECK(S) AND THE RELOCATABLE BINARY RECORD(S) ONTO
                     THE 'ISSYS' LIBRARY.  (SEE NOTE A)
     I     (TT) ---- INPUT FILE CONTAINING 'LIBEDIT' INSTRUCTIONS.  IF
                     THIS FILE IS ALREADY LOCAL IT MUST BE POSITIONED
                     BY THE USER.  (SEE NOTE B)
                     'LIBEDIT' EXAMPLES --
                        (CR)                   -- REPLACE OLD ROUTINES
                     OR,
                        *ADD,LIB1,REL/* (CR) -- ADD NEW BINARY
                        *ADD,LIB2,TEXT/*(CR) -- ADD NEW SOURCE DECKS
                        (CR)
     L     (TT) ---- FILE FOR PRINTED OUTPUT FROM 'FTN' AND 'LIBEDIT'.
     A      (R) ---- FILE CONTAINING 'ISSYS' FORMAT SOURCE DECKS.  WHERE
                     EACH DECK IS A 'TEXT' TYPE RECORD WITH A NAME THE
                     SAME AS THE NAME OF THE FIRST ROUTINE, PREFIXED BY
                     AN ASTERISK (*).
     B          ---- OPTIONAL 'FTN' OPTIMIZATION LEVEL.  THE DEFAULT IS
                     OPT=2 WITH "UO" (UNSAFE OPTIMIZATION).
     R          ---- THE DEFAULT FL USED IS 54100B, FOR COMPILATION
                     AND 50100B FOR LIBRARY MANIPULATION.
     OTHER CALLS -- CMPF,PUTUL
     CALLED BY ---- GPSCF
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                ---- ISREP,ISREL,ISCRO,ISX
```

```
PUTST
CALL(ISSYS(XQ=PUTST,I=INN,L=OUT,A=SDECK,B=OPT,R=RFL)
************************************************************************
     PUTST      ---- THIS COMMAND IS THE SAME AS 'PUTSF' EXCEPT THAT
                     'ISSYS' COMMAND 'CMPT' IS USED FOR COMPILATION IN
                     PLACE OF 'CMPF'.  (SEE NOTE A)
     I     (TT) ---- INPUT FILE CONTAINING 'LIBEDIT' INSTRUCTIONS.  IF
                     THIS FILE IS ALREADY LOCAL IT MUST BE POSITIONED
                     BY THE USER.  (SEE NOTE B)
                     'LIBEDIT' EXAMPLES --
                        (CR)                  -- REPLACE OLD ROUTINES
                     OR,
                        *ADD,LIB1,REL/* (CR) -- ADD NEW BINARY
                        *ADD,LIB2,TEXT/*(CR) -- ADD NEW SOURCE DECKS
                        (CR)
     L     (TT) ---- FILE FOR PRINTED OUTPUT FROM 'FTN' AND 'LIBEDIT'.
     A      (R) ---- FILE CONTAINING 'ISSYS' FORMAT SOURCE DECKS.  WHERE
                     EACH DECK IS A 'TEXT' TYPE RECORD WITH A NAME THE
                     SAME AS THE NAME OF THE FIRST ROUTINE, PREFIXED BY
                     AN ASTERISK (*).
     B          ---- OPTIONAL 'FTN' OPTIMIZATION LEVEL.  THE DEFAULT IS
                     OPT=2 WITH "UO" (UNSAFE OPTIMIZATION).
     R          ---- THE DEFAULT FL USED IS 74100B, FOR COMPILATION
                     AND 50100B FOR LIBRARY MANIPULATION.
     OTHER CALLS -- CMPT,PUTUL
     CALLED BY ---- GPSCF
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                ---- ISREP,ISREL,ISCRO,ISX
```

```
PUTUL
CALL(ISSYS(XQ=PUTUL,I=INN,L=OUT,A=ULIB,B=PRGMS,R=PFL)
*****************************************************************************
     PUTUL      ---- THIS COMMAND REPLACES OR ADDS THE GIVEN SOURCE AND/
                     OR BINARY RECORD(S) ONTO THE GIVEN 'USER LIBRARY'.
                     (SEE NOTE A)
     I    (TT) ---- INPUT FILE CONTAINING 'LIBEDIT' INSTRUCTIONS.  IF
                     THIS FILE IS ALREADY LOCAL IT MUST BE POSITIONED
                     BY THE USER.  (SEE NOTE B)
                     'LIBEDIT' EXAMPLES --
                        (CR)                    -- REPLACE OLD ROUTINES
                     OR,
                        *ADD,LIB1,REL/* (CR) -- ADD NEW BINARY
                        *ADD,LIB2,TEXT/*(CR) -- ADD NEW SOURCE DECKS
                        (CR)
     L    (TT) ---- FILE FOR PRINTED OUTPUT FROM 'LIBEDIT'.
     A     (R) ---- FILE CONTAINING 'NOS' USER LIBRARY.  THIS WILL
                     ALSO CONTAIN THE UPDATED USER LIBRARY UPON
     B     (R) ---- FILE CONTAINING INPUT SOURCE AND/OR BINARY RECORDS.
     R          ---- THE DEFAULT FL USED IS 50100B.
     OTHER CALLS -- NONE
     CALLED BY ---- PUTSF,PUTST
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                ---- ISCR0,ISCR1,ISX
```

```
PUTX
CALL(ISSYS(XQ=PUTX,I=INN,L=OUT,A=FILE)
==========================================================================
     PUTX        ---- THIS COMMAND REPLACES OR ADDS THE GIVEN COMMAND(S)
                      OR DATA PROCESSOR(S) ONTO THE 'ISSYS' LIBRARY.
                      (SEE NOTE A)
     I     (TT) ---- INPUT FILE CONTAINING 'LIBEDIT' INSTRUCTIONS.  IF
                      THIS FILE IS ALREADY LOCAL IT MUST BE POSITIONED
                      BY THE USER.  (SEE NOTE B)
                      'LIBEDIT' EXAMPLES --
                         (CR)                     -- REPLACE OLD COMMANDS
                      OR,
                         *ADD,LIB1,TEXT/*(CR) -- ADD NEW COMMANDS
                         *ADD,LIB2,TEXT/*(CR) -- ADD NEW DATA PROCESSOR
                         (CR)
     L     (TT) ---- FILE FOR PRINTED OUTPUT.  NOTE THAT 'LIBEDIT'
                      OUTPUT IS COPIED TO THE 'OUTPUT' FILE (=THE SCREEN),
                      FOR TELEX JOBS.
     A      (R) ---- FILE CONTAINING 'ISSYS' COMMANDS AND/OR DATA
                      PROCESSORS.  NOTE THAT EACH COMMAND OR PROCESSOR
                      MUST BE A SEPARATE RECORD.
     OTHER CALLS -- NONE
     CALLED BY ---- GPX
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                 ---- ISCPO,ISX
```

```
SORTH
CALL(ISSYS(XO=SORTH,L=OUT,R=RFL)
*****************************************************************************
     SORTH        ---- THIS COMMAND SORTS THE 'ISSHLP' FILE ACCORDING TO
                       RECORD NAME (DISPLAY CODE).
     L     (TT)   ---- FILE FOR PRINTED OUTPUT.
     R            ---- THE DEFAULT FL, FOR SORTING, IS 40100B.
     OTHER CALLS -- SRTXT
     CALLED BY   ---- LISTH
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                  ---- ISRCE,ISCR1,ISI,ISERR
```

```
SORTSC
CALL(ISSYS(XQ=SORTSC,L=OUT,R=RFL)
==========================================================================
     SORTSC     ---- THIS COMMAND SORTS THE 'ISSYS' SOURCE DECKS
                     ACCORDING TO RECORD NAME (DISPLAY CODE).
     L    (TT) ---- FILE FOR PRINTED OUTPUT.
     R          ---- THE DEFAULT FL, FOR SORTING, IS 40100B.
     OTHER CALLS -- SRTXT
     CALLED BY ---- LISTSC
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                ---- ISRCE,ISCRO,ISCR1,ISI,ISERR
```

```
SORTX
CALL(ISSYS(XQ=SORTX,L=OUT,R=RFL)
****************************************************************************
     SORTX       ---- THIS COMMAND SORTS THE 'ISSYS' COMMAND AND DATA
                      PROCESSOR SUB-LIBRARIES ACCORDING TO RECORD NAME
                      (DISPLAY CODE).
     L     (TT) ---- FILE FOR PRINTED OUTPUT.
     R           ---- THE DEFAULT FL, FOR SORTING, IS 40100B.
     OTHER CALLS -- SPTXT
     CALLED BY ---- LISTX
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                 ---- ISRCE,ISAVE,ISCR1,ISI
```

```
SRTPL
CALL(ISSYS(XO=SRTPL,L=OUT,A=OPL,B=NPL,R=RFL)
*************************************************************************
     SRTPL      ---- THIS COMMAND SORTS THE GIVEN 'MODIFY' (OPL) FORMAT
                     LIBRARY FILE ACCORDING TO RECORD NAME (DISPLAY CODE)
                     AND THEN RECORD TYPE (OPL OR OPLC).
     L    (TT)  ---- FILE FOR PRINTED OUTPUT.
     A          ---- 'MODIFY' OPL FILE NAME.
     B          ---- OUTPUT FILE CONTAINING THE SORTED LIBRARY.  THIS CAN
                     BE THE SAME AS FILE 'A'.
     R          ---- THE FL USED FOR SORTING IS 40100B.
     OTHER CALLS -- NONE
     CALLED BY ---- LISTDA,UMODF
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
                ---- ISCR0,ISCR1,ISI,ISX,ISY
```

```
SRTXT
CALL(ISSYS(XQ=SRTXT,L=OUT,A=OTXT,B=NTXT,R=RFL)
***********************************************************************
     SRTXT     ---- THIS COMMAND SORTS THE GIVEN 'TEXT' FORMAT FILE
                    ACCORDING TO RECORD NAME (DISPLAY CODE).
     L    (TT) ---- FILE FOR PRINTED OUTPUT.
     A         ---- 'TEXT' INPUT FILE NAME.
     B         ---- OUTPUT FILE CONTAINING THE SORTED 'TEXT' RECORDS.
                    THIS CAN BE THE SAME AS FILE 'A'.
     R         ---- THE FL USED FOR SORTING IS 40100B.
     OTHER CALLS -- NONE
     CALLED BY ---- SORTH,SORTSC,SORTX
     AUX. FILES (NOT AVAILABLE FOR OTHER USES -- SEE OTHER CALLS)
               ---- ISCRO,ISCR1,ISI,ISX,ISY
```