NASA Technical Memorandum 84486

Aircraft Noise Prediction Program User's Manual

Ronnie E. Gillian
Langley Research Center
Hampton, Virginia

NASA

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1982

ORIGINAL PAGE IS
OF POOR QUALITY

CONTENTS

PRECEDING PAGE BLANK NOT FILMED

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

1 INTRODUCTION

It is the primary function of the Aircraft Noise Prediction Program (ANOPP) to predict aircraft noise with the best currently available methods. The prediction methods implemented in ANOPP are based on referenceable methods and data. As new methods and data become available, they are evaluated and may be incorporated in ANOPP.

Although the primary function of ANOPP is accurate noise prediction of aircraft operations, several secondary functions provide the framework for the prediction capability. These secondary functions include the following:

1. Providing alternate prediction methods when available
2. Providing data for development and evaluation of prediction methods
3. Providing software that is flexible enough to adapt to future noise prediction needs
4. Providing for extension and revisions of current prediction methods
5. Providing optimum use of computer resources

ANOPP is documented in the "Aircraft Noise Prediction Program Theoretical Manual" (ref. 1), the "ANOPP Programmers' Reference Manual for the Executive System" (ref. 2), and this user's manual.

The ANOPP Theoretical Manual provides a rigorous description of the noise prediction methods implemented in ANOPP. Equations, diagrams, and technical references allow the user to select the methods applicable to the problem. Data requirements are described carefully and limiting values are given for each data item. Every ANOPP user should have a copy of the Theoretical Manual in addition to this manual.

The ANOPP Programmers' Reference Manual details the inner workings of the ANOPP executive system. Design strategies, development methods, and implementation specifics are presented in sufficient detail to facilitate program development and maintenance. The ANOPP user can refer to this manual to add or modify executive system capabilities, to transition to a new computer system, or to acquire in-depth knowledge of the executive system. This manual also includes a guide to the writer of a new functional module.

The ANNOP User's Manual provides the instructions needed to solve a wide variety of aircraft noise prediction problems. Input preparation, method selection, data storage, and programming shortcuts are described. The introduction of this manual contains brief overviews of the noise prediction problem, the capabilities and advantages of the ANOPP software, and the organization of the manual.

1.1 NOISE PREDICTION OVERVIEW

In order to predict the noise from an aircraft operation it is necessary to locate the aircraft and observer with respect to each other, model the noise-generating mechanisms, propagate the noise through the atmosphere to the observer, and calculate the quantity of the noise that is observed. The problem of predicting aircraft noise is divided within ANOPP into these four areas:

1. Aircraft flight definition

2. Source noise modeling

3. Propagation and ground effects

4. Received noise calculations

The ANOPP modules corresponding to each area are shown in figure 1.

The definition of the aircraft flight path is the first task to be performed. For this purpose the Flight Dynamics Module (FLI) was created. This module uses lift and drag tables and a model with two degrees of freedom to compute the flight path of the aircraft. It then expresses the output in a format compatible with the Geometry Module (GEO), which calculates source-to-observer geometry. The Geometry Module performs one or more coordinate-system transformations and attempts to reduce the number of times the source noise must be computed. The satisfactory completion of these modules furnishes the propagation-path information for the noise.

The selection of the noise source modules required to make a prediction can only be made after carefully consulting the ANOPP Theoretical Manual (ref. 1) and evaluating the physical hardware that is being modeled. ANOPP provides the source prediction modules along with their required data tables and attempts to validate the user input. It is possible to provide an incorrect model, and only familiarity with the prediction methods and the hardware can ensure proper usage. For these reasons the user is urged to obtain a copy of the ANOPP Theoretical Manual and use it as a companion volume to this manual.

The source prediction modules all perform similar functions: receive the input data, compute the source noise in the form of normalized unattenuated free-field mean-square pressure at a given distance from the source, and output the data in a dimensionless form to a data base. The output mean-square pressures are a function of frequency, directivity angle, and azimuth angle and are calculated for 1/3-octave bands with center frequencies specified by the user. Pure-tone data, if present, are added to broadband data in the band in which the tone falls.

Noise data that are computed by the noise source modules are computed for a reference distance from the source. To calculate the noise at the observer the Propagation Module (PRO) must be used. This module provides for spherical spreading losses, atmospheric attenuation, and ground-surface effects ~~in order to~~ predict the 1/3-octave spectra at the observer.

ORIGINAL PAGE IS
OF POOR QUALITY

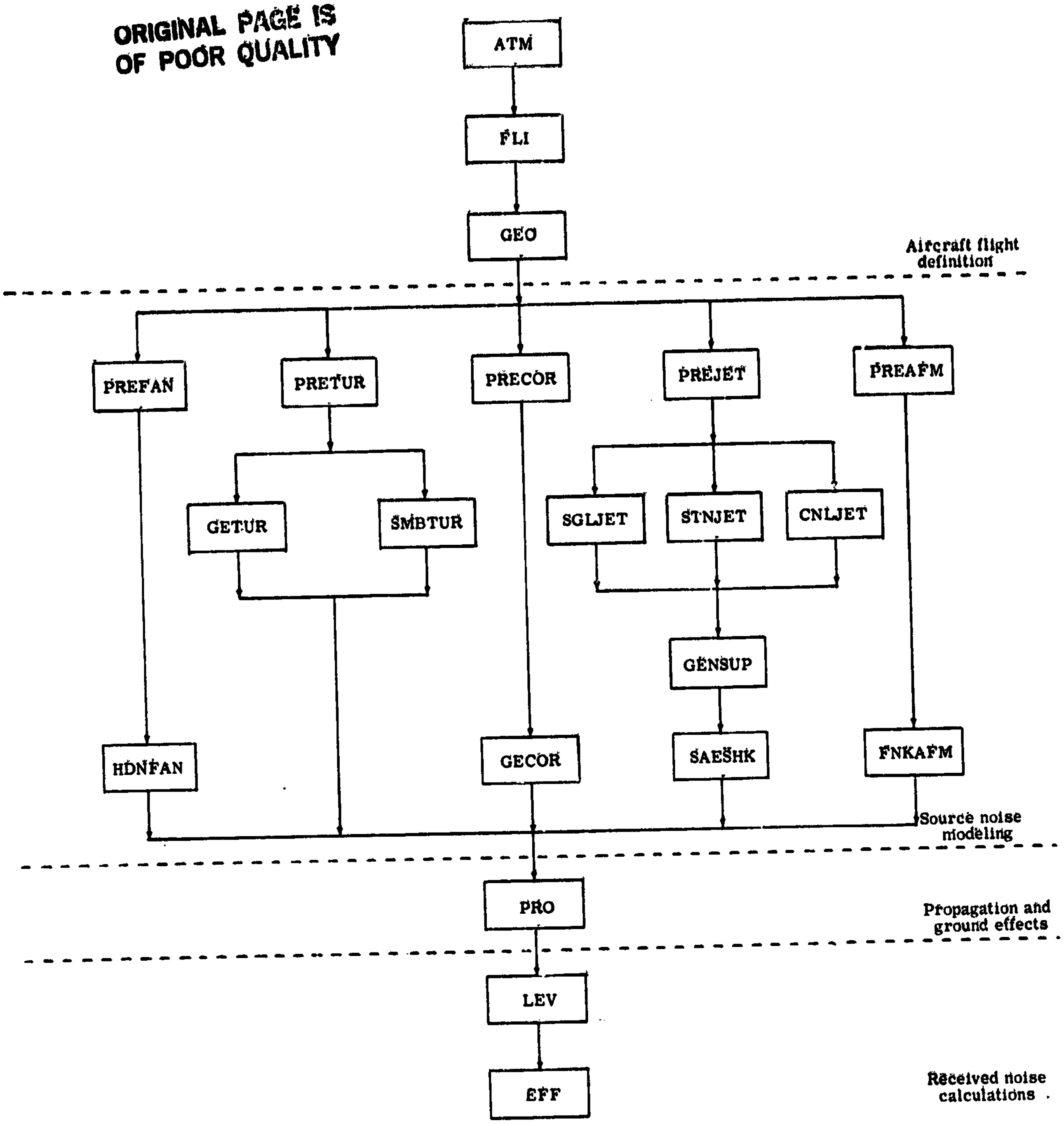


Figure 1.- Aircraft noise prediction using ANOPP.

The ANOPP user typically requires a single measure of noise received by an observer at a given time. The Noise Levels Module (LEV) performs the necessary integral over frequency to acquire this measure. This module takes the spectral information from the Propagation Module and, through application of an appropriate weighting factor, produces an A-weighted sound pressure level, a D-weighted sound pressure level, a perceived noise level (PNL), a tone-corrected perceived noise level (PNLT), or an overall sound pressure level (OASPL). In addition to the noise levels, the user can calculate the cumulative exposure to noise with the Effective Noise Module (EFF). This module uses the PNLT from LEV and produces the effective perceived noise level (EPNL) for each observer for the entire aircraft operation.

The data required for computing the solution to the general problem above are:

1. Atmospheric properties data

2. Lift- and drag-coefficient tables for the aircraft

3. Engine performance data for flight-path computation

4. Engine variable tables for each noise component to be modeled

5. Observer coordinate data

6. Specific engine operating parameters

7. Problem definition and control parameters

However, ANOPP is capable of producing predictions of aircraft noise with alternate input. Easily described flight paths can be entered manually without reference to FLI or GEO. Experimental data can be substituted for any noise source. In fact, any module can be replaced by the tables it would have generated. To obtain the exact data required for any module the user is referred to section 4 of this manual and to the description of that specific module in the Theoretical Manual (ref. 1).

1.2 SOFTWARE OVERVIEW

ANOPP emphasizes the needs of the user in all its phases. This is especially true when considering the design of the ANOPP software. The ANOPP software is organized to provide the user with a convenient method for supplying input data, for selecting between alternate prediction methods, for modifying existing methods, and for adding new capabilities.

Convenient methods for creating and modifying data are available within the ANOPP software. Large blocks of data have been separated from parametric data. Parameters may be small vectors or scalars and are maintained by name in a user area constantly available to the prediction methods. Larger blocks of data are grouped together in an optimum manner and are accessed by two names which define the logical file (called the data unit) and the subfile within that file (called the data member). The parameter capability functions similarly to the FORTRAN NAMELIST capability, while the data unit (data member) capability resembles closely the FORTRAN file input/output. In addition, most of the large data blocks needed during an ANOPP run are either created by the software or loaded during the run from auxiliary storage from a file provided by the system. This unique data management capability is also useful for storing user supplied data for problems better solved by extended studies. Although this overview gives a basic understanding of the data flow available within ANOPP, the full capabilities of data management can only be recognized by referring to the appropriate sections of the "ANOPP Programmers' Reference Manual for the Executive System" (ref. 2).

The capability for the user to easily modify the problem solution to fit his needs is another feature of the ANOPP software. Users of ANOPP can investigate the noise produced by individual engine components as well as the noise produced by an aircraft flyover. Commonly used problem solutions are available within ANOPP and minimize the user's programming task. However, the user is able to design and optimize a solution for his specific problem when necessary. This versatility is achieved through the use of a special command language described in section 3. The set of commonly used solutions (which are called procedures) is described in detail in section 5. This capability permits ANOPP to solve current noise prediction problems and also to anticipate problems of the future.

The concept of a functional module is yet another feature of ANOPP. The computer code is divided into pieces called modules according to the task the code was meant to perform. Noise computational modules are completely separate from each other and from the utility modules which serve them. Thus, new functional modules can be added and old ones modified or replaced with relative ease. Moreover, changes in the computer system on which ANOPP runs impact only those few utility modules which communicate directly with the outside system. This makes ANOPP a very reliable and maintainable tool for noise research.

1.3 USER'S MANUAL OVERVIEW

This manual is structured to accommodate the needs of users possessing various levels of ANOPP experience. For the novice, sections 1 and 2 document the basic concepts of the ANOPP system. The purpose of these sections is limited to presenting overviews of the program, data components, and terminology to the level required by the potential user. The user desiring a deeper knowledge of the internal ANOPP system should refer to the "ANOPP Programmer's Reference Manual for the Executive System" (ref. 2).

Section 3 covers ANOPP usage. It documents the capabilities of the executive system control statement language, the user's tool for formulating and executing ANOPP applications. The sections are ordered so that control statements with capabilities that are functionally related are documented together. Extensive examples of single control statements and control statement streams are provided.

Section 4 documents the installed ANOPP functional modules available on the system. These are the coded modules that perform specific acoustic-related functions and that are available for user execution during ANOPP runs. A brief functional description, input and output documentation, and examples are given for each functional module. For a detailed description of the methodology for each module, the user should refer to the ANOPP Theoretical Manual (ref. 1).

Section 5 presents the ANOPP control statement procedure library. Each procedure is a set of ANOPP executive system control statements that perform a group of related tasks, including the execution of one or more functional modules. The user is encouraged to use the procedure library whenever possible to save time and to reduce errors.

Section 6 defines the contents of the ANOPP permanent data base. Certain functional modules require acoustic-related tables and data sets. The permanent data base contains the recommended data to be used with the functional modules.

Appendixes A to E support the main sections: Appendix A contains a glossary of ANOPP terms and acronyms; appendix B contains an index to the control statements; appendix C contains techniques for error diagnostics and recovery; appendix D contains information for the operating systems for which ANOPP is available; and appendix E contains a summary of the ANOPP functional modules.

2 ANOPP CONCEPTS

This section contains definitions and vocabulary necessary to the user of the Aircraft Noise Prediction Program. It explains how the ANOPP executive system is designed, how noise prediction and data handling requirements are implemented, and how users control program flow. Once users become familiar with the basic concepts, they can refer to sections 3 and 4 for the control cards and functional modules needed to solve their problems.

2.1 SYSTEM DESIGN

The fundamental design requirements of the ANOPP system were defined as

1. Flexibility for the addition, replacement, or removal of prediction methods

2. User control for selective and effective use of the various methodologies

ANOPP meets these requirements by separating all executive functions from noise prediction functions. The program is segmented, or modular, in structure. All ANOPP executive system control tasks are performed in executive modules; all the noise prediction technology is contained in functional modules. The modular structure of ANOPP provides the desired flexibility for adding new noise prediction modules and for replacing or removing outdated ones.

ANOPP's executive system allows the user to formulate and execute numerous variations of the available noise prediction applications, without astering a programming skill or a detailed knowledge of the host operating system. The user prepares an input set of ANOPP control statements (Hollerith cards or card-image records). The user's control statement stream determines the processing flow during the ANOPP run. Using various control statements, the user selects the functional modules and establishes access to or ~~creates~~ the data items required for each functional module's execution.

The ANOPP executive system provides for all interfacing with the host computer's external operating system. It initializes the program, edits the control statement stream, processes each control statement by calling the required executive and/or functional module into execution, and performs error handling and exit procedures to the host operating system. The executive system performs data base management of the magnetic-tape or mass-storage external files on which data items identified by the user reside. It also performs data storage management within the central memory established for the ANOPP run. It manages global dynamic storage (GDS), a permanent area established within the program for every ANOPP run, where executive system and user tables are maintained throughout the run. It also manages local dynamic storage (LDS), the portion of the program's central memory left available after a module to be executed has been loaded. LDS is used for internal data storage that is required only during execution of the currently loaded module.

ORIGINAL PAGE IS
OF POOR QUALITY

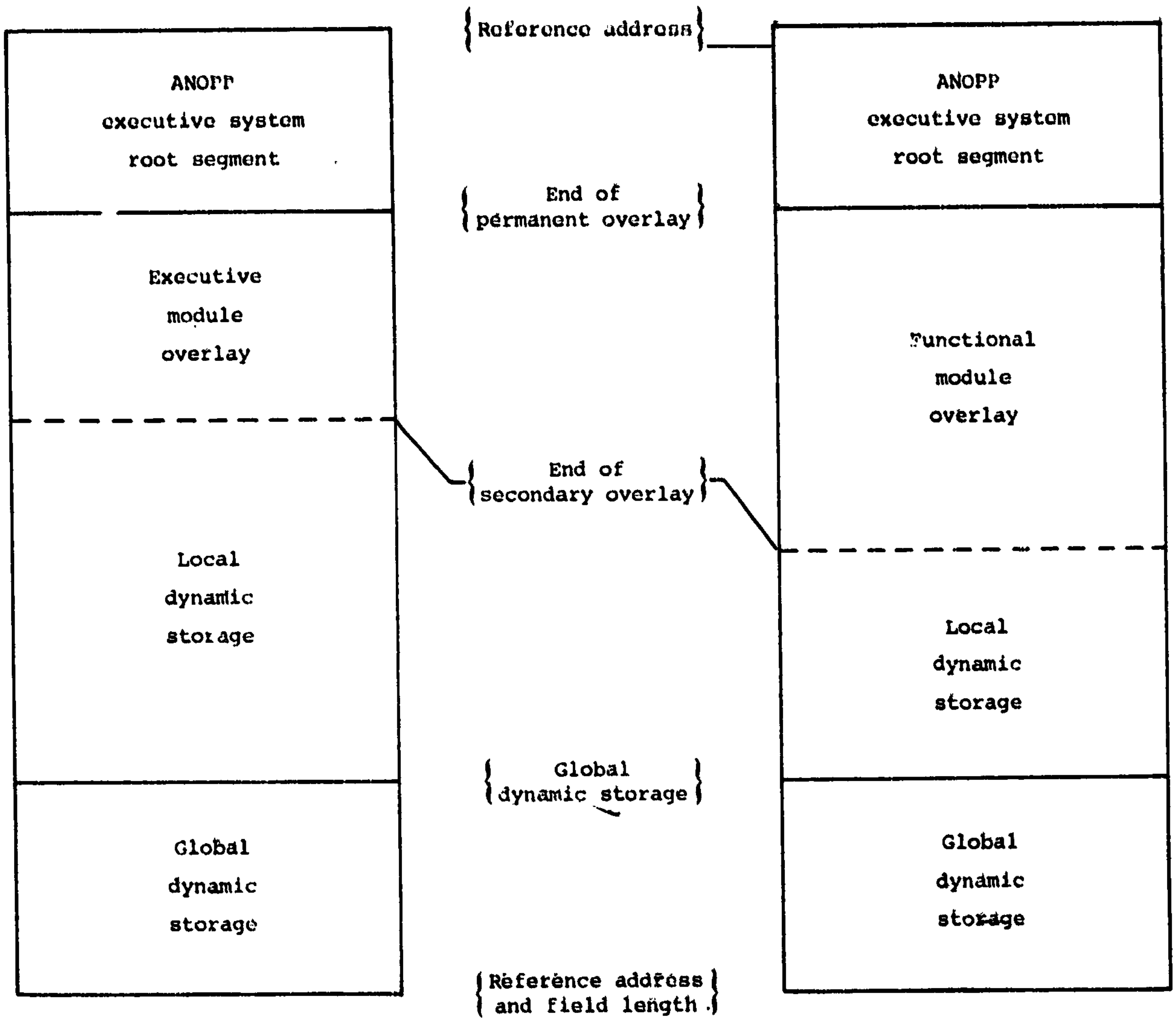
Figure 1 shows the logical structure of ANOPP during execution. Central memory is divided into four areas:

1. Executive system root segment

2. Executive or functional module overlay

3. Local dynamic storage

4. Global dynamic storage



(a) Executive module execution. (b) Functional module execution.

Figure 1.- ANOPP structure.

The executive system root segment contains all utility subprograms that are used repeatedly by the executive or functional modules. These include system-specific input and output routines, operating system interfaces, library subroutines, and others. The second memory division contains the specific module currently being executed. Local dynamic storage, as explained above, is that transient part of dynamic storage that may be overlaid when the next module is loaded. Global dynamic storage is located at the end of core memory, with the boundary between GDS and LDS set by the user at execution time.

2.2 FUNCTIONAL MODULE (FM)

A functional module (FM) is a logically independent group of subprograms that perform a specific noise-prediction-related function or a utility (data preparation or modification) function. A FM is loaded into central memory and executed when the ANOPP executive system processes a user control statement requesting its execution. After FM termination, other executive and functional modules loaded to process subsequent user control statements may occupy the same area of central memory.

Section 4 of this manual documents the functional modules currently installed on the ANOPP system. The section defines the noise prediction or utility capabilities of each FM and also describes the user parameters and data base items required as input to the FM or created within and output from it.

The FM internally satisfies its data requirements by referencing alphanumeric ANOPP names of data base items and user parameters. These names are used within section 4 to identify the items required by the FM. Before the control statement that requests execution of a particular FM, the user includes other ANOPP control statements that establish the required named data items in the ANOPP run. An item may have been created in a previously executed FM in the same run, and therefore is already in the executive system internal tables; thus, no further user control statements may be required. The item may be a user parameter or data base item that the user must create via control statements. Finally, the item may be a standard table that resides on the ANOPP permanent data base. For permanent data base items, the user must attach the item via a control statement prior to FM execution to establish it in the run.

ANOPP named data items (data base items and user parameters), and how an executing FM or a user control card specification references them, are described in the following subsections.

The user has the option of substituting alternate items with names other than those recommended in section 4 for the FM data. The alternate item may have been created by the user within the run, or may have been attached by the user from the permanent data base or user-maintained data base. When substitute data base items or user parameters are to be used, an alternate name feature of the FM executing control statement allows the user to identify which known named items are to be used to satisfy specific FM data requirements. During the FM execution, the identified alternate items are used. This alternate name concept applies as well to output data items. The user can establish output items under different names than those used within and documented as output from the functional module.

ORIGINAL PAGE IS
OF POOR QUALITY

2.3 DATA BASE

2.3.1 Data Base Components

All data items used within an ANOPP run reside within the ANOPP data base with the exception of system and user parameters. The ANOPP data base concept provides a method of storing and retrieving data on sequential and random-access devices. In addition, through the ANOPP convention of named primary data base components and the capabilities of the ANOPP data base manager, the user is relieved of the responsibility for interfacing with the host operating system when opening, closing, reading, and writing external files.

The ANOPP data base is a hierarchical structure consisting, from top to bottom, of

1. Library - A collection of data units, which resides on a sequential file

2. Data unit - An ANOPP named collection of members, which resides alone on a random-access mass-storage file except when the data unit has been unloaded onto a sequential library file for group storage or tape transfer

3. Member - An ANOPP named collection of records

4. Record - A collection of elements

5. Element - A collection of words

For the user of ANOPP who is not involved in programming functional modules or executive system routines, the data base concept of words within elements is irrelevant. Figure 2 illustrates the basic structure of an ANOPP data unit and the relationship between its resident components.

The ANOPP data base convention of assigning a name to a data unit and to each of its members allows easy identification of the specific primary data base components required by functional modules and referenced by the user throughout the ANOPP run.

The executive data base manager records the actual mass-storage address of each named member on the data unit. This technique results in efficient direct accessing of members in the internal system and is the reason that data unit resident files are always on random-access mass-storage devices.

2.3.1.1 Data Units

The ANOPP data unit (DU) is a named collection of named members. Except for those residing on sequential library files (described in section 2.3.1.3), they are equivalent to a random-access file in the external host operating system.

ORIGINAL PAGE IS
OF POOR QUALITY

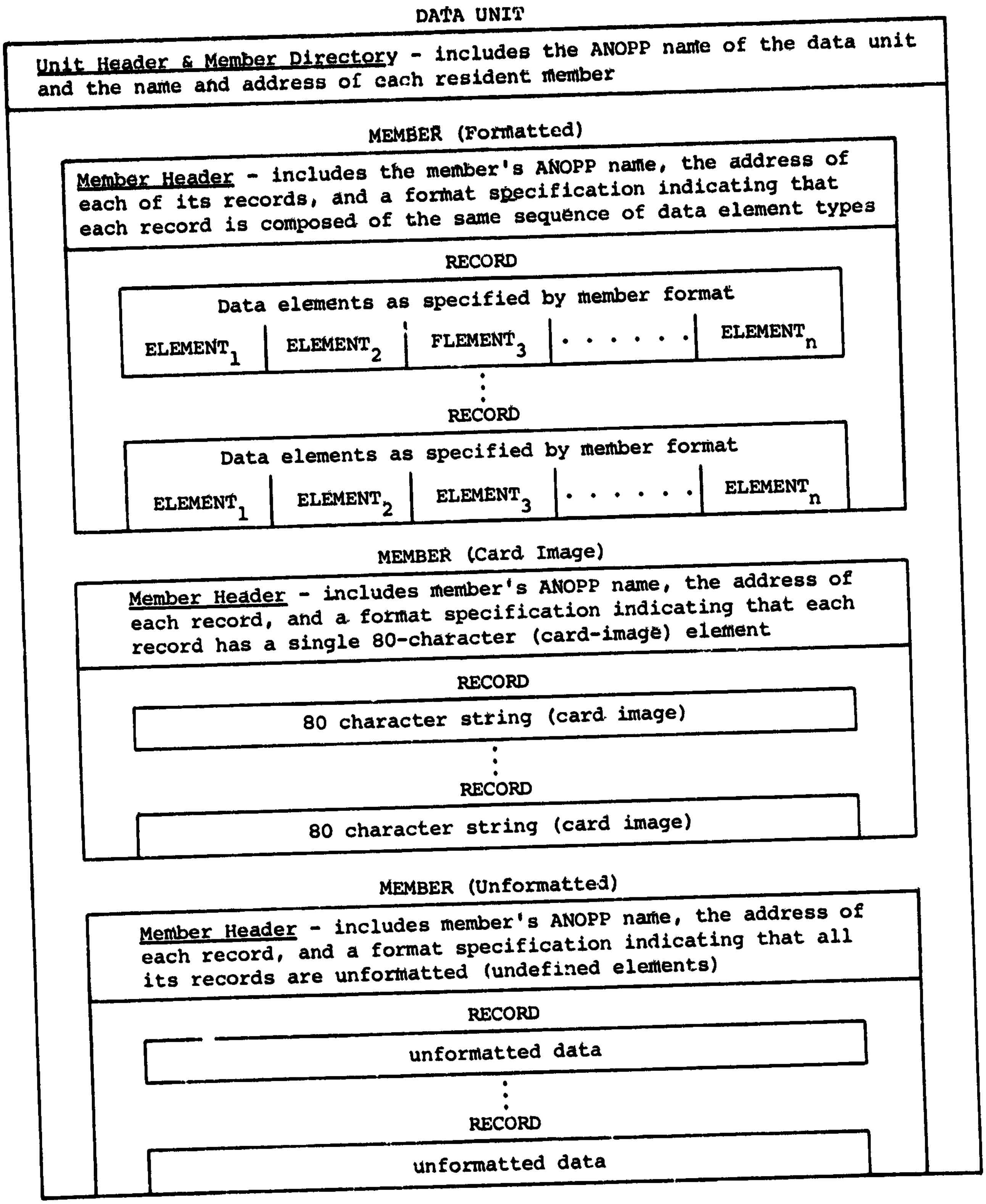


Figure 2.- ANOPP data unit structure.

The data unit is the primary component within the ANOPP data base. Section 4 documents the data base item requirements of a functional module by identifying each item by a combination of its data unit name and member name. As previously discussed in section 2.2, the internal references to a specific data item within the FM are also made with this unit (member) combination. Normally the combination documented in section 4 is used. However, the user can substitute other data unit (member) combinations via the alternate name capability of the control statement that executes the functional module.

The user establishes a data unit in the ANOPP run via control statements that attach existing data units or create new ones. When a data unit has been established in the ANOPP run, its name is recorded in the executive system's data unit directory (DUD). No two data units may have the same name at the same time in an ANOPP run.

The executive system maintains a set of its own data units within every ANOPP run. System data unit XSUNIT is used internally to satisfy executive system storage requirements. System data unit XRUNIT is used internally during checkpoint and restart runs. Both units are restricted from user reference except to list their contents. The system data unit DATA is available to the user as a working storage data unit. It may be used to create temporary run-life members that will cease to exist upon termination of the ANOPP run or to assemble members that will subsequently be copied to a new data unit that is being built within the run.

Additional user control statements allow the detachment of a previously established data unit by removing its entry from the DUD - the archiving of a unit which restricts future ANOPP access to read-only processing, the unloading of all or some of a data unit's members onto a library file, and the printing of an informative report of the content of all or some of the data units currently established in the run.

The user's responsibility for relating a data unit to its external file is limited to those units that reside either on files permanently established in the external system or on new files that the user wants to retain. In both cases, the user must know the name by which the external file is referenced on the required external system control cards. Section 2.5.5 documents this requirement.

A user may create a new data unit, from members residing on existing units and/or new members built using data from within the input control statement stream. If the new unit is temporary (not to be retained following the ANOPP run), the ANOPP system requests a temporary file on the external system.

2.3.1.2 Members

An ANOPP member is a named collection of logically related and organized records. Each member resides on a named data unit and must possess a name unique to all other members residing on the same unit. The member is the second and last of the named components in the ANOPP data base structure, the data unit being the first. Since no two data units may have the same name within an ANOPP run at the same time and because members must be uniquely named within their data unit, any combination of data unit name and member name is also unique within an ANOPP run.

The unit (member) combination is used to address a specific data base item

When specified on a user control statement

When referenced within a functional module

In Section 4, when identified as a functional module's data item requirement

In Section 5, when identified as a control statement procedure's data item requirement

In Section 6, when identified as an installed item (tables and data record sets) that is available to the user on the ANOPP permanent data base

When a user has established a data unit in a run, each named member that resides on that unit is automatically known in the run by its unit (member) combination. A user control statement or internal functional module reference to the combination results in access to the specified member.

User control statements can be used to create new members on new data units. The new members may contain combinations of records read directly from the user's input control statement stream and records read from any other member currently established in the ANOPP run.

A member consists of one or more records, each conforming in structure to the member's format specification. The ANOPP record is not a named data component. The user who builds or modifies a multirecord member must be aware of the sequential order of the records. Several control statements are available to print the record contents and format specification. The ANOPP record is not formatted in the sense of a format conversion as are FORTRAN computer language records. The format of an ANOPP record is specified for a member and must correspond to one of three general format types:

1. Unformatted records on a member are variable-length streams of data with no uniform structure (data element content) defined for the member.

2. Card-image records on a member are fixed-length (80-character) alphanumeric records, each with only one element corresponding to a Hollerith card image.

3. Formatted records on a member are fixed- or variable-length records with their structure defined by a member format specification of the sequence of data element types for each record in the member.

In the format specification of a member, the data types of the elements within records are designated by

I	integer
RS	real single precision
RD	real double precision
CS	complex single precision
L	logical
An	alphanumeric string of n characters

Internally, the executive system equates each element type specification with the number of computer words required to store a corresponding data value. Member format specification rules are documented in section 3.7.

There are two types of special purpose data base members, called procedure members and table members. Any member that does not fall within these categories is simply called a data member.

2.3.1.2.1 Procedure members

A procedure member contains a set of card-image records that are ANOPP control statements. This set of records is called a secondary input stream. A secondary input may be called into execution from within other input streams. Procedure members that have been established on the ANOPP permanent data base provide specific noise prediction and/or utility functions; they are documented in section 5 of this manual. As for all members, the user has control statement capabilities to create procedure members.

2.3.1.2.2 Table members

A table member is an unformatted member containing a single record referred to as a data table. A data table has an internal format that corresponds to a valid ANOPP data table type structure. Currently only type 1 data tables, defined below, are valid. A special set of control statements are available to create a table member and to print summary reports describing a table's structure and contents. Table members may also be copied from one data unit to another via control statements. Table members installed on the ANOPP permanent data base are documented in section 6.

A type 1 data table may be output from a functional module or created according to a set of description cards supplied by the user. Section 3.7.3 documents the description cards for the creation process. A type 1 data table may have from one to four independent variables. An independent variable in the table may be integer, real single precision, or real double precision. A dependent variable may be integer, real single precision, real double precision, or complex single precision. The independent variables need not be the same type, nor must the type of the dependent variables be the same as that of the independent variables.

While creating the table, the user specifies the acceptable interpolation procedures to be used on the table and, for each independent variable, the extrapolation method to be used if an interpolation request is outside the range of the independent variable. Interpolation and extrapolation procedures, number of independent variables, variable types, and data values are defined via description cards supplied by the user.

2.3.1.3 Sequential Library Files

Sequential library files are used to group copies of data units on a single external file. The library files are used primarily to transfer data bases between computer sites, to provide system backup, and to allow grouping of functionally related data units.

A set of control statements is provided to handle library files. Their capabilities include the building of a library, by selectively including copies of specific data units and members established in the ANOPP run, and the reverse process, resulting in the creation of new single data unit files.

2.3.2 Data Base Definition

An ANOPP data base is defined as a collection of sequential library files and data units. The following three subsections present data base concepts that are extensions of the above defintion.

2.3.2.1 Permanent Data Base

The ANOPP permanent data base is the collection of data units and library files established, maintained, and installed as part of the ANOPP system on the external files of the host computer. The external files, data units, and members that comprise this permanent data base are fully documented in sections 5 and 6 of this manual. The noise prediction functional modules whose data requirements are met by components of this data base are also identified in section 6.

2.3.2.2 User's Data Base

Typically, the user may wish to create acoustic-related table and data members that reflect new or experimental noise prediction technology and to substitute these items for those recommended to fulfill functional module data requirements in ANOPP runs. The ANOPP user's data base is that collection of library and data unit files created and maintained by the user on the host computer's external files.

The user is cautioned that external file names, identified to the ANOPP executive system as the residence of sequential libraries or data units, must begin with alpha followed by six or less characters.

Section 2.5.5 and appendix D of this manual documenting specific host operating system interface requirements assist in establishing and accessing the user's data base.

2.3.2.3 Run-Life Data Base

The ANOPP run-life data base is defined as all of the data units established in an ANOPP run and therefore currently identified by entries in the ANOPP executive system data unit directory (DUD). The terms run-life data unit and run-life member are also used in this manual. A run-life unit is one currently established in the run; a run-life member resides on a run-life unit. As a minimum, the run-life data base consists of the special purpose system data unit XSUNIT. In addition, any data unit established via user control statements in the current ANOPP run and not yet detached via subsequent control statements is part of the run-life data base. Copies of data units and members residing on sequential library files are not part of the run-life data base, although they may be used as a data source when building a run-life unit. In summation, the run-life data base consists of all currently established data units, which reside on an external random-access file that is part of the permanent data base, on the user's data base, or on a temporary file established internally by the executive system.

2.4 USER PARAMETERS

The ANOPP user may create data parameters within a run via specific control statements. The user parameter is a named data array of one or more elements of the same data type. User parameter elements may be integer, real single precision, real double precision, complex single precision, logical, or an alphanumeric string of characters.

User parameters are used as input to functional modules. Section 4 identifies the name, purpose, data type, and number of elements of each user parameter that a FM requires. The alternate name capability of the FM executing control statement allows the user to substitute an existing user parameter for the one named by the FM. Functional modules may internally create new user parameters, or change the value of existing parameters.

A user parameter is established for the duration of the ANOPP run, and its name is recorded in the executive system user parameter table. However, the user may change the value of, the data type of, or the number of elements in an existing user parameter via subsequent control statements. User parameter values may be changed to user-supplied values, to the current values of a second user parameter, or to values resulting from a specified algebraic operation performed on each element of an existing numerical user parameter. With other control statements capabilities, the user can list the values of all user parameters currently established in the run and test the value of a user parameter to determine the subsequent processing flow.

2.5 THE ANOPP RUN

An ANOPP run is a single execution of ANOPP. In contrast, the user's job submitted to the host computer may involve the execution of one or more

ANOPP runs. The structure of computer job decks depends on the operating system under which ANOPP is run, as documented in appendix D. Specific external control cards request execution of a program (ANOPP) and indicate the location of the input card or card-image deck to be read by that program (the ANOPP user's primary input stream).

By setting the value of an executive system initialization parameter, the user can specify an edit-only run. This selection limits the ANOPP run to the executive system primary edit phase. During this preprocessing phase, the control statements submitted by the user as his primary input stream are examined for correct syntax. In all ANOPP runs, this phase is performed; however, if syntax errors occur, the run always terminates after the edit phase.

2.5.1 Control Statements (CS)

Control statements (CS) are the functional directives by which the user controls an ANOPP run. They are all structured in accordance with general control statement specifications (documented in section 3.1.2), and each possesses a specific format that reflects the functional capabilities it provides to the user. An indexed list of all control statements appears in appendix B.

Specific capabilities provided by control statements include

1. Establishing the beginning and the end of the primary input stream
2. Setting the executive system run-control parameters
3. Calling secondary input streams (procedure members) and transferring control to called streams
4. Selecting print options including the printing of summary information and content of data units, members, tables, and user parameters
5. Establishing and reestablishing user parameter array values
6. Establishing conditional processing within the flow of the control statement stream, dependent on current user or system parameter values including those associated with an error condition
7. Assigning data units to the run via creation, attachment, or loading techniques
8. Terminating a data unit's current run assignment
9. Creating procedure, table, and data members on data units by using new data provided within the input stream and/or existing data residing on existing members
10. Executing functional modules

11. Creating sequential library files containing some or all of the current run-life data base

12. Establishing the checkpoint and restart capability within the run

2.5.2 Primary Input Stream

The primary input stream resides in the user's input deck to ANOPP. Only through special control statements that must immediately precede this input stream can the user set initialization run controls (including checkpoint and restart). Only within the primary input stream can the user submit new member data directly from input card images.

2.5.3 Secondary Input Stream

A procedure member is a set of control statements previously created on a card-image member. A procedure member may be called from within any other input stream. When a procedure member is called, the executive system first edits the control statements on the member and, if no syntax errors are found, passes run control to the called secondary input stream. Since called members may contain control statements that call other procedure members, the user may effect multiple levels of control within his run.

In the calling control statement, the user can also modify the content of control statements in the called member. The technique involves the text replacement of specified old CS field values with new CS field values wherever the old ones appear on control statements in the called member. The replacement takes place before the edit process. The feature allows substitution of functional modules to be executed or of data items to be used within the functional modules executed within the called stream.

2.5.4 Run-Control Parameters

Executive system run-control parameters, whose values may be altered by the user via specific control statements, are divided into two classifications: initialization parameters and system parameters.

2.5.4.1 Initialization Parameters

The initialization parameters may be assigned new values only through a special optional control statement. When used, this special control statement must immediately precede the user's primary input stream in his ANOPP input deck. The specific initialization parameters, and the user considerations involved in their value selection, are documented in section 3.1.3.

The run environment controls established via initialization parameter values include such items as

1. The selection of an edit-only run

2. The size of GDS (global dynamic storage)

3. The maximum number of entries in the executive data unit directory (DUD)

4. The maximum number of entries in the executive table directory

5. The number of lines per page to be used for all ANOPP-generated printed output

6. The maximum number of card images for new member generation in the primary input stream

2.5.4.2 System Parameters

The system parameters are those run-control parameters that may be set, and reset, via control statements within input streams. The specific system parameters, and the user considerations involved in their value selection, are documented in section 3.2.

The run environment controls established via system parameter values include such options as

1. Printing of each control statement card image during editing phases

2. Printing of each control statement card image executed during the control statement processing

3. Executive system's subsequent action when a nonfatal error condition occurs during the processing of the previous control statement

4. Accumulation of cost accounting information

The system parameters that control the control statement print options and cost accounting listed above may also be set at run initialization.

2.5.5 External System Interface

The ANOPP system was designed and structured to relieve the user of most responsibility for interface with the host computer operating system, particularly in the areas of execution time, data base, and central-memory resource management. However, like any other program, ANOPP must be submitted for execution within a job deck comprised of external system control cards in addition to the user's ANOPP input deck.

Depending on the assistance available at the host computer installation, the user must to some degree understand and satisfy the specific external system interface requirements of his ANOPP run. Appendix D identifies the specific control cards that satisfy these requirements.

ORIGINAL PAGE IS
OF POOR QUALITY

Interfacing the ANOPP run and host operating system requires knowledge of

1. ANOPP job deck structure - An appendix example is provided.

2. Job identification and permission cards - The appendix identifies these cards, but the host computer installation must provide local information.

3. ANOPP job central-memory and run-time requirements - The appendix supplies recommended settings and identifies the control cards that establish them. In addition, specific functional module and procedure member documentation in sections 4 and 5 may recommend adjustments to these settings, based on data unit, table, local dynamic storage (LDS), global dynamic storage (GDS), and time requirements.

4. ANOPP load sequence - The appendix identifies the control card or cards required to load and execute ANOPP.

5. External file control card requirements - The appendix identifies and provides usage examples of those control cards required to assign an existing external file to a job, to drop a file's assignment, and to permanently establish a file created during the ANOPP job. Computer site assistance may be required to determine any local user file-access password requirements. The user must determine which files will fulfill the FM and procedure member data base requirements of the job. These must be established in the run via control statements. Data items on the ANOPP permanent data base can be attached by referencing the names listed in section 6. If a substitute data unit (not part of the ANOPP permanent data base) is to be used, the external file name for that data unit must be known.

2.5.6 Checkpoint and Restart

Noise prediction applications, controlled by the ANOPP system, often require execution of multiple functional modules. Each functional module processed potentially establishes interim data items required by other functional modules that follow in the run. In so doing, the functional module may use considerable computer time and produce an erroneous or unwanted condition, whose cause is not immediately identified. To correct the error or to adjust input items to produce the desired final result, the user might reexecute an entire ANOPP run one or more times.

The checkpoint and restart features of the ANOPP system provide an alternative to reruns of entire ANOPP applications. The user can establish checkpoints within a run where a cycle will be created on an external checkpoint file. Each cycle on the file contains the internal system and user-established data that would be required to reestablish the operating environment that existed in the run at the time the checkpoint occurred.

Multiple checkpoints may be established throughout the user's input stream and may be activated or deactivated. The user may subsequently restart a run at any one of the checkpoints that was active during the initial run. New values of run-control parameters may be substituted, and an input stream may be inserted for processing before the restart of the original run. The restart is then initiated with the execution of the control statement that followed a specific checkpoint in the original run's input stream.

The execution of checkpoint runs and restart runs involves special control statements and parameters. Their usage is documented in section 3.10.

3 ANOPP USAGE

This section documents the specific control statements that allow the user to formulate and execute his noise prediction applications. Within major subsections, titled paragraphs discuss the variety of functionally related processing controls provided by the control statements. Each individual control statement (CS) is then documented as to purpose, specific format, functional parameter descriptions, examples, and restrictions. Where appropriate, a separate subsection presents usage examples utilizing combinations of related control statements. The contents of this section explain the use of ANOPP control statements. Any requirement for external operating system control cards associated with a specific ANOPP control statement function is indicated in the documentation of that function. The specific external control cards that satisfy these requirements may differ among operating systems and are documented in the appendixes to this manual covering external operating system to ANOPP interface. The material presented in this section is intended to be both a guide to learning ANOPP and a ready reference source for experienced ANOPP users. The potential user who does not already possess a basic understanding of what the ANOPP system is, and how it works, is advised to read section 2 of this manual. The experienced ANOPP user should find appendix B to this user's manual helpful. It is an alphabetically ordered index of the ANOPP control statements. Listed for each control statement is the subsection in section 3 that documents it and the titles and sectional locations of usage discussions related to its functional capabilities.

3.1 RUN INPUT DECK

3.1.1 Overview

The specific job-deck structure required by the operating system on which the user is executing ANOPP will contain the following:

External control cards identifying the user's job and permission to run

Pre-ANOPP-run external control cards that assign to the user's job the files and devices in the external system required by the ANOPP run, including the object code file of the ANOPP program

External control cards that load and execute the ANOPP program, occurring for each ANOPP run in the user's job

Post-ANOPP-run external control cards that, when required, permanently establish data base files created in the preceding run, detach files or devices no longer required by the user's job, or purge from the external system the permanent data base files no longer required

A run input deck of ANOPP control statements for each execution of the ANOPP program within the user's job

Depending on the host operating system, the run input deck may optionally reside on a card-image file identified on the external control card that executed the ANOPP run. The ANOPP program requires a user-supplied run input deck consisting of card-image control statements and, where required, card-image input data. The content of this deck controls the processing sequence for execution of ANOPP executive system modules and noise prediction functional modules (FM's) during the ANOPP run, and it establishes the assignment of data required by the FM's. The user's run input deck for each ANOPP execution consists of one of the following configurations:

The primary input stream

The primary input stream preceded by a single ANOPP initialization control statement

A single RSTRT CS, indicating an ANOPP restart run[a]

A single RSTRT CS, followed by the primary input stream[a]

At the initialization of every run, the ANOPP executive system performs a primary edit phase. During this phase, each card-image control statement in the user's primary input stream is edited for proper format and content. Provided edit errors are not found, each CS is rebuilt into a form acceptable to the executive processing phases that will follow.

3.1.2 Control Statement Format

Each executive control statement has a specific format presented in a corresponding subsection; all control statement formats adhere to the following conventions:

The CS directive is a free-form sequence of fields on multiple card images, using card-image columns 1 to 80. A field is a name or constant required on the directive, separated from other fields by delimiters. The fields may begin in any column as long as the sequence of fields specified for a directive is correct. A CS directive is terminated with the occurrence of an end-of-data ($) character. A CS may be continued across as many as five card images if required to complete the directive; the end-of-data ($) character appears only on the last card image.

A CS field may not be continued (split) across two cards, with these exceptions: a complex single-precision-type field, where one of its two real single-precision constants may be specified entirely on one card image and where the second single-precision constant may be specified entirely on the following card image; and a string-type field, where the character-value portion of the field may be split across card images but the leading nH type identification portion of the field specification may not. For examples of all the data-type specifications, see table 1 in section 3.1.2. Comments may be included in the columns that follow the occurrence of the end-of-data ($) character on a card image. Comments may not be continued across card images. However, a card image on which the $ is the first character to occur is treated as a comment card.

---

[a]Deck structures and user considerations required for ANOPP restart runs are documented in section 3.10.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 1.- DATA TYPES

Data type	ANOPP type code	Control statement field forms	Control statement field-size limit in characters	Value range	Control statement specification examples
Integer	I	±nnn...n (+ optional)	18 (sign excluded)	Largest, (2**31)-1; smallest, 0 (absolute values)	576460752 0
Real single precision	RS	±n. n.nn n.nnE+n n.nnE-n n.nnEn nEn nE+n nE-n (+ optional)	14	Largest, 10**+322; smallest, 10**-293 (absolute value)	7. +0.52 273.28E+18 375.15E-12 -1.33E05 1E322 322E+22 1E-293
Real double precision	RD	±n.nnDn n.nnD+n n.nnD-n nD+n nDn nD-n (+ optional)	29	Largest, 10**+322; smallest, 10**-293 (absolute value)	17.72D27 36245.1029D+25 -522.365D-02 -1D+322 2000000D15 1D-293
Complex single precision[a]	CS	(RS,RS) (RS RS) (Two real single values within parentheses)	31	See RS	(7.,1E279) (1E280 273.28E+2) (-1.7E02,0.52)
Logical	L	.TRUE. .FALSE.	7	.TRUE. .FALSE.	.TRUE. .FALSE.
Hollerith string	A	nHxxx....x (xx...x part may be split across cards)	136	1 to 132 characters (alphanumeric)	6H123456 11HABCDEFGHIJK
Name	N	xxxxxxxx	8	1 to 8 characters (first character alpha)	JETUNIT FANTABLE
Algebraic operator	AO	+ - (Must be followed by a blank)	1	+ -	+ -
Logical operator	LO	.xx.	4	.EQ. .LE. .LT. .NE. .GE. .GT.	.EQ. .GT.

[a]The complex single-precision (CS) value is specified as two RS values enclosed within a required set of parentheses and separated by one or more delimiter characters (space or comma). The CS field may be split across card images only at the delimiter position; i.e., neither RS value may be split.

The general format of a control statement (CS) is as follows:

labelb̸csnameb̸operands $ comments

label — The label is an optional one-to-eight character name tag which will be associated with the CS directive. It is used in providing conditional processing flow branching within the control statement stream via "GOTO label" capabilities of some control statements. A label is allowed on any directive except ANOPP, RSTRT, and STARTCS.

b̸ — The specification of b̸ on a CS format indicates the requirement for a field delimiter. The delimiter must be one or more commas or spaces used to separate CS fields. The comma and the space are equivalent and are processed identically by the execution system.

csname — This is the one-to-eight character control-statement name. Valid control statement names recognized by the executive system are as follows:

ANOPP	CONTINUE	EVALUATE	PARAM	TABLE
ARCHIVE	CREATE	EXECUTE	PROCEED	TABLIST
ATTACH	DATA	GOTO	PURGE	TITLE
CALL	DETACH	IF	RSTRT	UNLOAD
CATALOG	DROP	LOAD	SETSYS	UPDATE
CKPNT	ENDCS	MEMLIST	STARTCS	UPLIST

operands — These are the operand fields that are required for each of the individual control statements. Operand fields are ANOPP names, value expressions, and/or keywords.

An ANOPP name consists of from one to eight alphanumeric characters, with the first character always alpha. They are used where required by the various control statements to identify by name a user parameter, system parameter, data unit, member, external file, or functional module.

Value expressions consist of data that conform to the numerical constant, logical, and string data types shown in table 1 in section 3.1.2. Specification of algebraic or logical operations may be required.

Keywords are used on some control statements to specify the selection of an optional CS capability or the source of required data.

Additional conventions used within specific CS format specifications are as follows:

Optional fields are always shown enclosed within a set of [ ].

A field specification where the user has a choice of two or more operand types, one of which must be selected, is always shown within a set of { }.

Special delimiter characters are shown where required on control statement format specifications. These special delimiters include ( ) = / * and ;.

3.1.3 Initial Run-Environment Controls

3.1.3.1 Overview

The user has the capability of assigning values to specific executive initialization parameters and system parameters via a special ANOPP CS. When used, this optional control statement must be positioned immediately preceding the user's primary input stream in the run input deck. The initialization parameter values remain in effect for the duration of the run; the system parameter values may be subsequently and repeatedly changed via SETSYS control statements within the CS stream. The following paragraphs define each of the run-environment controls that the user may effect via the ANOPP CS capabilities.

Specifying the length of GDS (global dynamic storage).- The area of the ANOPP program's central-memory field length called GDS is used by the executive system throughout the run. Within GDS, the system establishes table areas during functional module (FM) execution and establishes storage areas used during the run. The default size of GDS is 12000 words. In section 4, documentation of specific FM's with the data units and tables used by that FM are identified. The user must determine from this documentation, and from the input CS stream, the highest number of units and tables required concurrently at any one time during the run. (Some FM documentation may include a required GDS-size value.) From section 4 documentation, and also from run experience, the user can determine if the need exists for increasing the size of GDS for specific ANOPP runs. The user can minimize the GDS requirements by dropping the run assignment of data units and library files no longer needed prior to assigning those required later in the run. An increase in GDS size may require a corresponding increase in the external control card specification of core memory (field length). Integer-type executive initialization parameter LENGL control GDS size. A sample ANOPP CS user specification would appear as LENGL=14000. Depending on the user's run requirements, GDS size may also be decreased. The minimum value assignable to LENGL is 3000.

Specifying the maximum number of executive system table-directory entries.- During functional module execution, ANOPP tables residing on table members, or created within the FM, are established within GDS as they are needed. When a table is opened for use, it is recorded in an entry of the system table directory. If section 4 documentation of an FM indicates that the maximum number of tables open simultaneously exceeds a default value of 10, the user must increase the value of integer-type initialization parameter NAETD. A sample ANOPP CS user specification would appear as NAETD=15. Depending on the user's run requirements, the maximum number of table-directory entries may be decreased, thus allowing for better utilization of GDS. The minimum value assignable to NAETD is 1.

Specifying the maximum number of executive system data-unit-directory entries.- Whenever a data unit is assigned to the user's run via the ATTACH CS, CREATE CS, or UNLOAD CS, it is known to the run via its entry in the ANOPP data unit directory (DUD). From the section 4 documentation of the FM's to be executed in his run, from the section 5 documentation of any procedure members the user may call in his run, and from the user's own control statement stream, the user may determine the maximum number of data units that must be known to his run at any one point in time. Through use of the DETACH CS and PURGE CS, the maximum may be kept as small as possible. If the number exceeds a default value of 25, the user must increase the value of integer-type initialization parameter NAEUD. A sample ANOPP CS user specification would appear as NAEUD=30. Depending on the user's run requirements, the maximum number of DUD entries may be decreased, thus allowing for better utilization of GDS. The minimum value assignable to NAEUD is 5. The user must remember that in any ANOPP run the minimum run-life data base will consist of system data units DATA and XSUNIT, that in a checkpoint or restart run system data unit XRUNIT will also exist, and that all three data units require their own DUD entries.

Specifying the number of lines per page for ANOPP printed output.- A page of printed output produced during an ANOPP run normally consists of the default value of 48 lines per page, controlled by initialization parameter NLPPM. The first five lines of each page are utilized for the standard ANOPP page header, documented within section 3.2.3 covering the TITLE control statement. The user may change the number of lines per printed page via the ANOPP CS. A sample ANOPP user specification would appear as NLPPM=40. The minimum value assignable to NLPPM is 15.

Specifying a primary-edit-phase-only run.- At initialization of each ANOPP run, the executive system performs a primary edit phase. During this phase, each control statement of the user's primary input stream is edited for correct format and content. The user submitting a new control statement stream may wish to limit the run to this editing phase by setting logical initialization parameter NOGO to .TRUE.. The ANOPP CS specification would appear as NOGO=.TRUE..

Specifying the maximum number of data cards allowed in the primary input stream.- The DATA CS, TABLE CS, and UPDATE CS allow the user the capability of defining the individual records that will reside on a member. One way to input the data is through the user's primary input stream. When the set of card images is included in the primary input stream, the user can specify the maximum number of card images allowed through the parameter MAXCARDS on the ANOPP CS. If the user does not define MAXCARDS, a default value of 10000 is used.

In addition to the initialization parameters discussed above, the user may change the default values of some system parameters via the ANOPP control statement. Usage discussions for system parameters are presented in section 3.2. Their functional description is also contained in table 2 presented in section 3.1.3.2.

3.1.3.2 ANOPP CS

Purpose: The ANOPP control statement allows the user to assign values to specified executive initialization parameters and system parameters during the ANOPP run.

Format: **ANOPP$_{\not b}$param$_1$=value$_1$[....$_{\not b}$param$_n$=value$_n$] $**

param the name of an executive system parameter or initialization parameter, subject to change via the ANOPP CS, whose value the user wishes to replace (see table 2)

value the value, of the correct data type, which is to replace the default value for the specified parameter

Examples:

```
ANOPP JECHO=.TRUE. $
ANOPP JLOG=.FALSE. LENGL=3500 $
ANOPP NLPPM=45, NOGO=.TRUE., NAEUD=6 $
```

Restrictions:

The ANOPP CS is valid only as the first CS in the user's run input deck.

The label field is illegal on the ANOPP control statement.

Assigned values must be of the correct data type and within the range prescribed for the system or initialization parameter.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 2.- ANOPP CS INITIALIZATION AND SYSTEM PARAMETERS

Parameter		Description	Value		Default value
Name	Type		Type	Range	
ACCOUNT	System[a]	Controls accumulation of cost-accounting data associated with each functional module or executive processor ACCOUNT=.TRUE. - accumulate accounting data ACCOUNT=.FALSE. - do not accumulate accounting data	Logical	.TRUE. .FALSE.	.FALSE.
CKPNT	Initialization	Identifies external name of checkpoint file to be generated during the run. Its specification also results in system parameter JCKPNT being initialized as .TRUE., automatically activating checkpoint processing in the run (See section 3.10)	Alphanumeric name of seven characters or less		CPFILE
JECHO	System[a]	Controls printing of CS card image upon validation in primary edit phase (primary input stream editing) and secondary edit phases (procedure member editing) JECHO=.TRUE. - print CS card images JECHO=.FALSE. - do not print CS card images	Logical	.TRUE. .FALSE.	.FALSE.
JLOG	System[a]	Controls printing of CS card images upon execution in executive processing phases JLOG=.TRUE. - print CS card images JLOG=.FALSE. - do not print CS card images	Logical	.TRUE. .FALSE.	.TRUE.
LENGL	Initialization	Controls size (number of computer words within user's job field length) assigned to global dynamic storage (GDS) for this ANOPP run	Integer	$\geq$3000	12000
MAXCARDS	Initialization	Controls maximum number of data cards allowed in primary input stream following an UPDATE, DATA, or TABLE CS	Integer	$\geq$1	10000
NAETD	Initialization	Controls number of table-directory entries initially allocated for this ANOPP run	Integer	$\geq$1	10
NAEUD	Initialization	Controls number of data-unit-directory entries initially allocated for this ANOPP run	Integer	$\geq$5	25
NLPPM	Initialization	Controls number of lines per page to be used for ANOPP printed output during this run	Integer	$\geq$15	48
NOGO	Initialization	Determines if ANOPP run is to be limited to primary edit phase execution only NOGO=.TRUE. - primary edit phase only run NOGO=.FALSE. - normal ANOPP run	Logical	.TRUE. .FALSE.	.FALSE.

[a]System parameter values may also be set via the SETSYS CS.

3.1.4 Primary Input Stream

3.1.4.1 Overview

The primary input stream is the set of CS card images that resides on the user's run input deck, optionally preceded only by a single ANOPP CS or RSTRT CS. The first control statement in a primary input stream must be the STARTCS control statement. The initiation of a normal ANOPP run always begins with the processing of the STARTCS CS. The last CS in a primary input stream must be the ENDCS control statement. The normal termination of a nonrestart ANOPP run always occurs with the processing of the ENDCS CS. Within the primary input stream, the user establishes the main processing control sequence for the run. Although control may be passed temporarily to a procedure member secondary input stream via a CALL CS, it must always return to the primary stream. Only within the primary input stream may the user directly input new data for the creation of members via capabilities of the DATA CS, TABLE CS, or UPDATE CS.

3.1.4.1.1 STARTCS CS

Purpose: The STARTCS control statement indicates the beginning of the user's primary input stream.

Format: **STARTCS $**

Restrictions:

STARTCS is valid only when used as the first control statement in the primary input stream. A label field is not allowed on the STARTCS.

A STARTCS CS may optionally be preceded by an ANOPP CS in the user's input deck.

3.1.4.1.2 ENDCS CS

Purpose: The ENDCS control statement indicates the end of the user's primary input stream and terminates the ANOPP run. Special usage of this control statement within ANOPP restart runs is documented in section 3.10.

Format: **[label$_{\not b}$]ENDCS $**

label label name

Examples:

```
ENDCS $
EOJ ENDCS $
```

Restriction: ENDCS is valid only when used as the last control statement in the primary input stream.

3.1.4.2 Examples

A normal ANOPP run would consist of a primary input stream consisting of

```
STARTCS $
   .
   .
other functional control statements
   .
   .
ENDCS $
```

The primary input stream may optionally be preceded by an ANOPP CS in the user's input deck:

```
ANOPP JLOG=.FALSE. $
STARTCS $
   .
   .
ENDCS $
```

For examples of primary input streams used for ANOPP restart runs, see section 3.10.

3.2 CHANGEABLE RUN-ENVIRONMENT CONTROLS

3.2.1 Overview

The user has the capability to assign values to specific executive system parameters via the SETSYS control statement. In addition, the user has the capability to establish, via the TITLE CS, a character string value that will appear as a title line in the header of each printed output page. The system parameters, unlike the initialization parameters discussed in section 3.1.3, may have their values reset repeatedly during the user's run. The following is a discussion of the usage considerations applicable to each of the system parameters.

Control the printing of each CS card image during executive editing phases.- Prior to executing the first control statement in either a primary or secondary input stream, the system performs an edit phase in which each card image in the stream is validated for correct format and syntax. At the initiation of an edit phase, the value of logical system parameter JECHO controls whether each validated CS card image is to be listed in the printed output of the ANOPP run. If JECHO is at default value .FALSE., the card images are not listed; if JECHO=.TRUE., each card image, upon edit-phase validation, will be printed. The user desiring that the primary input stream card images be listed (during the primary edit phase) must set JECHO to .TRUE. via an ANOPP CS. (See section 3.1.3.2.) During a run, the user may control the secondary edit phase printing of validated secondary input stream card images. The SETSYS CS may repeatedly be used to change the value of JECHO.

Control the printing of each CS card image during executive processing phase.- During the processing phase of an ANOPP run, each control statement card image, in either the primary or secondary input stream, may be printed when it is executed by the executive system. The user has the option of setting logical system parameter JLOG to .FALSE. which suspends the printing of control statement card images. Since JLOG is a system parameter, its current value may be reset at any time via the SETSYS CS. System parameter JLOG may also be set via the ANOPP CS (section 3.1.3.2), allowing optional control of printing the initial card images in the primary input stream.

Specify the subsequent processing flow within the CS stream following the occurrence of nonfatal error.- During the executive system's processing of an individual control statement, a "nonfatal error" may occur. The logical system parameter JCON controls the sequence of control statement processing within the CS stream following the occurrence of a nonfatal error. If JCON is at default value .FALSE., and a nonfatal error occurs, the executive processor will search forward in the CS stream, seeking the first/next PROCEED CS. If one is encountered, run control resumes with that CS; otherwise, the ENDCS will be found and the run will be terminated. However, if the value of system parameter JCON is .TRUE. (via a SETSYS CS), and a nonfatal error occurs during the processing of a CS, run control will resume with the next control statement in the CS stream.

An executing functional module may recognize certain abnormal conditions and indicate a nonfatal error occurrence to the system. Usually, a printed message will also be generated by the functional module to inform the user of what occurred. Section 4, documentation of an installed FM, will indicate what situations, if any, will result in a nonfatal error occurrence. The user's prior SETSYS CS value specification for parameter JCON will determine processing control following FM termination if a nonfatal error condition occurred. SETSYS CS specification may appear as JCON=.TRUE. or JCON=.FALSE.. System parameter JCON is not subject to ANOPP CS value specification.

Setting the title line in the ANOPP standard page header.- Each page of printed output generated during a run contains the standard ANOPP page header. The user has the capability of assigning a character string value to appear as the title line of the header via the TITLE CS. Through repeated use, he may change the content of this line during different phases of his run. The capability of this CS is documented in section 3.2.3. Also presented in section 3.2.3 is a figure which presents the ANOPP standard-page-header content.

Control checkpoint run and restart run processing.- For usage discussions on system parameters JCKPNT and JRSTRT, see section 3.10.

3.2.2 SETSYS CS

Purpose: The SETSYS control statement allows the user to assign new values to specified executive system parameters during an ANOPP run.

Format: [label$_{\not b}$]SETSYS$_{\not b}$sysparam$_1$=value$_1$[....$_{\not b}$sysparam$_n$=value$_n$] \$

label label name

sysparam the name of an executive system parameter subject to change via the SETSYS CS (see table 3)

value the value of the correct data type which is to replace the old value for the specified system parameter

Examples:

```
SETSYS JCON=.FALSE. $
LABEL SETSYS JLOG=.TRUE., JECHO=.TRUE. $
```

Restriction: Assigned values must be of the correct data type and within the range prescribed for the system parameter.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 3.- SETSYS CS SYSTEM PARAMETERS

System parameter name	Description	Value		Default value
		Type	Range	
ACCOUNT[a]	Controls accumulation of cost-accounting data associated with each functional module or executive processor ACCOUNT=.TRUE - accumulate accounting data ACCOUNT=.FALSE. - do not accumulate accounting data	Logical	.TRUE. .FALSE.	.FALSE.
JCKPNT	Controls performance of defined checkpoint operations during execution of CKPNT CS JCKPNT=.TRUE. - defined checkpoint operations will be performed during execution of a CKPNT CS JCKPNT=.FALSE. - no operation will be performed during execution of a CKPNT CS	Logical	.TRUE. .FALSE.	.FALSE. (.TRUE. if CKPNT keyword present on ANOPP or RSTRT)
JCON	Determines the following executive action when the system nonfatal error flag is set during processing of a control statement or functional module JCON=.TRUE. - execution will continue with processing of the next CS JCON=.FALSE. - execution will continue with next PROCEED CS encountered in the CS stream; if none is found, the ENDCS will normally terminate the ANOPP run	Logical	.TRUE. .FALSE.	.FALSE.
JECHO[a]	Controls printing of CS card images upon validation in primary edit phase (primary input stream editing) and secondary edit phases (procedure member editing); note that to effect printing of primary CS stream card images, JECHO must be set to .TRUE. via an ANOPP CS (see section 3.1.3.2) JECHO=.TRUE. - print CS card images JECHO=.FALSE. - do not print CS card images	Logical	.TRUE. .FALSE.	.FALSE.
JLOG[a]	Controls printing of CS card images upon execution in the executive processing phases JLOG=.TRUE. - print CS card images JLOG=.FALSE. - do not print CS card images	Logical	.TRUE. .FALSE.	.TRUE.
JRSTRT	Controls action of executive manager when ENDCS (simulated if not present) is encountered in inserted CS stream of a restart run JRSTRT=.TRUE. - this restart run will continue with execution of the defined, checkpointed CS stream; execution will begin with the CS following the last CKPNT CS processed in the checkpoint run JRSTRT=.FALSE. - this restart run will terminate after execution of the ENDCS CS in the inserted input stream	Logical	.TRUE. .FALSE.	.FALSE. (.TRUE. if RSTRT CS begins run input stream)

[a]Parameter value may also be set via the ANOPP CS.

ORIGINAL PAGE IS
OF POOR QUALITY

3.2.3 TITLE CS

Purpose: The TITLE control statement sets the value of the title line to be printed within the ANOPP standard page header. (See fig. 1.)

Format: [label$_{\not b}$]TITLE$_{\not b}$ $\left\{ \begin{matrix} \text{paramname} \\ \text{string constant} \end{matrix} \right\}$ $

label label name

paramname the valid ANOPP name of a user parameter (defined via a PARAM CS) that contains the title line value, which must be a string constant not exceeding 128 characters (16A8)

string constant the title-line Hollerith string value, not to exceed 128 characters (128Hxxxx....x)

Examples:

```
LABEL1 TITLE 9HJET NOISE $
TITLE PARAM1 $
```

Restriction: The title value must not exceed 128 characters and will be printed with the left margin justified.

Line 1	mm/dd/yy	ANOPP Lnn/nn/nn	PAGEnnnn
Line 2	Title	(128 character maximum)	
Line 3	Subtitle	(128 character maximum)	
Line 4	Label	(128 character maximum)	
Line 5		blank	

ANOPP page-header contents include:

mm/dd/yy	calendar date of ANOPP run
Lnn/nn/nn	current release level of ANOPP system
nnnn	page count initiated at zero for each run
Title	title-line value last assigned via a TITLE CS; default value = spaces
Subtitle	subtitle-line value last assigned from within a system or functional module
Label	label-line value last assigned from within a system or functional module

Figure 1.- ANOPP standard-page-header content.

3.3 USER PARAMETERS

3.3.1 Overview

The user parameter is a named array of one or more elements of the same data type. User parameters are established in an ANOPP run via the

PARAM CS, the EVALUATE CS, or via the execution of a functional module. See section 2.4 for discussion of the basic concepts concerning their use. Once established, a user parameter will be known by its assigned name for the duration of the run. A value, data type, and number of elements of a parameter may be changed repeatedly during the run. The system maintains a set of internal user parameter tables that record the last assigned value, type, and number of elements of each parameter known in the run. A user parameter may be of type A (Hollerith character string), L (logical), I (integer), RS (real single precision), RD (real double precision), or CS (complex single precision). The user may generate a listing of currently established user parameter values via the UPLIST CS.

Establishing a user parameter.- Unless a user parameter is initially established internally within a functional module, the first PARAM CS or EVALUATE CS processed that references a specific parameter results in its entry in the system user parameter tables. The type, and number of array elements, is derived from the value specification on the PARAM CS or EVALUATE CS. The value specification on a PARAM CS for a user parameter may be as follows:

One or more data constants

A reference to a second user parameter whose current value is to be assigned to the new parameter

An algebraic operation to be performed against each array element of a second numerical-type user parameter, with the individual results of the operation becoming the corresponding array elements of the new parameter

A reference to a specified element of a second user parameter whose current value will be assigned to the new parameter

The value specification on an EVALUATE CS for a user parameter may be as follows:

A combination of one or more arithmetic and/or functional operations to be performed with various parameters and values, with the result to be assigned to the new user parameter or to a specific element of an already established parameter

A reference to a specified element of a second user parameter whose current value will be assigned to the new parameter or to a specific element of an already established parameter

An example of each of these value specification forms is provided in sections 3.3.2 and 3.3.3.

Changing a user parameter.- A previously established parameter may have its value and type changed via a subsequent PARAM CS or EVALUATE CS, or during the execution of a functional module. A previously established parameter may have its array size changed via a subsequent PARAM CS or during the execution of a functional module.

ORIGINAL PAGE IS
OF POOR QUALITY

Satisfying functional module requirements for user parameters.- In section 4, documentation of installed FM's identifies the name, function, and type of each user parameter required for a given FM. Once established in a run, a parameter is available by name from within any functional module. The user may create and substitute a different user parameter name for the standard one identified in section 4 documentation of a FM data requirement. The type and array size must be as specified.

Listing user parameter values.- The UPLIST CS allows the user to generate a system user parameter table dump. The printed dump lists the name, type, system internal type code, and value of each user parameter currently known in the run. Section 3.3.4 documents the informative content of this dump.

3.3.2 PARAM CS

Purpose: The PARAM control statement establishes a new user parameter's value array of one or more elements, or it changes the value array of an already existing user parameter. A parameter's current data type and number of array elements is determined by the last value assigned it.

Format:

[label$_{\emptyset}$]PARAM$_{\emptyset}$pname$_1$=expression$_1$[....pname$_n$=expression$_n$] $

where expression may take any of the following forms:

$$\text{pname}_2\begin{pmatrix}\text{pname}_3\\ \text{integer}\end{pmatrix}$$

$$\text{pname}_2\begin{Bmatrix}+\\ -\\ *\\ /\end{Bmatrix}\text{numerical constant}$$

value

label label name

pname$_1$ the name of the user parameter for which a value is to be established or changed

pname$_2$ the name of a previously established user parameter

pname$_3$ the name of a previously established single-element user parameter of integer type

value an array of one or more elements of type numerical constant (I, RS, RD, or CS), logical constant (L), or string constant (A). Array elements must be separated by one or more delimiters (comma or space). Examples of each value-type specification are presented below. For all variations of a data-type control statement specification, see table 1 in section 3.1.2. When the value specification is for a multielement Hollerith string-type parameter, each string constant must specify the same character field length (nHxx...x).

numerical constant a numerical constant of type I, RS, RD, or CS

In the first form, the current value of an unsubscripted parameter $pname_2$ will be assigned to user parameter $pname_1$. If $pname_2$ has a subscript, then the value of the specified element from the $pname_2$ array will be assigned to the user parameter $pname_1$. If $pname_3$ is the subscript, it must be of integer type and contain only one element. Otherwise, the subscript must be an integer constant. $Pname_2$ must contain at least the number of elements specified in the subscript.

In the second form, the specified algebraic operation will be performed against each element of this parameter's value array, with the result of each operation established in the corresponding element of parameter $pname_1$'s value array. The type of $pname_2$ (I, RS, RD, or CS) must be the same as that of the specified numerical constant.

When performing an algebraic operation on complex single-precision-type (CS) numbers, each CS number is treated as a two-element single-precision array (RS, RS). Then, addition, subtraction, multiplication, and division of complex numbers will be calculated as follows, with A, B, C, and D each being real single-precision numbers:

```
(A,B)+(C,D) = (A+C,B+D)
(A,B)-(C,D) = (A-C,B-D)
(A,B)*(C,D) = ((AC+BD),(AD+BC))
(A,B)/(C,D) = (((AC+BD)/(C**2+D**2)),((BC-AD)/(C**2+D**2)))
```

Examples:

The following establishes real double-precision-type parameter L (1 element) and integer-type parameters G (1 element) and C (1 element):

```
PARAM L=1.5D-2, G=1234, C = 2 $
```

The following establishes string-constant-type user parameter KEYS (3 elements):

```
LABEL1 PARAM KEYS=4HFAN 4HJET 4HPROP $
```

The following establishes logical parameter RESULTS (2 elements):

```
PARAM RESULTS=.TRUE. .FALSE. $
```

The following establishes integer-type parameter NUMBS (2 elements) and real double-precision-type parameter STATS (2 elements):

```
PARAM NUMBS=3476 5962, STATS=17.67D27 -1.D+194 $
```

The following establishes complex single-precision-type parameter AGROUP (2 elements):

```
PARAM AGROUP=(7.,177E+17) (265.17E+2 -1.75) $
```

The following establishes real double-precision-type parameter T by using an algebraic operation based on the current value of parameter L; establishes integer-type parameter A with the value of current parameter G; and establishes real double-precision parameter B based on the current value of parameter T (note that the add operator + is followed by spaces):

```
PARAM T=L*17.62D-07, A=G, B=T + 2.0D00 $
```

The following assigns the third element of parameter KEYS to parameter E and the Cth element of parameter KEYS to parameter D:

```
PARAM E=KEYS(3) , D=KEYS(C) $
```

Restrictions:

The numerical constant used in a PARAM CS specified algebraic operation must be of the same numerical type as the specified user parameter ($pname_2$) used in that operation.

The operators + or - must be followed by a blank.

Each value specified for a multielement user-parameter array must be of the same type.

Each value specified for a multielement Hollerith string parameter must be of the same character length (nHxx...x).

For a subscripted parameter name, that parameter must contain at least the number of elements specified within the parentheses. If the subscript (value inside the parentheses) is a parameter name, it must be of integer type and contain only one element.

3.3.3 EVALUATE CS

Purpose: The EVALUATE CS establishes a new user parameter of one element or changes the value of an existing user parameter element.

Format: [label] EVALUATE param-exp $

where param-exp may take any of the following forms:

$pname_1$ = expression

$pname_2\begin{pmatrix} pname_3 \\ integer \end{pmatrix}$ = expression

label label name

$pname_1$ the name of the user parameter for which a value is to be established or changed. The value assigned to $pname_1$ is the result of the evaluated expression. The type assigned to $pname_1$ is the type associated with the expression.

$pname_2$ the name of a previously established user parameter whose specified element will be changed to the result of the evaluated expression. If $pname_3$ is the subscript, it must be of integer type and contain only one element. Otherwise, the subscript must be an integer constant. $Pname_2$ must contain at least the number of elements specified in the subscript. The type associated with the expression must match the type associated with $pname_2$.

$pname_3$ the name of a previously established single-element user parameter of integer type

expression a sequence of constants, user parameters, subscripted user parameters, and function references separated by operators and parentheses

The arithmetic operators are as follows:

+	addition
-	subtraction
*	multiplication
/	division
**	exponentiation

The following functions are available:

Name	Definition	Number of arguments	Type of arguments	Example
ABS	$\lvert X \rvert$	1	Any type	Y=ABS(X)
ANTILOG	10^X	1	I,RS,RD	Y=ANTILOG(X)
COS	cos(X)	1	Any type	Y=COS(X) with X in deg
INT	Express argument as integer	1	Any type	Y=INT(X)
LOG	$\log_{10}(X)$ where $X > 0$	1	I,RS,RD	Y=LOG(X)
REAL	Express argument as real	1	Any type	Y=REAL(X)
SIN	sin(X)	1	Any type	Y=SIN(X) with X in deg
SQRT	$\sqrt{X}$ where $X > 0$	1	Any type	Y=SQRT(X)
TAN	sin(X)/cos(X)	1	Any type	Y=TAN(X) with X in deg

All operations must be specified explicitly. For example, to multiply two variables A and B, the expression A*B must be used. Thus, AB, (A)(D), or A.B will result in a syntax error with the exception of AB which is treated as a parameter name.

An expression is comprised of subexpressions. In the example

A/B - C*D**E

the following are subexpressions:

(1) A/B

(2) D**E

(3) C* $temp_1$ (where $temp_1$ = D**E)

(4) $temp_3$ - $temp_2$ (where $temp_2$ = C*(D**E) and $temp_3$ = A/B)

The sequence in which an expression is evaluated is governed by the following rules which are listed in descending precedence:

(1) Subexpressions delimited by parentheses are evaluated, beginning with the innermost subexpressions.

(2) Subexpressions defined by arithmetic operators are evaluated according to the following precedence hierarchy:

**	exponentiation
/ *	division or multiplication
+ -	addition or subtraction

(3) Subexpressions containing operators of equal precedence (+ and - or * and /) are evaluated from left to right.

(4) Subexpressions containing the exponentiation operator (**) are evaluated from right to left.

In exponentiation, the following types of base and exponent are permitted:

Base	Exponent
Integer	Integer
Real	Integer, real, and double precision
Double precision	Integer, real, and double precision
Complex	Integer

The exponentiation is evaluated from right to left. The expression A**B**C is evaluated as (A**(B**C)).

A subexpression in a given level is evaluated at the highest type (CS RD RS I) of any parameter or constant in that subexpression. In exponentiation both the base and exponent are converted to the highest type in the subexpression with the exception of a complex base. For a complex base, any noninteger-type exponent is converted to an integer type. The type of the expression is the type associated with the last subexpression evaluated.

Examples:

Given the following parameter values, the expression to the left of the equal sign on the EVALUATE CS is evaluated and the result, 43.24, is stored in the user parameter table (UPT) under parameter X:

```
PARAM BAD=16.3, BC=5.3, CD=.96 $
PARAM DE=.3, EF=2.0, F=3 $
EVALUATE X = (BAD - BC)*F + (CD/DE)**EF $
```

The following results in the parameter Z taking the value 3.5 in the UPT:

```
EVALUATE Z = SIN(30.0) + REAL(3) $
```

Given the following parameter values, the following operations are performed. The square root of 100.0 is taken, converted to an integer, and then is added to the negative value of Z. The result, 6.5, is stored as the Ith, or 2nd, element of the parameter Y. The value of the expression must have the same type as Y:

```
PARAM Y=10.0, 3.9, 8.2, I=2 $
EVALUATE Y(I) = -Z + INT(SQRT(100.0)) $
```

Restrictions:

The operators + and - must be followed by a blank. For a subscripted parameter, that parameter must contain at least the number of elements specified within the parentheses.

If a multielement nonsubscripted parameter is in the expression, only the first element is used.

Each left parenthesis must have a corresponding right parenthesis.

The type associated with the expression must match the type associated with the parameter to the left of the equal sign if that parameter is a subscripted parameter.

3.3.4 UPLIST CS

Purpose: The UPLIST control statement will produce a printed list of all user parameters currently defined within the ANOPP run. The list includes each user parameter array's name, the alpha data type, internal ANOPP numeric data-type code, and the number and value of each element in the array.

Format: label UPLIST $

label label name

Example:

The following control statement

LABEL1 UPLIST $

if executed when the user parameters established in a run were those created by the PARAM CS examples and EVALUATE CS examples presented in the two previous sections, would produce the following listing:

USER-PARAMETER-TABLE DUMP

NAME	TYPE[a]	TYPE CODE	ELEMENT	VALUE	
AGROUP	CS	4	(1)	.70000000000000E+01	.17700000000000E+20
			(2)	.26517000000000E+05	-.17500000000000E+01
A	I	1	(1)	1234	
B	RD	3	(1)	.20000000264299999999999999999D+01	
BAD	RS	2	(1)	.16300000000000E+02	
BC	RS	2	(1)	.53000000000000E+01	
C	I	1	(1)	2	
CD	RS	2	(1)	.96000000000000E+00	
D	A	-4	(1)	JET	
DE	RS	2	(1)	.30000000000000E+00	
E	A	-4	(1)	PROP	
EF	RS	2	(1)	.20000000000000E+01	
F	I	1	(1)	3	
G	I	1	(1)	1234	
I	I	1	(1)	2	
KEYS	A	-4	(1)	FAN	
			(2)	JET	
			(3)	PROP	
L	RD	3	(1)	.14999999999999999999999999999D-01	
NUMBS	I	1	(1)	3476	
			(2)	5962	
RESULTS	L	6	(1)	T	
			(2)	F	
STATS	RD	3	(1)	.17669999999999999999999999998D+29	
			(2)	-.99999999999999999999999999975+194	
T	RD	3	(1)	.26429999999999999999999999989D-07	
X	RS	2	(1)	.43240000000000E+02	
Y	RS	2	(1)	.10000000000000E+02	
			(2)	.65000000000000E+01	
			(3)	.82000000000000E+01	
Z	RS	2	(1)	.35000000000000E+01	

[a]TYPE CODE is an integer internal-system-type code. Of interest to the external ANOPP user is the TYPE CODE for type A (Holerith string), expressed as a negative n value, where n is the uniform character length of the string for elements of the user parameter array.

Restrictions: None.

ORIGINAL PAGE IS OF POOR QUALITY

3.4 FUNCTIONAL MODULE (FM) EXECUTION

3.4.1 Overview

A functional module (FM) is a group of subprograms within the ANOPP system that perform a specific noise prediction or utility-type function. The ANOPP concepts applicable to FM's are presented in section 2.2.

Section 4 of this manual contains subsections, each of which documents a specific module available for user execution via an EXECUTE CS. The subsection identifies the FM by ANOPP name, presents a purpose and functional description, and describes each of the user parameters and run-life data base units and members required for its execution. Also documented are the FM characteristics relevant to run environment.

Satisfying FM user parameter requirements.- In the section 4 documentation of each functional module (FM), each required user parameter is identified by its FM internally known name and is documented as to its functional usage, data type, and array size (number of elements). The user parameter may be one of the following: a required input item; an optional input item that, if not currently established in the run at FM execution time, will be established by the FM using documented "default" values; or an output item always established within the FM.

Any user parameters that are required as input to an FM must be established in the run prior to the FM execution. Unless created by a prior execution of another FM within the same run, the user must establish the required parameter via the PARAM CS or EVALUATE CS. The type and array size of the user parameter must match those documented for the FM requirement.

If the name under which a user parameter is established is the same as the FM internal name, that parameter will be automatically accessible within the FM during its execution. However, if the parameter is established under a different name, the alternate name feature of the EXECUTE CS must be used. This capability establishes the required internal FM association of the name of an alternate user parameter candidate with the name by which the parameter is known within the FM.

In the same way, the user may specify that a parameter output by a FM be established under an alternate name instead of the documented FM internal name. Once established within a FM, a user parameter is subject to the same accessing rules as one created via a PARAM CS or EVALUATE CS; in either case, the parameter will be known by its single assigned name for the duration of the run. Section 3.4.3 gives examples of CS streams satisfying FM user parameter requirements.

Satisfying the run-life data-base requirements of a FM.- Section 4 documentation identifies each run-life data base requirement of a FM. A table member or data member is documented and known within the FM by the combination of its resident data unit name and member name. This applies to both FM input and FM output data base items. Section 6 of the manual documents all data items installed on the ANOPP permanent data base by using the same naming convention as described previously. Tables in section 6 indicate which permanent data base items satisfy data requirements for specific FM's; for a primary candidate identified in the tables, the data unit name (member name) usually matches the name combination known within the FM. If a data unit is currently established within a run via a prior CREATE CS, ATTACH CS, or LOAD CS, and the names of the unit and one of its members match the name combination documented as a data requirement within a functional module to be executed, that data item is automatically accessible from within the FM. If the user decides to satisfy a FM data requirement using an alternate item, the alternate-name capability of the EXECUTE CS must be used.

For example, suppose a FM has a data requirement documented as UNIT1(MEMBER1). If the user's alternate candidate existed on the same unit, but as RHOTAB., the EXECUTE CS alternate name specification would include MEMBER1=RHOTAB. However, if the alternate candidate resided on unit JETDAT, but had the same member name as the documented FM requirement (MEMBER1), then the EXECUTE CS specification would include UNIT1=JETDAT. Finally, if both the unit and member names of the alternate item were different from the documented names, the alternate-name specification could include UNIT1=JETDAT, MEMBER1=RHOTAB.

Because of this convention for identifying a run-life data item by the combination of its resident unit and member names, and because two data units of the same name cannot be currently known within the run at the same time, the user must be cautious in his specification of alternate names.

3.4.2 EXECUTE CS

Purpose: The EXECUTE control statement calls a specified functional module (FM) into execution and allows the user to identify alternate data items to be used in satisfying specific data requirements of the functional module.

Format:

[label$_ɓ$]EXECUTE$_ɓ$fmname$_ɓ$[refname$_1$=altname$_1$....$_ɓ$refname$_n$=altname$_n$] $

label label name

fmname the name of a functional module to be executed. A list of functional modules including a brief description is given in appendix E.

refname the name known within the functional module for a required data unit, data member, table member, or user parameter. It is also the name by which the data item is identified in the section 4 documentation of the FM's data requirements.

altname the altname name corresponding to the FM known refname. The altname is the name of a data unit, data member, table member, or user parameter to be used in satisfying the requirement for a data item known within the FM by the refname.

Examples:

```
L1 EXECUTE JET UNIT1=MYUNIT, B=D $

EXECUTE PROP $
```

Restrictions:

The fmname must be a functional module currently installed on the ANOPP system.

User-supplied data items must be of the type and format specified for the FM application.

3.4.3 Examples

The following examples illustrate typical control statements required to satisfy FM data requirements. Functional documentation of the various control statements used are found in other subsections of this manual.

Example 1: The FM JET1 is to be executed. It requires a logical user parameter JSELECT, a real single-precision user parameter NSPEED with three elements, and data unit JETUNIT on which reside members TAB1, TAB2, DMEM1, and DMEM2. No substitute data items are to be used; once established in the run, all required items will be accessible by the FM.

```
STARTCS $
 :
ATTACH JETUNIT/FILE29/ $ COMMENT - ESTABLISH JETUNIT IN RUN
PARAM JSELECT=.TRUE., NSPEED=1.33E05    -0.52    19.7 $
EXECUTE JET1 $
 :
ENDCS $
```

Example 2: Execute FM MACH8. Build a required data member identified in FM documentation as UNIT1 (STATJ) via the CREATE CS and UPDATE CS. Functional module MACH8 creates a logical user parameter documented as RESULT. Because the user has already created a parameter named RESULT earlier in the run for a different purpose, an alternate name specification is used to cause the FM-created user parameter to be called M8RESULT.

```
STARTCS $
 :
CREATE UNIT1 $
UPDATE NEWU=UNIT1 SOURCE=* $
  -ADDR OLDM=*, NEWM=STATJ FORMAT=11H2I,RS,RD,L$ MNR=2 $
  1757 294372 2E+01 3D-01 .FALSE. $
  297 87536 375.15E-12 15.81D19 .TRUE. $
END* $
EXECUTE MACH8 RESULT=M8RESULT $
```

3.5 RUN-LIFE DATA UNITS

The run-life data units are those units known within an ANOPP run at any one point in time; this concept is documented in section 2.3.2.3. Each run-life data unit is currently known within the system data unit directory (DUD) via the processing of a CREATE CS, ATTACH CS, or LOAD CS. A data unit is removed from the run-life data base (and its entry from the DUD) via the processing of a DETACH CS or PURGE CS. Informative reports about the data format and contents of any or all run-life data units or members are provided via the features of the CATALOG CS, MEMLIST CS, or TABLIST CS. A current run-life data unit may be assigned a read-only status, prohibiting further output to the unit even in subsequent runs, via the ARCHIVE CS. The system units DATA, XSUNIT, and XRUNIT (restart runs only) are also run-life data units.

3.5.1 Assignment

The CREATE CS, ATTACH CS, and LOAD CS provide user capabilities to establish the run assignment of a data unit(s). The DETACH CS and PURGE CS provide the capability to terminate the run assignment of a data unit. The LOAD CS establishes the assignment of a sequential library file and, in addition, establishes new run-life data units using data copied from the library file. Section 3.6 documents library-file-related run assignments.

Assignment of an existing data unit, residing on an external random-access file, is accomplished via the ATTACH CS. That data unit must have been either of the following: previously assigned to the run, and then subsequently deassigned from the run via a DETACH CS (freeing its DUD entry); or built during an earlier ANOPP run, and residing on an external file currently assigned to the user's job. If the data unit was created during a previous ANOPP execution, the permanent file on which it resides must be made available to this job via external control cards.

Provided that an existing data unit has not been previously ARCHIVE'd (read-only status), the user may rebuild its data content (members) via UPDATE CS capabilities. When an existing data unit, already resident on a permanent file, is to be rebuilt, additional external control cards may be required to expand the file.

Assignment of a new data unit to be built during the run is performed via the CREATE CS. The user may optionally specify an external file name of the random-access file, on which the new unit will reside, if he intends to establish the file in the host computer's permanent file system (via external control cards). Once assigned within the run, the new data unit is available for the generation of its members via the TABLE CS and UPDATE CS.

Dropping the assignment of a run-life data unit is performed via the DETACH CS or PURGE CS. Both control statements remove the specified currently assigned data unit's entry in the data unit directory (DUD), thus ending the unit's run assignment. Following a unit deassignment via the DETACH CS, the user may still access the data unit via a subsequent ATTACH CS; this reassignment may occur either in the same run or in a second ANOPP run within the user's job. However, the PURGE CS, besides terminating the run assignment of an ANOPP data unit, also terminates the job assignment of the external system data unit's residence file to the user's job. The PURGE'd unit cannot be reaccessed in the same ANOPP run. It may not be accessed in a subsequent ANOPP run in the same job, unless an external control card is processed that reestablishes the external system's job assignment of the data unit's file, prior to initiation of the subsequent ANOPP run.

3.5.1.1 ATTACH CS

Purpose: The ATTACH control statement establishes one or more previously created data units in an ANOPP run via entries in the system data unit directory (DUD). Each data unit must have previously been created on an external mass-storage file currently assigned to the job in the external system.

Format: **[label$_{\not b}$]ATTACH$_{\not b}$dun$_1$/efn$_1$/[...$_{\not b}$dun$_n$/efn$_n$/] $**

label label name

dun the name of the data unit to be established in the ANOPP run

/efn/ the name of the external mass-storage file, currently assigned to the user's job, on which data unit dun resides

Examples:

```
APP ATTACH JETU1/FILEJ/, FANU1/FILEF/ $
ATTACH PROPU1/EFN/ $
```

Restrictions:

Each data unit name (dun) must be unique with respect to all other data unit names currently known in the run.

Each external file name (efn) must be unique with respect to all other efn's currently known in the run.

Each data unit (dun) specified must have previously appeared on a DETACH CS.

If the data unit was created in a previous ANOPP job, the external file (efn) it resides on must be currently assigned to this job via external control cards.

3.5.1.2 CREATE CS

Purpose: The CREATE control statement initially defines in the ANOPP run an empty data unit, to reside on a random-access mass-storage device, and makes it available for subsequently generated table, data, and procedure members.

Format: **[label$_{\not b}$]CREATE$_{\not b}$dun$_1$[/efn$_1$/][...$_{\not b}$dun$_n$ [/efn$_n$/]] $**

label label name

dun the name of the new data unit to be entered in the system data unit directory whose members are to be generated during the run

/efn/ the name of the external mass-storage file, assigned to the job in the external system, on which the corresponding dun will reside. If efn is omitted, a scratch (temporary) file will be established by the ANOPP system.

Examples:

```
LABEL1 CREATE UNIT1, UNIT2/EFN2/, JETUN1/JETFILE/ $
CREATE JOBFILE1, JOBFILE2 $
```

Restrictions:

The dun and efn cannot, respectively, be the same as any other dun and efn that are either currently known to the ANOPP run (entered in the data unit directory) or appear on the same CREATE CS.

Any data that currently reside on external file efn will be destroyed during the creation of new data unit dun, even if the old data were on an ARCHIVE'd ANOPP data unit.

Note: Reference to a unit that has not been defined by a CREATE CS during an ANOPP run will cause the unit to be dynamically created on a scratch file.

3.5.1.3 DETACH CS

Purpose: The DETACH control statement removes a data unit from the run-life data base. Unless the data unit resides on a scratch (temporary) file, it may be assigned to the current or a future ANOPP run via an ATTACH control statement.

Format:

[label$_{\not b}$]DETACH$_{\not b}$dun$_1$[...$_{\not b}$dun$_n$] $

label label name

dun name of the data unit, known to the ANOPP run, that is to be detached

Example:

```
L1 DETACH PROPU1, JETU1 $
DETACH AIRPRTS $
```

Restriction: Each data unit named must be currently known in the ANOPP run (via an entry in the system data unit directory).

3.5.1.4 PURGE CS

Purpose: The PURGE control statement removes a data unit from the run-life data base; in addition, the external file on which the unit resides will no longer be assigned to the user's job in the external system.

Format: **[label$_{\not{b}}$]PURGE$_{\not{b}}$dun$_1$[.....$_{\not{b}}$dun$_n$] $**

label label name

dun name of a data unit whose current run and job assignment is to be terminated

Example:

```
LABEL1 PURGE DU1, DU2, DU3 $
PURGE JETU1 $
```

Restriction: If the data unit is to be re-ATTACH'ed in the same job, or the external file on which the dun resides is to be made permanent via external control cards that follow the ANOPP run (in the user's job deck), the DETACH CS should be used instead of the PURGE CS.

3.5.2 List Options

The user is provided the CATALOG CS, MEMLIST CS, and TABLIST CS, each designed to produce reports on the structure and data content of his run-life data units and their members. In addition, the UPDATE CS (documented in section 3.7.4) provides optional reports on the new members it generates.

Producing data-unit-level reports via the capabilities of the CATALOG CS.- The user may produce a CATALOG report on specifically named data units, or, optionally, on all of the data units currently known in the run-life data base (having an entry in the data unit directory). The user may select either the SUMMARY or the FULL report to be produced for each of the data units indicated on the CATALOG CS. The SUMMARY CATALOG report presents the name of the data unit, the name of its residence external file, and the name of each of its resident members. The FULL CATALOG report presents the above summary information and, in addition, the following information about each member: the current maximum number of records, the member (record) format specification, the maximum record length, and the date and time of member creation.

Producing member-level reports via the capabilities of the MEMLIST CS.- The user may produce a MEMLIST report for any member residing on a run-life data unit. The user may select either the SUMMARY or FULL report to be produced for each member specified on the MEMLIST CS. The SUMMARY MEMLIST report includes, for each specified member, all of the data produced by a FULL CATALOG report. (See the preceding usage discussion.) The FULL MEMLIST report contains the summary information for each specified member, plus the data values of some (specified) or all of each member's records.

Producing table-member level reports via the capabilities of the TABLIST CS.- The user may produce a TABLIST report on any table member residing on a run-life data unit. The user may select either the SUMMARY or FULL report to be produced for each table member specified on the TABLIST CS. The SUMMARY TABLIST report includes for each specified table member all of the data produced by a FULL CATALOG report. In addition, it produces the following information on resident data table: the number of independent variables, the table length in computer words, and the allowable interpolation procedures defined at its creation. The FULL TABLIST report includes, in addition to the above summary data, specific information about the variables defined for each table. For each variable, the type code, number of table values, and the actual assigned values are listed. In addition, for each independent variable, any defined extrapolation procedures are identified.

3.5.2.1 CATALOG CS

Purpose: The CATALOG control statement produces printed summary information about named data units or, optionally, about all data units known to the ANOPP run. In addition, an optional FULL report presents summary information about each resident member of each specified unit.

Format: $\text{[label}_{\not b}\text{]CATALOG}_{\not b}\left[\text{LIST=}\begin{Bmatrix}\text{SUMMARY}\\ \text{FULL}\end{Bmatrix}_{\not b}\right]\text{[du}_1\text{,....,}_{\not b}\text{du}_n\text{]}$ $

label label name

LIST= optional specification allows selection of the type of catalog report desired for the specified du's (data units). If omitted, LIST=FULL is assumed.

LIST=SUMMARY: A SUMMARY CATALOG report will be produced for each data unit specified, and will include:

The data unit name
The name of the external file on which the data unit resides
The name of each member that resides on the unit

LIST=FULL (default): A FULL CATALOG report will be produced for each data unit specified and will include:

The SUMMARY report of items listed above for each member on the data unit:
The maximum number of records allowed
The current number of records
The member (record) format
The maximum record length
The date and time of day the member was created

du (optional) name of a data unit, currently known in the run, about which the user wants a CATALOG report of the selected type. If omitted, the selected report will be produced for all run-life data units, except for system units XSUNIT and XRUNIT (if a restart run). The user may specify: XSUNIT and XRUNIT, in a checkpoint/restart run.

Examples:

The following will produce FULL reports on all run-life data base units, except for system units XSUNIT and XRUNIT:

```
LABEL1 CATALOG $
```

The following will produce a SUMMARY report on data unit DUN7:

```
CATALOG DUN7 LIST=SUMMARY $
```

The following will produce FULL reports on all run-life data base units except for system units XSUNIT and XRUNIT:

```
CATALOG LIST=FULL $
```

The following will produce FULL reports on data units DUN1 and XSUNIT:

```
CATALOG LIST=FULL, DUN1, XSUNIT $
```

Restriction: Specified du's (data units) must be currently established in the run-life data base.

3.5.2.2 MEMLIST CS

Purpose: The MEMLIST control statement produces a printed report about specified members residing on run-life data units. The report may optionally be limited to summary information (member structure and unit residence) on each of the specified members; or the FULL report can be produced to include the summary information plus the listing of values currently stored in the specified members' records.

Format: $[\text{label}_{ȸ}]\text{MEMLIST}_{ȸ}\left[\text{LIST=}\left\{\begin{matrix}\text{FULL}\\ \text{SUMMARY}\end{matrix}\right\}_{ȸ}\right]\text{member}_1[,\ldots,_{ȸ}\text{member}_n]$ $

where each specified member_i combination has the form

$\text{du(mn)}\left[(\text{rn}[,\text{rn}])\right][_{ȸ}\text{FORMAT=format}]$

label label name

LIST= optional keyword allows selection of the type of MEMLIST report desired for the specified members. If omitted, LIST=FULL is assumed.

LIST=SUMMARY: A SUMMARY report will be produced for each member specified and will include:

The name of the data unit on which the member resides
The name of the external file on which the member's data unit resides
The name of the member

The member's:
Maximum number of records
Current number of records
Format specification
Maximum record length
Date of creation
Time of creation

LIST=FULL (default): A FULL MEMLIST report will be produced for each member specified and will include:

The SUMMARY information listed above
The data values of each member record specified by param (rm,rn) below, or all member records if (rm,rn) is omitted; printed in the format specified by keyword FORMAT= below, or in the format under which the member was created if FORMAT= is omitted.

du(mn) the name of the data unit (member) to be reported on

(rm) or **(rm,rn)** the optional specification of the integer sequence number (rm) of the record or the range of integer sequence numbers (rm,rn) of the records on the member whose values are to be listed. If omitted, the values of all the member's records will be displayed. If LIST=SUMMARY, this parameter is ignored.

FORMAT = optional keyword parameter specifies the member format to be used in printing the indicated records. (See table 4 of section 3.7.) This field is ignored if LIST=SUMMARY.

If omitted, the indicated records will be printed in accordance with the format specification under which the member was created via a DATA CS, LOAD CS, TABLE CS, or UPDATE CS. Unformatted records are always printed in octal format.

FORMAT=0 produces an octal printout of the data content of the specified records, regardless of the format under which the member was created.

FORMAT=2HCI indicates that the member has a card-image format; each card-image record will be listed on a single print line.

FORMAT=specified ANOPP format (See table 4 of section 3.7.) A valid ANOPP format specification produces a printout of records in that format. The specified format is assumed to be compatible with the records to be printed and is not validated internally. A member (record) format is expressed as a character string terminating with a $, as shown in the following examples:

```
FORMAT=12H5(I),A30,RS$          (fixed length format)
FORMAT=17HI,2A10,*(I,10RS)$     (variable length format)
```

Examples:

The following will print a SUMMARY report on data members DMEN1 and DMEN2, both resident on data unit DUN1:

```
LABEL1 MEMLIST LIST=SUMMARY, DUN1(DMEN1), DUN1(DMEN2) $
```

The following will print a FULL report on data member DMEN3 of data unit DUN1. Each record on the member will be listed, and the data values will be displayed in octal code.

```
MEMLIST LIST=FULL, DUN1(DMEN3) FORMAT=0 $
```

The following will print a FULL report on data member DM9 of data unit DU1, listing the values of member records 11 and 12 in the format under which the data member was created:

```
MEMLIST DU1(DM9), (11,12) $
```

Restrictions:

Specified data units and members, du(mn), must be currently assigned to the run-life data base.

Specified member record occurrences, (rm) or (rm,rn), must be valid for the member being listed.

3.5.2.3 TABLIST CS

Purpose: The TABLIST control statement produces a printed report containing summary information about the ANOPP data tables that reside on specified run-life members. In addition, a FULL TABLIST report may be selected that includes the listing of the specified tables' assigned values. (Section 3.7.3 documents the ANOPP table types that may be created on a table member.)

Format: $[\text{label}_{\cancel{b}}]\text{TABLIST}_{\cancel{b}}\left[\text{LIST=}\begin{Bmatrix}\text{FULL}\\ \text{SUMMARY}\end{Bmatrix}_{\cancel{b}}\right]du_1(tm_1)[\ldots,_{\cancel{b}}du_n(tm_n)]\ \$$

label label name

LIST= optional keyword allows selection of the type of TABLIST report desired for the specified member resident tables. If omitted, LIST=SUMMARY is assumed.

LIST=SUMMARY (default): A SUMMARY report will be printed about the ANOPP data table that resides on each specified data unit. The SUMMARY report includes the following:

Residence data unit name
Residence external file name
Residence table member name
Creation date
Creation time of day

Number of independent variables
Length in words
Allowable interpolation procedures

LIST=FULL: A FULL TABLIST report will be printed on each specified table and will include:

The SUMMARY report information (above)
The following data on ~~each independ~~ent variable defined for the table:

The type code
The number of table values
The procedure for extrapolation beyond the last assigned value
The assigned values

and the following data on the dependent variable:

The type code
The number of table values
The assigned values

du(tm) the run-life data unit name (table member name) on which the data table to be reported on resides

Examples:

The following will produce a FULL TABLIST report on data unit DUN1 resident table-member DTBL1:

```
LABEL1 TABLIST LIST=FULL, DUN1(DTBL1) $
```

The following will produce a SUMMARY report on data unit DUN2 resident table member DTBL2:

```
TABLIST LIST=SUMMARY DUN2(DTBL2) $
```

The following will produce a SUMMARY report on data unit DUN1 resident table members TBL1 and TBL2, and on data unit DUN3 resident table member TBL1:

```
TABLIST DUN1(TBL1), DUN1(TBL2), DUN3(TBL1) $
```

Restrictions:

The ANOPP table residing on a specified member must have been created either via a TABLE CS or internally by a functional module.

Each specified-table residence data unit (table member) must currently be assigned to the run.

3.5.3 Archiving (ARCHIVE CS)

The ARCHIVE CS is used to assign a permanent ANOPP read-only status to an existing data unit. It is intended for the user who has built a data unit and now wants to prevent any possibility of a future change being made to the data content of that unit as a result of subsequent TABLE CS, UPDATE CS, or functional module processing. The archive (read-only) status, once established, is recorded in the internal header of the unit and cannot be reversed during the current or any future ANOPP run or job.

Purpose: The ARCHIVE control statement prohibits any future request to write on the named unit or units via TABLE CS, UPDATE CS, or functional module processing, whether in the current run or in any subsequent ANOPP run or job.

Format: **[label$_{\not b}$]ARCHIVE$_{\not b}$dun$_1$[......$_{\not b}$dun$_n$] $**

label label name

dun name of the data unit to be archived

Examples:

```
LAB ARCHIVE UN1, UN2, UN20 $
ARCHIVE, U50 $
```

Restrictions:

Each unit name must be unique with respect to the other unit name(s) on the control statement.

Each unit to be archived must be currently assigned in the ANOPP run.

The archive (read-only) status prevents data unit content modification via subsequent TABLE CS, UPDATE CS, or functional module processing. It does not prevent either the modification of the content of the file via subsequent external (non-ANOPP) processing or the creation of a new unit on the file via CREATE CS or LOAD CS processing.

ANOPP system scratch units (a unit whose associated external file name begins with the letters "XEFN") may not be archived.

3.6 SEQUENTIAL LIBRARY FILES

An ANOPP sequential library file is used to store previously established run-life data units and their members. The data are unloaded onto a sequential mass storage or tape file for the following reasons:

(1) To create a data base backup copy

(2) To create a transferable copy of data base units

(3) To group a set of data base units

The ANOPP library file is identified by its external file name. It does not possess an ANOPP name that differs from its external file name, as does a data unit.

In section 6 the sequential library files available to the user are identified. The grouping of FM required data base items onto a single library reduces the number of external control cards required for file assignments of a job. Cross-reference tables in section 6 relate specific FM data base item requirements to the permanent data unit files and/or library files they reside on. The combined capabilities of the LOAD CS, UNLOAD CS, and DROP CS provide for library file creation, run assignment and deassignment, and stored-data retrieval.

Assigning a sequential library file.- Just as a data unit is known to the run via its entry in the data unit directory, a library file is known via its entry in the system's library file directory. The first LOAD CS or UNLOAD CS processed in a run that specifies the name of a particular sequential library file establishes the library-file-directory (LFD) entry. The LOAD CS must identify as its data source a library file that was previously created via an UNLOAD CS, either in the current run or in an earlier run. The UNLOAD CS must specify as its library file a sequential external file name not currently known to the run. (An UNLOAD CS may not specify a library file that is currently known in the run via an earlier LOAD or UNLOAD CS.)

Terminating the run and job assignment of a sequential library file.- The DROP CS terminates the run assignment and job assignment of specified library files by removing their entries from the system library file directory. Once a library file has been dropped from a run, that physical external file is no longer accessible in the same run. In a multirun ANOPP job, a dropped library file is unaccessible until its job assignment is reestablished via external control cards (processed prior to the initiation of the subsequent ANOPP run).

Establishing run-life data units using library file resident data.- The LOAD CS provides the capability to add new run-life data units, using as a source of data the units and members stored on a library file. The data units stored on a run-assigned library file are not run-life data units; they are copies of previously established run-life units. The LOAD CS allows the user to add new run-life data units (none of which may be currently established in the run). A new unit may be an exact copy (optionally renamed) of a library stored unit, or it may be generated by using combinations of members (each optionally renamed) that reside on any of the units stored on the library. More than one LOAD CS may use the same library file as its data source.

Building a new sequential library file.- The UNLOAD CS provides the capability to build library files. The UNLOAD CS must specify the name of a sequential external file name (sefn) that is assigned to the user's job and that is not currently known in the run via a previous LOAD or UNLOAD CS. The UNLOAD CS allows the user to do the following:

(1) To copy any or all of the current run-life data units to the library

(2) To limit the library file copy of ~~a run~~-life data unit to be a subset of its members

Each member copied to the library will be stored under its run-life data unit name (member name) combination.

The user wishing to rebuild an existing library is advised to use the following sequence of steps:

(1) Perform a LOAD CS or series of LOAD CS's, by using the old library file as the source of data and utilizing LOAD CS member-regrouping capabilities if required.

(2) Perform any additional data modification required by using the UPDATE CS capabilities (see section 3.7.4).

(3) Build the new version of the library via an UNLOAD CS specifying a new library file name.

(4) Use external control cards either to purge the old external library file and retain the new one or to retain the new library as a new version (cycle) of the original file.

3.6.1 LOAD CS

Purpose: The LOAD control statement creates new run-life data units by using as a source of data the units and members previously stored on a sequential library file via an UNLOAD CS (in either the current or a previous ANOPP run). The specified library file must be currently assigned to the user's job in the external system. The LOAD CS provides for selective loading and/or renaming of data units and data members stored on the sequential library file.

Format: **[label$_{\not{b}}$]LOAD/sefn/[du$_1$,....,du$_n$] $**

where each du$_i$ has the format

ndu[[/efn/][=odu] [(mn$_1$[,..mn$_n$])]]

where each mn$_i$ has the format

nmn=omn

label label name

sefn the name of a sequential library file, currently assigned to the user's job in the external system, from which stored data unit (member) data will be copied ndu (optional) the name of a new data unit to be established in the run-life data base. If members residing on more than one LF stored data unit (odu) are to be copied to the same new ndu, repeat specification of this field is required. If no ndu is specified, all of the library resident units will be established in the run-life data base on scratch files with no change in names or data contents.

/efn/ (optional) the name of the external random access file, assigned to the user's job, on which the new data unit will reside. If omitted, a scratch (temporary) file will be used for that ndu. If more than one LF stored data unit (odu) is to supply members for the new du, the /efn/ specified for that new ndu should occur only once on the LOAD CS.

=odu (optional) the name of the library resident data unit whose members are to be copied to the new du. If omitted, the library resident data unit (odu) and the run-life data unit (ndu) will have the same name.

nmn (optional) the name of the member to be created on the ndu. If no nmn is specified, all odu members are copied to ndu.

=omn (optional) the name of library unit odu resident member to be copied to the ndu member nmn. If omitted, the odu resident member (omn) and the run-life member (nmn) will have the same name.

Examples:

The following examples will illustrate the various features of the LOAD CS. Each example will use a previously created sequential library file named OLDLF, assumed to be currently assigned to the user's job in the external system. The OLDLF was created by an UNLOAD CS that stored copies of data units UNIT1, UNIT2, UNIT3, and UNIT4; each stored data unit contains members MEM1, MEM2, MEM3, and MEM4.

Example 1: The following would result in all four OLDLF stored units being established as run-life data units under their stored names with no change in their member content. All of the new units would be created on scratch (temporary) external files since no efn's were specified.

```
LABELA LOAD/OLDLF/ $
```

Example 2: The following would result in new run-life data unit UNIT3 being established on external file FILEB. The UNIT3 would contain MEM1, MEM2, MEM3, and MEM4.

```
LOAD/OLDLF/UNIT3/FILEB/ $
```

Example 3: The following would result in new run-life data unit JETU being established on external file FILEC. Unit JETU would contain members (MEM1 (a copy of library stored member UNIT4 (MEM1)) and member ALTTBL (a renamed copy of library stored member UNIT4 (MEM2)).

```
LOAD/OLDLF/JETU/FILEC/=UNIT4 (MEM1, ALTTBL=MEM2) $
```

Example 4: The following would result in the establishment of new run-life data units JU and PU. Unit JU would reside on external file JFILE and would be a renamed copy of OLDLF stored unit UNIT1. Unit PU would reside on external file PFILE and contain six members. The PU members MEM1 through MEM4 would be exact copies of OLDLF stored unit UNIT2's four members. The PU members MEM5 and MEM6 would, respectively, be renamed copies of members MEM1 and MEM2 of OLDLF stored unit UNIT3. Note that /PFILE/, the external file name for new unit PU, is specified only once.

```
LOAD/OLDLF/JU/JFILE/=UNIT1
          PU/PFILE/=UNIT2
          PU=UNIT3 (MEM5=MEM1, MEM6=MEM2) $
```

Restrictions:

Each specified new run-life data unit (ndu) and resident external file name /efn/ must be currently unknown to the run.

Each specified /efn/ and the external library file (/sefn/) must be assigned to the user's job in the external system. Any data currently residing on file efn will be destroyed during the creation of run-life unit ndu.

Each specified library file must have been previously created via an UNLOAD CS, either in the current or a prior ANOPP run.

Each new member must possess a name unique as to the names of all other members created on the new unit.

No member residing on the library file may be the data source for more than one new run-life member created by a single LOAD CS. (See example 4.) Following the specification of JU/JFILE/=UNIT1, no subsequent data source specified on that LOAD CS, for any new data unit, could be a member of OLDLF stored unit UNIT1.

3.6.2 UNLOAD CS

Purpose: The UNLOAD control statement establishes a new ANOPP library file on a sequential external tape or mass-storage file. Copies of current run-life data units, optionally consisting of all or some (specified) of their resident members, are unloaded onto the library file. Once established, the library file is available as a data source for subsequent LOAD CS applications within the current or a future ANOPP run.

Format: [label$_{\not b}$]UNLOAD/sefn/[du_1,.....du_n] $

where each du_i has the format

du(mn_1,....mn_n)

label label name

/sefn/ the name of the sequential external tape or mass-storage file, assigned to the user's job in the external system, onto which the run-life unit (member) data are to be copied. The sefn must not be currently known in the run via a previous UNLOAD CS or LOAD CS.

du the name of a current run-life data unit whose member data are to be copied (unloaded) onto the sefn. If no du is specified, all run-life data base units (except system units DATA, XSUNIT, and XRUNIT) will be copied to the library file.

mn the name of a specified du resident member that is to be copied to the library. If one or more mn's are specified, only those named members are copied from that du. If no mn list is specified for a du, all of that unit's members are copied to the library.

Examples:

All data units currently known to the ANOPP run, except for system data units XSUNIT, DATA, and XRUNIT, will be unloaded onto the following sequential library file TAPE1:

```
LABEL1 UNLOAD/TAPE1/ $
```

All run-life data units UNIT1 and UNIT3 will be unloaded as follows; UNIT2 will be unloaded but will contain only members MEM3 and MEM5:

```
UNLOAD/BACKUP/UNIT1, UNIT2(MEM3, MEM5),UNIT3 $
```

Restrictions:

No duplication of a combination of data unit and data member is allowed on the UNLOAD CS. If a data unit appears more than once on the same UNLOAD CS, each occurrence must specify a unique (for that du) member list.

Once a library file has been established in a run via either a LOAD CS or UNLOAD CS, a subsequent UNLOAD CS may not address the same file in the same run unless it has been removed from the system by using the DROP CS.

3.6.3 DROP CS

Purpose: The DROP control statement terminates the ANOPP run assignment and the job assignment of the external system of a sequential library file. The dropped library file will be unaccessible for the rest of the run and unaccessible for the rest of the user's job unless its job assignment is reestablished via subsequently processed external control cards.

Format: **[label,]DROP /sefn/[,..../sefn/] $**

label label name

sefn the external file name of a run-life sequential library file whose run and job assignments are to be dropped

Examples:

```
END DROP /TAPE1/,/DISK1/,/JETSET/ $
DROP /MASTERF/ $
```

Restrictions:

When a tape resident library file is dropped, the tape is physically unloaded by the external system.

When a mass-storage resident library file is dropped, that file is no longer assigned to the user's job in the external system.

A new library file (created via an UNLOAD CS) must not be dropped if the user intends to reference that file in the external system via subsequently processed external control cards.

Once a specific library file (sefn) has been dropped in a user's job, the same sefn specification may appear on a subsequent UNLOAD CS. However, unless the job assignment of the dropped file was reestablished via external control cards, the subsequent UNLOAD CS will not address the same physical file that was dropped.

3.7 MEMBER GENERATION

New members are generated (established on run-life data units) via the capabilities of the DATA CS, TABLE CS, UPDATE CS, and LOAD CS. The LOAD CS generates new run-life members and units by copying data stored on an ANOPP sequential library file. A library file is created via the UNLOAD CS and contains complete and/or partial copies of previous run-life data units. Section 3.6 documents the member generation capabilities of the LOAD CS. The DATA, TABLE, and UPDATE control statements allow the user the capability to define the individual records to reside on new members. All three CS's either require or allow the user to input card-image data directly from within his primary control statement stream. This card-image data are either a set of data records to reside on a new member or a set of CS subordinate directives that define new units, members, or records being generated. When the set of card images is included in the primary stream, it follows immediately behind the control statement requiring it. In this case, the card-image set is terminated in the CS stream by a special END* Input Terminator card.

The DATA CS creates a new member, or recreates an existing member, that resides on the system work unit DATA. The member will be comprised of card-image format records that were read directly from the user's primary control statement stream.

The TABLE CS creates a new table member which is defined by a set of subordinate card-image directives. The directives may either be read from the user's primary input stream or from a run-life member containing the card-image directives. The table member may be created on system unit DATA or any user-established run-life data unit.

The UPDATE CS is a primary tool provided for the generation of new run-life members on a specified data unit. The specified unit must be either a new run-life unit (established via a CREATE CS) or an existing run-life unit (established via an ATTACH or LOAD CS) that will be wiped clean (zero resident members) at the initiation of the UPDATE CS processing phase. The new members may be assigned any ANOPP format specification. They may consist of records either read from any other run-life member or from within the user's primary input stream.

Generation of a procedure member.- A procedure member consists of a set of card-image records that are valid ANOPP user control statements. A procedure member becomes a secondary CS stream when brought into execution via a CALL CS. The UPDATE CS or DATA CS provide the capability to generate card-image formatted records. In addition, UPDATE allows for the record-level modification (card-image deletion and insertion) of an existing procedure member input stream during the generation of a new procedure member.

Generation of table members.- A data table is a specially formatted single record member generated by a TABLE CS. It contains the independent and dependent table values for table lookup and/or interpolation required by a functional module.

Generation of data members.- Data members are members other than procedure or table members. The member may be unformatted or of a card-image format, fixed-length format, or variable-length format. The UPDATE CS allows the user to generate a data member of any valid format type. The DATA CS allows for the generation of a card-image data member.

Member format.- The structure of all records on a member is defined by the member format specification. Table 4 presents the specification rules for the various types of member formats.

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 4.- MEMBER-FORMAT SPECIFICATION RULES

Format type	Format-type specification rules	Examples
Unformatted	An unformatted member requires the integer 0 (zero).	FORMAT=0
Card image	A card-image formatted member requires the Hollerith character string 2HCI.	FORMAT=2HCI
Fixed-length format	A member containing formatted fixed-length records requires a Hollerith string, the last character of which is a $. The basic specification form is nHet[...,et]$ where et is an ANOPP alpha data-type specification of I, RS, RD, CS, L, or An that indicates a data element type occurring within each of the member's records. (See table 1 in section 3.1.2.) When two or more elements of the same type or two or more series of elements of the same types occur within the format of a record, they may be shown by using a group specification of net or n(et,...,et). Integer n indicates the number of times the element type or series of element types repeat. Every second example shown to the right is a respecification of the same format as the example preceding it, but utilizing group specifications.	FORMAT=11HI,CS,RS,RS$ FORMAT=9HI,CS,2RS$ FORMAT=15HA9,A9,A9,CS,CS$ FORMAT=8H3A9,2CS$ FORMAT=14HA9,L,I,A9,L,I$ FORMAT=10H2(A9,L,I)$
Variable-length format	Variable-length records always contain a trailing portion of one or more elements which may occur from zero to an undefined number of times. If specified, each variable-length record will contain a leading portion of one or more elements. The * is used as a special character within variable-length format specifications to indicate the beginning of the variable portion of its records. The basic specification form is nH[et...,et]*(et]...,et])$ The element-type specification rules, presented above for fixed-length records, also apply within the fixed and variable portions of variable-length format specifications. The parentheses are required if more than one element exists in the variable portion. NOTE: For each occurrence of the variable portion in any variable length record, each element specified as part of that portion must exist.	FORMAT=4H*RS$ FORMAT=6H*(RD)$ FORMAT=8H*(RS,I)$ FORMAT=12H*(2A8,10RS)$ FORMAT=13H*(2(L,10RS))$ FORMAT=11HA8,*(25RS)$ FORMAT=12HL,I,*(10CS)$ FORMAT=15HA7,*(I,CS,5RS)$

ORIGINAL PAGE IS
OF POOR QUALITY

3.7.1 END* Input Terminator

Purpose: The END* input terminator card indicates the end of a primary input stream card-image set required by the previous DATA, UPDATE, or TABLE control statement.

Format: **END*** $

Examples:

See DATA CS, UPDATE CS, and TABLE CS for specific usage examples.

Restrictions:

The END* CS is valid in the primary input stream only.

A label field is illegal on the END* card.

3.7.2 DATA CS

Purpose: The DATA control statement creates or recreates a card-image (CI) formatted member on system work unit DATA. The created member may be a new version of a member created via a previous DATA CS in the same run. The records of the member will be the card images that immediately follow the DATA CS in the user's primary input stream. This input card-image set is terminated by the first END* card encountered following the DATA CS.

Format: **[label]DATA DM=mn $**

label label name

DM=mn name of the card-image member to be created on system data unit DATA

Examples:

```
LABELA DATA DM=FM1DATA $
1     2     3     4     5     .TRUE.
1.5   2.5   3.4   4.5   5.5   .TRUE.
END* $

DATA DM=PROCMEM $
     LOAD/ANOPDAT/COANJET, JRS $
     EXECUTE FMNEWJ UNIT1=COANJET UNIT2=JRS $
     DETACH COANJET JRS $
END* $
```

Restrictions:

The DATA CS is legal only in the user's primary input stream.

The terminating END* card will not be a card image on the created member.

3.7.3 TABLE CS

Purpose: The TABLE control statement builds a user-specified ANOPP data table on a run-life table member. The data table is structured in accordance with a set of user-supplied table-description card images. The card-image set may reside immediately following the TABLE CS in the user's primary input stream or on a user-specified card-image run-life member. Description cards required for the definition of ANOPP data table types are documented below.

Format: **[label␢]TABLE␢du(tm)␢type␢SOURCE=⟨* / dun(mn)⟩ $**

label label name

du(tm) the data unit name (table member name) on which the data table is to reside. The du may be any user-established run-life unit or system unit DATA.

type the integer ANOPP table type of the data table being built. Current valid table types are documented below.

SOURCE= specifies where the table-description card-image set, required to define the ANOPP table-type specified, resides. Table-description card sets are documented below.

SOURCE=* indicates that the card-image set follows immediately behind the TABLE CS in the user's primary input stream. The set must be terminated by an END* input terminator card.

SOURCE=dun(mn) indicates run-life data unit name (member name) on which the card-image table-description set resides.

Examples:

A type-1 data table will be built on unit FMIUNIT resident table member FANTAB. The table-description set will be read from card-image member FANTDCS residing on system unit DATA as follows:

```
LABELA TABLE FMIUNIT(FANTAB), 1, SOURCE=DATA(FANTDCS) $
```

The same table creation results as for example 1 in section 3.6.1, but here the table-description set is read from the user's primary input stream and is terminated by an END* card as follows:

```
TABLE FMIUNIT(FANTAB)  1  SOURCE=* $
    INT= .....   )
    IND1=.....   }  table-type-1 description card set
    DEP= .....   )
END* $
```

Restriction: A TABLE CS having a specification of SOURCE=* is valid only in the user's primary input stream; specifications of SOURCE=dun(mn) are valid in any CS stream.

ANOPP data table type-1 description cards: The concepts and definition of an ANOPP type-1 table are presented in section 2.3.1.2.2. The table type-1 description cards are of the following format:

INT= ipc, ipc, ipc
IND_n= ifc,nv,exu,exl,$ival_1$,....,$ival_{nv}$
DEP= dfc,$dval_1$,$dval_2$.....

which are defined as follows:

INT card — contains the integer codes for the interpolation procedure that will be permitted on this table

- ipc — integer code:
  - 0 no interpolation
  - 1 linear interpolation
  - 2 cubic-spline interpolation

IND_n card — contains the description of and the table values for the nth independent variable where $n \leq 4$

- ifc — (format code) the alpha data-type code of the nth independent variable
  - 0 ordered position from 1 to nv; independent variable values not entered
  - I integer
  - RS real single precision
  - RD real double precision

- nv — number of values for the nth independent variable

- exu — integer code for the extrapolation procedure to be used (during interpolation) if a specified value for this independent variable falls beyond the last table value ($ival_{nv}$) for the independent variable
  - 0 no extrapolation allowed
  - 1 use the independent-variable table value closest to the specified value
  - 2 extrapolation is linear when using the last two table values for the independent variable

- exl — integer code for the extrapolation procedure to be used (during interpolation) if a specified value for this independent variable falls before the first table value ($ival_1$) for the independent variable
  - 0 no extrapolation allowed
  - 1 use the table value of the independent variable closest to the specified value
  - 2 extrapolation is linear when using the first two table values for the independent variable

ival   nv table values for the nth independent variable in ascending or descending order. Values are separated by blank or comma and may extend over several card images. If ifc=0, values are not included. See table 1 in section 3.1.2 for numeric-value specification examples.

NOTE: If only a single value is given for the independent variable (nv=1) and either exu or exl equals 2, the system will print an informative diagnostic and build the table with the extrapolation param (exu or exl) set to 1.-

DEP card   contains the description and table values for the dependent variable and must follow the IND_n cards

dfc   (format code) alpha data type of the dependent variable
- I integer
- RS real single precision
- RD real double precision
- CS complex single precision

dval   the table value for the dependent variable, separated by commas or blanks, and extending over several card images as required. The order of dependent variables is such that the first independent variable varies first, the second variable varies second, the third variable varies third, and the fourth variable varies last. See table 1 in section 3.1.2 for numeric-value specification examples.

END* card   required if SOURCE=*.

Examples of TABLE CS type-1 table descriptions:

```
LAB TABLE UNI(DMS),1,SOURCE=* $
INT=0,1
IND2=I,2,0,1,5,10
IND1=RS,3,0,1,1.5,2.0,4.5
DEP=I,3,5,7,8,9,10
END* $

TABLE UNI(DMT) 1 SOURCE=* $
INT=0 1
IND1=I 2 1 1 2 3
IND2=RS 2 2 2 3.4 7.8
IND3=RD 2 2 2 4DO 3DO
IND4=RS 2 2 2 3. 2.
DEP=CS (12.,12.) (13.,13.) (13.,13.) (14.,14.) (11.,11.)
       (12.,12.) (12.,12.) (13.,13.) (11.,11.) (12.,12.)
       (12.,12.) (13.,13.) (10.,10.) (11.,11.) (11.,11.) (12.,12.)
END* $
```

NOTE: Currently, the only valid data-table type recognized by ANOPP is type 1.

3.7.4 The UPDATE Processing Phase

3.7.4.1 Overview

The processing of an UPDATE CS in the user's input stream initiates the update processing phase. During this phase, the system will generate all new members onto a specified run-life data unit. During the update phase, and in accordance with a user's set of update directives, the system will generate the new unit members by:

(1) Copying the contents of specified existing run-life members

(2) Copying new-member data records directly from the primary input stream

(3) Building new members by combining records copied from an existing run-life member and from the user's update directive set

The UPDATE CS that initiates the update processing phase specifies:

The new unit to receive the generated members
The source where the card-image update-directive set resides
The processing mode (create or revise) for this update phase
The content of the update processing report

The new unit to be built during the update phase should have been previously established in the run via a CREATE, ATTACH, or LOAD CS. At the start of the update phase, any resident members on the user-specified new unit are destroyed. The update-directive set is a set of card images that either resides on a user-specified run-life member or that immediately follows the UPDATE CS in the user's primary input stream (an END* input terminator card establishing the end of the directive set). The update directives allow the user to define the new members to reside on the unit being built and to specify the source of each new member. The update processing mode is determined by the user's inclusion or omission of a primary old data unit specification on his UPDATE CS.

The update create mode is in effect if a primary old unit is not specified on the user's UPDATE CS. Under this mode, the -ADDR member-level directive may be used to generate new members copied from specified run-life units or from within the input stream. The -CHANGE member-level directive, and its subordinate record-level directives, may be used to mix existing and new records onto a new member. In create mode the user must always specify on each member-level directive the run-life unit residence of any data-source member. Each new member must be generated by a single member-level directive application.

The update revise mode is in effect if the user specifies a primary old unit on his UPDATE CS. A specified primary old unit should be the primary source of run-life member and record data for the new unit being built. Typically, this is an existing unit, the content of which is being revised or updated. UPDATE CS applications never result in a change of data occurring on the old source unit or member. Under the revise mode, the user has access to all the directive capabilities available under the create mode, including the use of any run-life member as a record source. In addition, he has access to two additional member-level directives, the -COPY and -OMIT directives. The -COPY directive allows for copying (unchanged) specified members from the primary old data unit. The -OMIT directive allows for specific identification of primary old data-unit resident members that are not to be copied to the new unit. The -COPY and -OMIT directives, used in conjunction with the optional UPDATE CS keyword ALL, allow the user to copy by default all the primary old unit members not specifically omitted, copied, replaced, or changed by a member-level directive. Revise-mode processing also allows the user to default on his data-source unit specification on most update directives, in which case the primary old unit is assumed.

A printed update processing report is produced by the system to describe the data unit, member, and record generation that occurred during the processing phase just completed. The report will contain as a minimum the header section described below. The optional LIST= parameter of the UPDATE CS allows the user to select additional sectional content to appear in the report. He may specify any combination of the four alpha characters, each of which requests a different titled update report section to be printed. The report sections, the LIST= value specification that selects them, and their information content are given as follows:

Header section:
- Printed following the completion of each update processing phase
- Contains:
  - The update mode (create or revise) in effect during the processing phase
  - The name of the new data unit generated
  - If revise mode, the name of the primary old unit used
  - The source from which the update directive set was read, either the user's primary input stream or a run-life data unit name (member name)
  - The user specified UPDATE CS LIST= selections

Directive echo section:
- Included in report if LIST= selection contains letter E
- Prints each card image in the user's update directive set

Summary section:
- Included in report if LIST= selection contains letter S
- Contains:
  - The total number of members generated on the new data unit
  - For each new member generated:
    - The name of the member
    - The number of records on the member
    - The format specification of the member (records)
    - The maximum record length (number of computer storage words)
    - The type of the update member-level directive (-COPY, -ADDR, or -CHANGE), or the UPDATE CS keyword ALL that controlled the generation of the member

ADDR member section:
  Included in report if LIST= selection contains letter A
  Contains, for each member generated by a user's -ADDR member-level directive:
    The name of the member
    The format specification of the member (records)
    A list containing the stored data in each of the member's records, presented in either:
      Card-image (10A8) form if member is type CI (card image)
      Octal code, five internal computer words per line, if member is type UNFORMATTED or FORMATTED

CHANGE member section:
  Included in report if LIST= selection contains letter C
  Contains, for each member generated by a user's -CHANGE member-level directive (and its subordinate record-level directives), the same information as presented on ADDR member-section reported members

The typical ANOPP user will build data units via the UPDATE CS specifying either an existing run-life member or a set of input stream records as the only source of data for each new member generated. Since the -ADDR member-level directive will be their primary tool, these users may wish to skip the documentation of the other directives. The large ANOPP application user requires the capability to perform record-level member modification and generation. He will use the -COPY, -OMIT, and -CHANGE member-level directives and the subordinate -INSERT, -DELETE, and -QUIT record-level directives.

3.7.4.2 UPDATE CS

Purpose: The UPDATE control statement initiates the ANOPP update processing phase, during which new resident members are generated onto a user-specified run-life data unit. The UPDATE CS requires a set of subordinate update directives, read either from within the user's primary input stream or from a specified card-image run-life member. The update directives allow the user to copy specific old members from run-life units onto the new unit; to add new members consisting of records read from within the user's update directive set; and to build new members by using selective copying, insertion, and omission of records residing on an old run-life member and/or new user input records. At the initiation of the update processing phase, the specified run-life unit being built (or rebuilt) is wiped clean.

Format:

[label$_{\not b}$]UPDATE$_{\not b}$[OLDU=pdu$_{\not b}$]NEWU=ndu$_{\not b}$[ALL$_{\not b}$]SOURCE=$\left\langle \mathrm{d(mn)} \right\rangle_{\not b}$[LIST=x...x] $

label label name

OLDU=pdu (optional) keyword specifies the name of a run-life data unit (pdu) as the primary source of member and record data for the new member ndu being generated

If specified, the update revise mode will be in effect: the -COPY and -OMIT update member-level directives may be used; the UPDATE CS keyword parameter ALL may be specified; and record-source data unit specifications on certain directives may be defaulted to primary unit pdu.

If omitted, the update create mode will be in effect: the -COPY and -OMIT directives may not be used, and the UPDATE CS keyword parameter ALL is ignored by the system.

NEWU=ndu required keyword parameter specifies the name (ndu) of the run-life data unit onto which new members are to be generated. Any members resident on the unit will be destroyed at initiation of the update processing phase.

ALL (optional) keyword is meaningful only in update revise mode processing.

If specified under this mode, the final-member generation step performed in the processing phase will be a copy to the new unit of every member resident on the specified primary old data unit (OLDU=pdu), except for:

A member that was specified as the data-source member via the OLDM= keyword parameter on a -ADDR or -CHANGE directive

A member that was specified on a -COPY or -OMIT directive

A member whose name matches the name of a new member generated on the new unit via a -ADDR or -CHANGE directive

SOURCE= this keyword specification identifies where the card-image update directive set resides

SOURCE=* is legal only on an UPDATE CS occurring in the user's primary input stream. It indicates that the directive-set card images follow immediately behind the UPDATE CS. Directive-set termination is indicated by the occurrence of an END* input terminator card. (See section 3.7.1.)

SOURCE=dun(dmn) is a legal specification on any UPDATE CS. It indicates that the update directive set resides on run-life unit dun (card-image member dmn).

LIST=x...x (optional) value assigned must be a list from one to four alpha characters. Each valid character in the specified list results in a specific update report section being printed. The information content of each section is documented in section 3.7.4.1. If LIST= is omitted, the update report will contain only the header section. The valid list characters and the report sections they generate are:

E Directive echo section

S Summary section

C CHANGE member section
A ADDR member section

__Examples:__

The following run-life unit U1 will contain the new members generated in accordance with the update directive set. Since no primary data-source unit was specified (OLDU=), only create mode valid directives may be used. An update processing report will be printed which contains all sections except for the directive echo section.

```
LAB1 UPDATE NEWU=U1, SOURCE=*, LIST=SCA $
     -ADDR.........  member-level directives
     -CHANGE.......                             Primary input stream
         -INSERT...                             update directive set
         -DELETE...  record-level directives
         -QUIT.....
END* $
```

The following run-life unit NEWFAN will contain the new members generated in accordance with the card-image directive set residing on unit U7(member MEM9). The update revise mode is in effect with run-life unit FANU1 identified as the primary data-source unit; all update directives may be used. The ALL keyword specification will result in all unit FANU1 members that were not specified on any member-level directive being copied to unit NEWFAN. The omission of the LIST= keyword will result in an update report consisting only of the header section.

```
UPDATE OLDU=FANU1 NEWU=NEWFAN ALL SOURCE=U7(MEM9) $
```

NOTE: Examples illustrating the use of the UPDATE CS and specific member-level and record-level directives are presented in section 3.7.4.5.

3.7.4.3 Member-Level Directives

The general format of update member-level directives is

directivename$_{b}$field$_{b}$....field$_{d}$ $

The valid directivenames are as follows:

-COPY (valid in update revise mode only)
-OMIT (valid in update revise mode only)
-ADDR (valid in all update processing modes)
-CHANGE (valid in all update processing modes)

The fields are as specified in the following subsections that document the individual directives. Field separation, continuation, and value specification rules are as documented for ANOPP control statements in section 3.1.2. Labels are illegal on member-level directives.

3.7.4.3.1 -COPY directive

Purpose: The -COPY update member-level directive will generate one or more user-specified new unit members. Each new member is an exact copy of the member of the same name residing on the primary old unit (identified by the UPDATE CS keyword OLDU=).

Format: -COPY␢mn[,...,␢mn] $

mn the name of the member to be generated on the new unit that will be an exact copy of the primary old-unit (OLDU=) resident member of the same name

Examples:

```
-COPY LJET023, LJET127, LJET219 $
-COPY LJET019 $
```

NOTE: Examples of UPDATE CS applications utilizing -COPY directives in addition to other member-level directives are presented in section 3.7.4.5.

Restrictions:

Label fields are not valid on member-level directives.

The -COPY directive is valid in update revise mode processing only.

Each specified new member name (mn) must be unique as to all other members generated on the new unit via any member-level directive in the same update directive set.

3.7.4.3.2 -OMIT directive

Purpose: The -OMIT member-level directive is valid only during update revise mode processing initiated by an UPDATE CS on which the keyword "ALL" was specified. (See section 3.7.4.2.) The -OMIT directive is used to specify the names of one or more primary old-unit (UPDATE CS parameter OLDU=) resident members that are not to be copied to the new unit during the final revise mode processing step.

Format: -OMIT␢mn_1[,...,␢mn_n] $

mn the name of a member residing on the UPDATE CS specified primary old unit (OLDU=) that is to be omitted from the new unit (NEWU=) during update revise mode "ALL" processing

Examples:

```
-OMIT LJET024, LJET126, LJET220 $
-OMIT LJET020 $
```

Restrictions:

Label fields are not valid on member-level directives.

The -OMIT directive is valid only in a revise mode processing phase initiated by an UPDATE CS on which the keyword "ALL" was specified.

If the same OLDU resident member is specified on both a -COPY directive and -OMIT directive in the same set, the specified member will be copied to the new unit.

3.7.4.3.3 -ADDR directive

Purpose: The -ADDR update member-level directive allows the user to define and generate a member on the new unit being built under any update processing mode. The record content of the new member may be copied from any specified run-life member (format 1 below) or read from the card images that follow the -ADDR in the user's primary input stream (format 2 below).

When the record data are read from primary input stream card images, the user may optionally define the format specification and maximum number of records for the new member.

Format 1: (valid within a primary or secondary input stream)

$$\text{-ADDR OLDM=}\left\{\begin{matrix}\text{pmn}\\\text{du(mn)}\end{matrix}\right\}\text{ [,NEWM=nmn] \$}$$

OLDM= required keyword specification of the run-life member whose records are to be copied to the new unit member being generated. Also, the name of the new member if optional keyword NEWM= is omitted

OLDM=pmn is a valid specification only in update revise mode processing. The specified pmn must be the name of the member residing on the primary old unit identified by UPDATE CS keyword OLDU=.

OLDM=du(mn) is a specification of the run-life data unit name (member name) whose records are to be copied to the new member. (This is valid in any update processing mode.)

NEWM=nmn (optional) specifies the name of the new member nmn being generated on the new unit. If omitted, nmn equals the name of the copied member, regardless of its unit residence.

Format 1 examples:

The following revise mode must be in effect since the run-life residence of old member JETMEM1 is assumed to be the primary old unit specified by UPDATE CS keyword OLDU=. The new member will be named JET008 and will be an exact copy of member JETMEM1.

```
-ADDR OLDM=JETMEM1, NEWM=JET008 $
```

The following specification is valid in any update processing mode. The record content of member MEM8, resident on run-life unit UNIT7, will be copied to the new unit member being generated. The new member will also be named MEM8.

```
-ADDR OLDM=UNIT7(MEM8) $
```

Format 1 restrictions:

The name of the new member, whether specified or assumed, must be unique on the new unit.

Label fields are illegal on member-level directives.

Format 2: (valid within a user's primary input stream only)

```
-ADDR OLDM=*,NEWM=mn[,FORMAT=format][,MNR=n] $
```

OLDM=* this specification indicates that the record data for new member mn is to be read from the card images immediately following the -ADDR in the user's update directive set; the record data set is terminated by the next member-level directive read or by the END* card that terminates the update directive set in the user's primary input stream. The record card-image data must correspond to the FORMAT specification supplied or assumed for the new member, as documented below.

NEWM=mn required specification of the name (mn) of the new member being generated. The name cannot be a duplicate of the names of the other members generated on the new unit.

FORMAT= (optional) keyword specification of the format type of new member mn. Valid format specifications are as documented in table 4 in section 3.7. If omitted, FORMAT=2HCI.

FORMAT=0 indicates that the records are of undefined format and variable length (unformatted). Each data field on the card images will be interpreted as an ANOPP data type. (See table 1 in section 3.1.2.) The internally converted data will be written to the unformatted records. The occurrence of a $ on a card image terminates and is not part of the data record being built; the data for the next record are assumed to begin on the following card image.

FORMAT=2HCI (default) indicates that a card-image member is to be generated. Each image following the -ADDR, up to but not including the next member-level directive or END* card, will be a record on the new member.

FORMAT=nHc$_1$,...,c$_{n-1}$ $ indicates a fixed or variable-length format specification for the new member. As documented in table 4 in section 3.7, the FORMATTED member will contain records whose elements are of the data types shown in the specification. The input data card images must contain value expressions that correspond in type and sequence with the specified format. The occurrence of a $ terminates and is not part of the data record being generated. The next input record's data are assumed to begin on the next input card image.

MNR=n (optional) integer specification of the maximum number of records that new member mn may contain. If omitted, the system default of MNR=10000 is used for the new member.

Format 2 examples:

New unit member JETM8 will be generated. Its format will be default-type card image (CI). It will contain three records, each consisting of one of the subsequent card images. Member JETM8, which may contain a maximum of 10 records, is given as follows:

```
-ADDR OLDM=* NEWM=JETM8 MNR=10 $
RECORD 1 WILL BE THIS CARD-IMAGE
RECORD 2  "    "    "    "    "
RECORD 3  "    "    "    "    "
(Next update member-level directive or the END* input terminator
card)
```

Member M4 is created as a new unit unformatted member containing three records and having a default maximum number of record value of 10000. The individual records generated will each possess the record length (in internal computer storage words) required to store the converted data values supplied for it. Those values will be:

For record 1 - one type RS (real single) value
For record 2 - two type RD (real double precision) values, one type L (logical), one type A3 value (Hollerith string of three characters)
For record 3 - three type I (integer) values

Member M4 is given as follows:

```
-ADDR OLDM=*, NEWM=M4, FORMAT=0 $
.693 $ COMMENTS ANYONE?
.70D+01   .898D+20   .TRUE.   3HABC $
10   20   30 $
(Next member-level directive or the END* input terminator card)
```

The following member M5 will be a formatted fixed-length member containing two records and a value of default of 10000 for a maximum number of records:

```
-ADDR OLDM=*, NEWM=M5, FORMAT=17HI,2RS,A9,RD,CS,L$ $
1   1.5   2E+01   9HJET-STATS   3D-01   (1.5,2.0)   .TRUE.   $
11   977.8   -.696E-110   9HPROP-STAT   1075.5D+01   (.696E+29
.32E+31)   .FALSE.   $
(Next member-level directive or the END* input terminator card)
```

NOTE: Examples of UPDATE CS applications utilizing -ADDR directives in addition to other member-level directives are presented subsequently in section 3.7.4.5.

The following member COORD contains the X, Y, Z needed for the geometry module:

```
-ADDR OLDM=*, NEWM=COORD, FORMAT=4H3RS$ $
0.   0.   4.   $
END*
```

Format 2 restrictions:

The ADDR keyword specification OLDM=* is valid only within a user's primary input stream.

The name of the new member (NEWM=) must be unique on the new unit being built.

The FORMAT= specifications must correspond with the rules documented in table 4 in section 3.7. Individual data-record values specified on input card images must correspond in type and sequence with the FORMAT= specification and must correspond to the value specification rules presented in table 1 in section 3.1.2.

```
*****************************************************************************
*                                                                           *
*                                  NOTE                                     *
*                                                                           *
*****************************************************************************
*                                                                           *
*       The remaining directives in this section are intended for large-    *
*   application users who will establish and maintain large data base      *
*   applications.                                                           *
*                                                                           *
*****************************************************************************
```

3.7.4.3.4 -CHANGE directive

Purpose: The -CHANGE update member-level directive allows for the generation of a new unit member, whose records will be selectively copied from a user-specified old run-life member, with optional insertion of additional new records read from within the user's card-image update directive set. The sequence and source of new member records to be generated are in accordance with a set of card-image record-level directives subordinate to the -CHANGE directive.

The new member will possess the same format as does the specified old member; the name and maximum-number-of-records value of the new member may optionally be changed from that of the old member.

A set of record-level directives must immediately follow the -CHANGE directive it supports in the update directive set. The occurrence of a subsequent member-level directive, or the termination of the user's update directive set, ends the -CHANGE subordinate record-level set.

Format: -CHANGE␢OLDM= $\left\{\begin{matrix}\text{pmn} \\ \text{du(mn)}\end{matrix}\right\}$ [␢NEWM=nmn] [␢MNR=n] $

OLDM= specifies the run-life member whose resident records will be selectively copied onto the new member in accordance with the record-level directives set

OLDM=pmn is valid only in update revise-mode processing. The specified old member (pmn) must be resident on the primary old unit identified via the UPDATE CS keyword OLDU= specification.

OLDM=du(mn) is valid in any update processing mode and specifies the run-life data unit name (member name) of the old member.

NEWM=nmn (optional) specifies the name of the new unit member being generated. If omitted, the name of the specified old member (OLDM=) will also be assigned to the new member.

MNR=n (optional) specifies the maximum number of records for integer n that the new member may contain. If omitted, the system will calculate the new member's MNR value based on the old member's MNR value and current number of records.

<u>Examples:</u>

Since the run-life data unit residence of old member M1 was not specified, this must be a revise-mode update, initiated by an UPDATE CS on which a primary old unit (OLDU=) was specified. In accordance with the record-level directive set, the following new unit member JETM1 will contain records selectively read from primary old-unit member M1 and, optionally, records read from within the record-level directive set:

```
-CHANGE OLDM=M1,NEWM=JETM1 $
     :
(Record-level directives)
     :
(Member-level directive or the end of the user's update directive
  set)
```

The following new unit member MEM2 will be created with an assigned value of 2000 for maximum number of records. In accordance with the record-level directive set, records will be selectively copied to the new member from run-life unit FANU member MEM2.

```
-CHANGE OLDM=FANU(MEM2) MNR=2000 $
     :
(Record-level directives)
     :
(Member-level directive)
```

NOTE: Examples of -CHANGE member-level directive applications using specific subordinate record-level directives are presented in the following sections. Section 3.7.4.5 presents examples using combinations of all update member and record-level directives.

3.7.4.4 Record-Level Directives

The general format for -CHANGE subordinate record-level directives is

directivename[$_\not b$field... $_\not b$field] $

The valid directivenames are

-INSERT -DELETE -QUIT

The fields are as specified in the following subsections that document the individual directives. Field-separation and value-specification rules are as documented for ANOPP control statements in section 3.1.2. Labels are illegal on record-level directives. The record-level directives are processed under control of the preceding -CHANGE member-level directive. The -CHANGE directive, documented in the previous section, performs the following:

Identifies the new unit member whose record content will be a changed copy of the current content of a run-life old member

Identifies the old member (OLDM=)

Remains in control until each of its subordinate record-level directives have been processed

Each directive generates the copying of single or groups of records either from the old member or from within the directive set. Throughout a record-level directive set, the integer operands i and j are used as pointers to relative record positions in the old member. Directives within a set must be processed (and positioned) sequentially with respect to any i and j operand specifications they possess. This is necessary because the new member is generated in one pass.

The system monitors an old member reference pointer P throughout the generation of a new member via a -CHANGE and its record-level directive set. Here, P is initiated at zero. Upon completion of each directive's processing, P assumes one of the following values:

(1) The j value (if specified on the directive)
(2) The i value (if specified, and j value not specified)
(3) Unchanged (if i value not specified)

The next record-level directive must have an i value (if present) greater then the new value of P.

The i operand on a directive specifies the relative record position of the first record in a sequence of one or more old-member records. Depending on the directive, this single or sequence of old-member records will be treated as follows:

> Copied to the new member, optionally to be followed by inserted records read from the card images immediately following the directive or from elsewhere in the old member
>
> Omitted from copying to the new member

The j operand, optionally used on the -DELETE directive, is used to specify the last old-member record in a series to be omitted from copying to the new member. The i value (if present) on any directive must be greater than the i (and j, if present) value specified on a preceding directive in the set. For example, a directive referencing old-member records 5 through 8 (i through j) must be processed before one referencing 9 through 10 (i_2 through j_2), and never vice versa.

The -QUIT directive, if encountered, terminates the new-member generation process and the -CHANGE directive's subordinate set. It may specify that some, all, or none of the remaining old-member records be copied to the new member as a final generation step. If -QUIT is not used, the set terminates with the occurrence of another update member-level directive or with the termination of the update directive set. Deletion (or omission) of old-member records from the new-member copy is performed via the -DELETE directive. Insertion of new records read from directive set card images is performed via the -INSERT directive. Resequencing of old-member records onto the new-member records is performed via the -INSERT directive.

3.7.4.4.1 -INSERT record-level directive

Purpose: The -INSERT record-level directive allows the user to insert onto the new member a group of one or more sequentially positioned records, either currently residing on the old member or immediately following the directive card image in the record-level directive set.

Optionally, the user may specify that a group of one or more old-member records will be copied onto the new member to precede the inserted records.

Format: **-INSERT**$_{\not b}${i$_{\not b}$} [**FROM**=⟨**OLDM**⟩$_{\not b}$] [(m[,n])] $

i (optional) the integer relative position of the last old-member record to be copied to the new member prior to the specified records to be inserted. Integer i must be greater than old-member record pointer P. Records P + 1 through and including i will be copied to the new member.

If omitted, P remains unchanged, allowing records to be inserted at any position on the new member, including at the beginning when P = 0. The concepts of old-member record-pointer P are presented in section 3.7.4.4.

FROM= (optional) specifies the source of the records to be inserted onto the new member. If omitted, FROM=* is assumed.

FROM=* (default) indicates that the records to be inserted on the new member immediately follow in the directive-set card images. The new record card images are terminated by the next record-level directive, member-level directive, or termination of the directive set. FROM=* is valid in any CS stream. The new record data must correspond to the format specification of the -CHANGE specified old member. (See table 4 in section 3.7.)

If the format of OLDM is unformatted, fixed, or variable, then a $ must terminate the data of each input record and the next record must begin on a separate card image. Applicable examples are presented below and in the following subsections.

FROM=OLDM indicates that the records to be inserted on the new member reside on the -CHANGE specified OLDM and are specified by the next operand field.

(m[,n]) (optional) specifies which old-member record (m) or sequence of old-member records (m,n) are to be inserted on the new member. This field specification has no effect on old-member record pointer P. The integer value of m or n may be greater, less than, or equal to an i-parameter specification; n, when specified, must be greater than m. If FROM=*, this field is ignored. If omitted, and FROM=OLDM, all OLDM records will be inserted on the new member.

Examples:

Example 1: Assume that N1 is a card-image formatted member on the UPDATE CS specified primary old data unit. Assume that the user desires to insert two new cards following the sixth card-image record of the old member. The following would copy the first six records onto the new member and then insert the two new ones:

```
-CHANGE OLDM=N1 $
-INSERT 6 $
 (Card image)
 (Card image)
 (Next member-level directive)
```

Example 2: The following directive would copy the first six old-member records to the new member, to be followed by an insertion of a second copy of old-member records 1 through 3. Following this process, P would equal 6.

```
-INSERT 6, FROM=OLDM (1,3) $
```

Example 3: The following is a second example of the -INSERT capability to reorganize old-member records onto the new member. The old memoer contains 30 records, consisting of 3 functionally related groups of 10 records each. Assume that the user wishes to retain all the records but intends to position the second related group (old-member records 11 through 20) at the beginning of the new member. The following would perform the desired resequencing:

-INSERT FROM=OLDM (11,20) $	Copy records 11 to 20.
-INSERT 10 FROM OLDM (21,30) $	Copy records 1 to 10.
	Copy records 21 to 30.

Example 4: For the following example, assume that the OLDU data unit is UN1 and the user wishes to create a new unit member based on run-life unit UN3 member MEM7. MEM7 has a fixed-length format specification of FORMAT=6H3I,3L$. It contains 10 records, all to be retained on the new member. The user intends to insert from within his directive set one new record behind old-member records 5 and 10. The new member will retain the name MEM7.

```
-CHANGE OLDM=UN3(MEM7) $
-INSERT 5 FROM=* $
 2765 8329 10754 .TRUE. .TRUE. .FALSE. $
-INSERT 10 FROM=* $
 6439 10376 7420 .TRUE. .FALSE. .FALSE. $
(Next member-level directive)
```

NOTE: Additional -INSERT directive examples are included in the following subsections presenting the -DELETE and -OMIT record-level directives.

Restrictions:

New data records must correspond to the member format of OLDM.

A value specification for operand i must be greater than that specified for operands i or j on any preceding record-level directive in the set.

3.7.4.4.2 -DELETE record-level directive

Purpose: The -DELETE member-level directive allows the user to bypass old-member records that are not to be copied to the new member being generated. Optionally, the user may cause a preceding group of old-member records to be copied to the new member.

Format: -DELETE␣i[,j] $

i[,j] the integer operands i and j specify a range of old-member positioned records that are not to be copied to the new member. The i must be greater than the current value of old-member pointer P; and j, if specified, must be greater than i.

If P + 1 is less than i, old-member records P + 1 through i - 1 are first copied to the new member.

Pointer P is set to the value of j, or to the value of i if j is not specified.

Example:

Assume that old-member JET1 resides on the primary old data unit and contains 20 records. Old-member records 9, 10, 15, 16, and 17 are to be deleted. New records are to be inserted following current old-member records 5 and 12 on the new member. In addition, current old-member records 2 and 3 are to be moved to the end of the records on the new member.

-CHANGE OLDM=JET1 $	At initiation, old-member record pointer is initiated. (P=0)
-DELETE 2,3 $	Old-member record 1 is copied. (P=3)
-INSERT 5 FROM=* $ (New record N1 card images) (New record N2 card images)	Old records 4 and 5 are copied. New records N1 and N2 are inserted. (P=5)
-DELETE 9,10 $	Old records 6 through 8 are copied. (P=10)
-INSERT 12 FROM=* $ (New record N3 card images)	Old records 11 and 12 are copied. New record N3 is inserted. (P=12)
-DELETE 15,17 $	Old records 13 and 14 are copied. (P=17)
-INSERT 20 FROM=OLDM (2,3) $	Old records 18, 19, and 20 are copied. Old records 2 and 3 are then copied. (P=20)

(Next member-level directive)

The new member would now contain:

```
OLDM Rec  1
 "    "   4
 "    "   5
 New Rec N1
 "    "  N2
OLDM Rec  6
 "    "   7
 "    "   8
 "    "  11
 "    "  12
New Rec  N3
OLDM Rec 13
 "    "  14
 "    "  18
 "    "  19
 "    "  20
 "    "   2
 "    "   3
```

NOTE: Additional -CHANGE directive applications using combinations of subordinate record-level directives are presented in section 3.7.4.5.

3.7.4.4.3 -QUIT record-level directive

Purpose: The -QUIT record-level directive allows the user to terminate the generation of the -CHANGE identified new member, optionally specifying the final old-member record to be copied to the new member.

Format: -QUIT [i] $

i (optional) specifies the integer relative position of the old-member record that will be the last record copied to the new member. If specified, i must be greater than old-member record pointer P. Records P + 1 through and including record i will be copied to the new member. If omitted, processing of the new-member generation is terminated immediately.

Examples:

Assume for both subsequent examples that the old member contains 20 records and that the preceding record-level directive processed was the following:

```
-INSERT 15 FROM=* $
 (New record N1 card images)
```

Example 1: The following would result in termination of the -CHANGE-initiated new-member generation. The last new-member record will be N1.

```
-QUIT $
```

Example 2: The following would result in old-member records 16 and 17 being the last records to be generated on the new member:

```
-QUIT 17 $
```

Restrictions:

The -QUIT directive, if present, must be the last record-level directive under the control of the -CHANGE directive.

A value specification of operand i must be greater than that specified for operand i or j on any preceding record-level directive in the set.

3.7.4.5 Examples

The following update processing phase examples incorporate all of the member-level and record-level directives. It is assumed that each data source unit or new unit identified was established in the run-life data base via a preceding CREATE CS, ATTACH CS, or LOAD CS.

Example 1: A revised mode update is to be performed by using run-life unit OLDJET as the primary old unit. Unit OLDJET contains 20 members named MEM1 through MEM20.

Unit NEWJET is the run-life unit to be built (or rebuilt). NEWJET is to receive the following members:

MEM1 through MEM6	copied unchanged from unit OLDJET
MEM7	rebuilt, using OLDJET Unit MEM7 as its primary source member. MEM7 contains eight records; records 6 and 7 are to be replaced by user input records.
MEM8	to be generated by using new input records
MEM9	to be a renamed copy of run-life unit UNITFAN resident member UFMEM1
MEM10 through MEM15	copied unchanged from unit OLDJET
MEM16	to be a rebuilt version of run-life unit UNITFAN resident member UFMEM8. Member UFMEM8 consists of 10 records; records 4 and 9 are not to be copied to new member MEM16.
MEM17 through MEM20	copied unchanged from unit OLDJET

Thus,

```
UPDATE OLDU=OLDJET NEWU=NEWJET SOURCE=* LIST=ESCA $
    -COPY MEM1, MEM2, MEM3, MEM4, MEM5, MEM6 $
    -CHANGE OLDM=MEM7 $
        -INSERT 5 FROM=* $
         .TRUE. .TRUE. 17.65D27 7278 (7.53 1E320) $
         .FALSE. .FALSE. -1D+322 6920 (375.8E-12,270E+12) $
        -DELETE 6,7 $
        -QUIT 8 $
    -ADDR OLDM=* NEWM=MEM8 FORMAT=8H2I,L*RS$ MNR=10 $
     757 9329 .TRUE. 17.97 -3.18E03 $
     14 723 .TRUE. 1E216 275E+10 18.327 $
     1278 87 .FALSE. $
    -ADDR OLDM=UNITFAN(UFMEM1), NEWM=MEM9 $
    -COPY MEM10, MEM11, MEM12, MEM13, MEM14, MEM15 $
    -CHANGE OLDM=UNITFAN(UFMEM8) NEWM=MEM16 $
       -DELETE 4 $
       -DELETE 9 $
       -QUIT 10 $
    -COPY MEM17, MEM18, MEM19, MEM20 $
END* $
```

<u>Example 2</u>: A revised mode update is to be performed by using run-life unit STAT1 as the primary old unit. STAT1 contains 15 members named MEM1 through MEM15.

Unit NEWSTAT is the run-life unit to be built and is to receive the following members:

MEM1 through MEM4	copied unchanged from unit STAT1
MEM5	rebuilt by using system unit DATA member MEMB as the primary source member. MEMB is a card-image formatted member containing seven cards; cards 2 and 7 are to be replaced by user input records.
MEM6	to be generated by using new input records
MEM7 and MEM8	renamed copies of unit STAT1 members MEM14 and MEM15
MEM9 through MEM13	copied unchanged from unit STAT1

Thus,

```
UPDATE OLDU=STAT1 NEWU=NEWSTAT ALL SOURCE=* $
     -CHANGE OLDM=DATA(MEMB) NEWM=MEM5 $
          -INSERT 1 FROM=* $
          THIS CARD IMAGE WILL BE NEW MEM5 RECORD 2
          -DELETE 2 $
          -INSERT 6 FROM=* $
          THIS CARD IMAGE WILL BE NEW MEM5 RECORD 7
     -ADDR OLDM=* NEWM=MEM6 FORMAT=16HA7,L,I,RS,RD,CS$ $
          6HHEIGHT .TRUE. 89 + 7.93 17.16D27 (9.75,1E279) $
          5HSPEED .FALSE. 94757 -1.33E05 1.36D-03 (-1.7E02 0.92) $
     -OMIT MEM14, MEM15 $
     -ADDR OLDM=MEM14 NEWM=MEM7 $
     -ADDR OLDM=MEM15 NEWM=MEM8 $
END* $
```

NOTE: The ALL keyword specification on the UPDATE CS above results in all primary old-unit STAT1 members that were not specified on a -OMIT directive, and that do not possess a name that is the same as a new unit member generated via a -CHANGE or -ADDR directive, being copied unchanged to the new unit NEWSTAT during the final update processing phase step.

3.8 CONDITIONAL PROCESSING FLOW

3.8.1 Overview

The normal processing flow occurring in an ANOPP run is the sequential execution of each control statement in the user's input streams. Following execution of the last CS in a called secondary stream, the system executes the control statement that follows the CALL CS in the calling stream. Execution of the ENDCS control statement, marking the end of the user's primary input stream, terminates the ANOPP run.

The user, however, has the capability to deviate from normal sequential CS processing through error handling and branch control statements.

Controlling subsequent CS processing following the occurrence of a system "nonfatal" error.- Prior to execution of each control statement, the system sets an internal nonfatal error flag to "off." If during the execution of the CS or FM an abnormal but nonfatal condition occurs, the flag is set to "on." Documentation of error diagnostics of specific functional module (section 4) and ANOPP system (appendix C) iden~~tifies~~ conditions that result in nonfatal errors.

Following the execution of a functional module or a control statement, the system performs this sequence of steps:

- Test the value of the internal nonfatal error flag.
  - If "off," continue processing with next CS in stream.
  - If "on" (error occurred), test value of system parameter JCON:
    - If JCON=.TRUE. (set by user via SETSYS CS), continue processing with next CS in stream.
    - If JCON=.FALSE. (default value), search forward sequentially seeking a PROCEED CS:
      - If a PROCEED CS is found, continue processing with the execution of the CS following it.
      - If the ENDCS ending the user's primary input stream is found first, terminate the job.

If, at the time of the error, control was within a secondary input stream and JCON=.FALSE., the search for a PROCEED CS will continue through the rest of the secondary stream. The search continues with the remainder of the calling stream beginning with the CS following the CALL CS until either a PROCEED CS or the run terminating ENDCS is found.

Determining subsequent CS processing flow based on the current value of a user parameter.- The IF control statement allows the user to specify a logical condition of several types. The user also specifies a name that must be the label field value of another control statement in the same input stream. If the specified condition test is .TRUE., processing will continue with the execution of the CS having the specified label name; if the condition test is .FALSE., processing continues normally with the next sequential CS in the stream.

The GOTO CS performs the single function of transferring processing control unconditionally to the control statement having a specific label name. The CONTINUE CS is a control statement that may contain a label-name field and is used as the object of IF CS or GOTO CS processing, but it causes no other action to occur.

3.8.2 GOTO CS

Purpose: The GOTO control statement allows for an unconditional deviation from the sequential processing within an input stream. Processing will continue with the execution of the control statement having the label-field value specified.

Format: [label$_{\not b}$]GOTO$_{\not b}$labname $

label the optional label name of this control statement

labname the required label name of another control statement within the same input stream where processing will continue

Examples:

```
ENDST1 GOTO STEP3 $
GOTO STEP2 $
```

Restriction: Labname must be the label name of a control statement within the same input stream as the GOTO CS.

3.8.3 IF CS

Purpose: The IF control statement permits an alteration in the flow of processing if a specified condition exists. The conditional statement is specified within parentheses. In the first form, the value of a user parameter (or parameter array) is logically compared with the value of either another user parameter (array) or user-specified constant(s). In the second form, the existence or nonexistence of a unit member is determined. If the condition within the parentheses is true, then processing continues with the control statement having the specified label; otherwise processing continues with the next control statement.

Formats:

$$\text{label}_{\not b}\text{IF}\left(\text{paramname}_1\ \ {}_{\not b}\begin{matrix}\text{logical}\\ \text{operator}\end{matrix}\begin{Bmatrix}{}_{\not b}\text{paramname}_2 \\ {}_{\not b}\text{value}\ \ \ \ \ \ {}_{\not b}\end{Bmatrix}\right)\text{GOTO}_{\not b}\text{labnam}\ \$$$

$$\text{label}_{\not b}\text{IF}\left(\text{unit(mem)}\begin{Bmatrix}\text{AS}\\ \text{NAS}\end{Bmatrix}\right)\text{GOTO}_{\not b}\text{labnam}\ \$$$

label label name

paramname$_1$ name of a user parameter whose value(s) is to be compared with the value(s) following the logical operator

logical operator a logical operator used in comparison of the two values. Any logical operator is valid for comparing values which are type integer, real single precision, or real double precision. (See table 1 in section 3.1.2.) The operators .EQ. and .NE. are the only valid operators for logical, complex single, and character string data types. For the .NE. operator, the results of element comparisons are combined with an OR condition. For all other operators, the results of element comparisons are combined with an AND condition. For the .NE. operator, only one element of the first array need be unequal to the corresponding element of the second array in order for the IF CS to yield a true condition. However, for all other operators, every element of the first array compared to corresponding elements of the second array must yield a true condition in order for the IF CS to yield a true condition.

value array of one or more elements of type numerical constant (I, RS, RD, CS), logical constant, or string constant. All elements must be of the same type.

$paramname_2$ name of an existing user parameter array to be used in comparison

unit name of data unit on which data member, mem, resides

mem name of data member

AS flag indicating condition is true if unit (mem) exists. If unit (mem) exists as a data unit member and flag AS is indicated, a true condition results. If unit (mem) does not exist and AS is indicated, a false condition results.

NAS flag indicating condition is true if unit (mem) does not exist. If unit (mem) does not exist as a data unit member and flag NAS is indicated, a true condition results. If unit (mem) exists and NAS is specified, a false condition results.

labnam the label of the control statement at which processing should continue if the comparison of the two values results in a true condition.

Examples:

```
LABEL1   IF (A .GE. B) GOTO LABEL2 $
         IF (D .EQ. .TRUE.) GOTO LABEL1 $
         IF (F .EQ. 6HFVALUE) GOTO LABEL2 $
         IF (G .GT. 10 20 30) GOTO LABEL1 $
         IF (H .EQ. .FALSE. .TRUE.) GOTO LABEL1 $
         IF (PROC (SOURCE) AS) GOTO LABEL3 $
         IF (PROC (SOURCE) NAS) GOTO LABEL2 $
         .
         .
LABEL2   CONTINUE $
         .
         .
LABEL3   CONTINUE $
```

Restrictions:

Labname must be in the label field of a control statement within the same input stream as the IF CS.

The data type of the second value array must agree with the value type of $paramname_1$.

When comparing string constants of unequal length, trailing blanks are ignored.

The number of elements of the second-value array should agree with the number of elements of $paramname_1$; if not, the condition is assumed false.

3.8.4 CONTINUE CS

Purpose: The CONTINUE control statement provides a no-action step within the input stream. It is used mainly with the IF CS and/or GOTO CS to allow for an alteration in the flow of control-stream processing. The processing continues with the execution of the CS following the CONTINUE CS.

Format: **[label₉]CONTINUE $**

label label field

Example:

```
LABEL1 CONTINUE $
```

Restriction: None.

3.8.5 PROCEED CS

Purpose: The PROCEED control statement marks the location within the input stream where ANOPP execution will continue if a nonfatal error condition occurs during the processing of a preceding CS or functional module and if the current value of system parameter JCON is .FALSE.. The PROCEED CS is a no-operation statement when encountered during normal sequential CS processing.

Format:

[label₉]PROCEED $

label label field

Examples

```
ERROR1 PROCEED $
PROCEED $
```

Restrictions:

During the forward search in the input stream of the system for a PROCEED CS, any CALL statements encountered are not processed. (The procedure-member resident input stream is not searched or expanded.)

If the nonfatal error condition initiating the search for the PROCEED CS occurred during processing within a CALL'ed secondary input stream, and no PROCEED CS is found to exist in the remainder of that stream, the search will continue in the input stream that contained the CALL, beginning with the control statement following the CALL CS.

If the ENDCS control statement ending the user's primary input stream is found before any PROCEED CS, the run will be terminated.

3.8.6 Examples

The following examples illustrate conditional processing flow within input streams.

Example 1: The following input stream illustrates the processing flow that will occur if a system nonfatal error occurs during the execution of a functional module:

```
          STARTCS $
              :
          SETSYS JCON=.TRUE. $
          EXECUTE FM1 $
          SETSYS JCON=.FALSE. $
          EXECUTE FM2 $
          ATTACH FM3UNIT/EFN3/ $
          EXECUTE FM3 $
          DETACH FM3UNIT $
          GOTO LASTSTEP $
          PROCEED $
          ATTACH FM4UNIT/EFN4/ $
          EXECUTE FM4 $
          DETACH FM4UNIT $
LASTSTEP  EXECUTE FM5 $
              :
          ENDCS $
```

Since system parameter JCON equals true when functional module FM1 is executed, processing will continue with the SETSYS CS that follows it, regardless if a nonfatal error occurs during FM1 execution. However, JCON=.FALSE. when functional module FM2 is executed. If no nonfatal error occurs in FM2, processing continues with the following ATTACH CS. If a nonfatal error does occur in FM2, processing will continue with the ATTACH CS following the PROCEED CS.

Example 2: The following input stream illustrates conditional processing flow that occurs following the test of a logical condition based on the value of a user parameter:

```
          STARTCS $
              :
          PARAM EXCOUNT=0 $
REPEAT    PARAM COUNT=EXCOUNT + 1 $
          EXECUTE JET $
          PARAM EXCOUNT=COUNT $
          IF (EXCOUNT .NE. 5) GOTO REPEAT $
          IF (JETRSLT .EQ. .TRUE.) GOTO JETT $
          EXECUTE JET9 $
          GOTO ENDJET $
JETT      EXECUTE JET7 $
ENDJET    CONTINUE $
```

Functional module JET is to be executed five times. The value of user parameter COUNT is used both to control the specific function to be performed by FM JET and to count the execution repetitions. Following the fifth execution of FM JET, the value of logical user parameter JETRSLT is tested. The value of JETRSLT is set internally in FM JET. If JETRSLT equals .TRUE., FM JET7 will be executed next; otherwise, FM JET9 will be executed.

3.9 PROCEDURE-MEMBER INPUT STREAM

3.9.1 Overview

Procedure members are card-image formatted members whose records are valid control-statement card images. This set of CS card images is referred to as a secondary input stream. A secondary input stream may be called into execution via a CALL CS. Section 5 documents secondary input streams maintained as part of the ANOPP system and available to the user for execution of specific noise prediction or utility-type functions. A secondary input stream residing on a procedure member may consist of any valid control statement card images except for the ANOPP CS, DATA CS, ENDCS CS, RSTRT CS, and STARTCS CS. Use of the END* input terminator card is also illegal within secondary input streams.

More than one CALL CS in the same run may specify the same procedure member. The input stream resident on a procedure member may be altered during a run via the member regeneration capabilities of the UPDATE CS or DATA CS. In addition, the field-replacement capability of the CALL CS allows the user to modify the procedure-member input stream prior to processing control being passed to the secondary stream. The first execution of a specific CALL CS during a run generates an executable version of the called secondary input stream, based on the current CS card-image content of the specified procedure member and on any field replacements also specified on the CALL CS. Any subsequent reexecution of the same CALL CS (as a result of internal input-stream looping via IF CS or GOTO CS processing) will result in the same version of the input stream being executed. The initial execution of a CALL CS also results in a system edit phase being performed on the secondary stream. Each CS is examined for correct syntax. A CS edit phase error results in run termination.

The user is provided a CALL CS capability to perform text modification on the field contents of the CS card images residing on the specified procedure member. The modified version of the secondary input stream is then examined during the system CS edit phase. Provided no edit errors are encountered, the system builds the form of the secondary input stream that will be executed each time this specific CALL CS is processed during the run.

The field-replacement feature of the CALL CS requires the user to specify one or more old field values (of any valid CS data type including the type N name of a control statement) and, for each, a new field value, also of any type.

The system will examine the individual field content of each CS card image on the procedure member. Wherever a specified old value is found it

C-2

will be replaced by the matching specified new value. The rebuilt control statements will reside on a system-maintained member; the original content of the CALL CS specified procedure member is never altered via a CALL CS.

Typical use of the field-replacement feature would be the substitution of a library file name, data unit name, functional module name, and/or user parameter value within a secondary input stream that performs a commonly required sequence of ANOPP processing steps.

3.9.2 CALL CS

Purpose: The CALL control statement allows the user to transfer run processing control to a secondary input stream. Following execution of the last CS in the called stream, run control continues with the CS following the CALL control statement.

The initial processing of a specific CALL CS establishes the secondary input stream to be executed by this and all subsequent executions of the same CALL CS in the run. The content of the executable input stream is based on the CS card images residing on a specified run-life procedure member and any field replacements specified on the CALL CS.

A field-replacement specification consists of an old value and a corresponding new value. Any field of any CS card image residing on the procedure member that matches a specified old value will be replaced by the corresponding new value on the executable version of the secondary input stream.

A system CS edit phase is performed prior to the initial execution of a CALL'ed secondary input stream. Any CS format or syntax error will result in run termination.

Format:

[label$_{\not b}$]CALL$_{\not b}$dun(pmn)[$_{\not b}$oldvalue$_1$=newvalue$_1$...$_{\not b}$oldvalue$_n$=newvalue$_n$] $

label label name

dun name of the run-life unit on which pmn resides

pmn name of the procedure member containing control statement card images to be used in the generation of the secondary input stream to be executed

oldvalue (optional) the exact field value which, if found to occur on a pmn resident CS card image, is to be replaced with the corresponding new value. The old value may be of any type valid on any control statement, including the name of a control statement which is a type N (Name) data type. (See table 1 in section 3.1.2.)

newvalue the CS image field to replace each occurrence of old value. The new value may be of any type valid on a control statement and does not have to be the same as the type of the old value.

Examples:

Example 1: The following CS will call into execution the secondary input stream consisting of the card images residing on data unit UN1, procedure member MEM3, with no field replacement involved:

```
LABEL1 CALL UN1(MEM3) $
```

Example 2: Assume that procedure member NOISEUN, residing on run-life unit TBLTEST, contains the following card images:

```
                LOAD/SEFN/ $
                PARAM INDX=0 $
REDO            PARAM TBLINDX=INDX+1 $
                EXECUTE FMNAME $
                IF (TBLINDX .EQ. REPEATS) GOTO FINI $
                PARAM INDX=TBLINDX $
                GOTO REDO $
FINI            DROP /SEFN/ $
```

The user wishes to execute functional module JET1 10 times. The JET1's data base requirements are contained on library file JETLIB. The following CALL CS would build and execute the desired secondary input stream:

```
CALL NOISEUN(TBLTEST) /SEFN/=/JETLIB/ FMNAME=JET1  REPEATS=10 $
```

Following the system's performance of the CALL CS specified field replacements, the executable version of the secondary input stream would be as follows (the new field values are underlined):

```
                LOAD /JETLIB/ $
                PARAM INDX=0 $
REDO            PARAM TBLINDX=INDX+1 $
                EXECUTE JET1 $
                IF (TBLINDX .EQ. 10) GOTO FINI $
                PARAM INDX=TBLINDX $
                GOTO REDO $
FINI            DROP /JETLIB/ $
```

Restrictions:

Specified old values must be an exact specification of an entire control statement field.

Only the initial processing of a specific CALL CS in a run results in the generation and editing of the secondary input stream to be executed. Subsequent reprocessing of the same CALL CS, regardless of any modifications made to the specified procedure member's content, will execute the same secondary stream as did the initial processing of the CALL CS.

The procedure-member resident secondary input stream may not include any ANOPP CS, DATA CS, ENDCS CS, RSTRT CS, STARTCS CS, or END* input terminator card images.

3.10 CHECKPOINT/RESTART

3.10.1 Overview

The ANOPP system provides the capability to establish checkpoints within an input stream(s) of a run and to restart that run in a subsequent job without reprocessing the input stream statements that preceded a specific checkpoint in the initial run. The intent of checkpoint/restart is to save the run environment in the event of either a planned or unplanned interruption in an ANOPP execution. This capability may be used to protect against system crashes or to halt a run's execution for examination of intermediate data without losing execution time or data base characteristics. Depending on the results of that examination, the user may specify that the operating environment that existed at any specific checkpoint be reestablished and execution continued.

The checkpoint run.- A checkpoint run is any ANOPP run during which a CKPNT CS is executed. During the execution of CKPNT CS, the system writes to a checkpoint file the necessary internal system and run-life data base information required to recreate this run's current operating environment for use in a subsequent restart run.

The checkpoint/restart file.- During each checkpoint run the user must ensure that an external sequential checkpoint file is assigned to the job. This checkpoint file must be permanently retained in the external system following job termination if it is to be used as the restart file in an ANOPP restart run application. The checkpoint/restart file may reside on tape or rotating mass storage.

The checkpoint/restart file structure is divided into primary divisions called cycles. Each cycle is referenced by the integer number of its sequential position on the file. Cycle 1 would be the first cycle written to the file resulting from the first CKPNT CS executed in the checkpoint run. Any subsequent CKPNT control statement execution in the run would generate another cycle on the checkpoint/restart file. Whenever a checkpoint-file cycle is generated during a run, a report is printed identifying the cycle number, the current run-life data units, and other checkpoint status information that will assist the user in selecting his future restart options. In a subsequent restart run the user specifies at what position (checkpoint) he wishes to restart his run within the initial processing stream of the checkpoint run.

Each cycle of the checkpoint file contains the necessary information to preserve the current operating environment that existed at the time its generating CKPNT CS was executed. The cycle information contains the following:

All required internal system tables, including those that contain:
- User parameter names and values
- Run-control (initialization and system) parameter values
- Run-life data base directories

A copy of each nonarchived run-life unit and its members, including system unit DATA

A list of the archived run-life units

The external name under which the sequential checkpoint file is assigned to a user's job must also be established within the run. The user may specify the external name of the checkpoint file via the ANOPP CS or RSTRT CS specification of initialization parameter CKPNT. If not specified in this manner, the system uses the default name of CPFILE.

The CKPNT CS.- The CKPNT CS is used to establish a checkpoint within any input stream in a user's run. The CKPNT CS is not automatically executed when encountered during the processing phases of an ANOPP run. The CKPNT CS is treated as inactive (a no-operation CS) if logical system parameter JCKPNT is set to system default value .FALSE. at the time that the CKPNT CS is processed. This allows the user to insert CKPNT control statements at potential restart locations within the input streams and control activation and deactivation via system parameter JCKPNT. A CKPNT CS may contain an optional specification of the keyword STOP. If specified on an executed CKPNT CS, the ANOPP run will be terminated following generation of the checkpoint-file cycle.

System parameter JCKPNT.- The value of JCKPNT controls all checkpoint (CKPNT CS) processing in ANOPP runs. Only when the current value of JCKPNT equals .TRUE. is a CKPNT CS executed, resulting in the generation of a new checkpoint-file cycle. If the user specifies the name of an external checkpoint file via initialization parameter CKPNT on an ANOPP CS or RSTRT CS, the initial value of JCKPNT is set to .TRUE.. The user may also activate and deactivate checkpoint processing during his runs via SETSYS CS specifications of parameter JCKPNT values.

The restart run.- Each ANOPP restart run is initiated via the special RSTRT control statement. The RSTRT CS must be the first card in the run input deck and provides run-initialization capabilities similar to those provided by the optional ANOPP CS in nonrestart runs. The RSTRT CS performs the following:

- Identifies the run as an ANOPP restart application
- Indicates which previous checkpoint run is to be restarted at what checkpoint
- Allows for the change of previous initialization or system parameter values

The optional RSTRT= keyword parameter of the RSTRT control statement is used to specify the name under which the user's external restart file is assigned to his job. If omitted, the system assumes that the user's restart file has the name RSFILE.

The RSCYCL= keyword parameter of the RSTRT CS is used to specify the integer number of the cycle on the restart file to be used to restart the previous checkpoint run. If omitted, the last cycle generated on the file will be used.

Restart-run initialization.- The initialization phase of a restart run begins when the RSTRT CS is encountered. The RSTRT CS specified restart file and cycle are accessed. The operating environment preserved by the specified cycle is reestablished. The initialization steps are as follows:

(1) System and initialization parameters are assigned their checkpointed values or are assigned new values if specified on the RSTRT CS.

(2) The checkpointed run-life data base is reestablished. Each nonarchived data unit and its members will be copied from the restart file to an external file with the same name as the one it resided on at checkpoint time. If the user has an external file assigned to the restart job that matches a file name of a checkpointed unit stored on the restart file, that external file will receive the copy; otherwise, the system will use a scratch (temporary) file of the same name. The system unit DATA is regenerated by using checkpoint cycle copies. No archived data unit file that existed in the checkpointed run-life data base will be regenerated. The user has the responsibility of ensuring that the external files on which archived units reside are assigned to his restart job. The user must ensure that any library files required by subsequent restart-run processing are externally assigned to his job.

(3) The user parameter tables and data unit directory are reestablished with checkpointed values.

(4) System parameter JRSTRT is set to .TRUE., indicating that ANOPP restart run processing is to occur.

Following initialization, the next processing step depends on whether the user has supplied a new primary input stream in his run input deck. If the RSTRT CS was the only card image in the input deck, run processing resumes with the execution of the control statement within the initial-run input stream which followed the CKPNT CS that generated the applicable checkpoint (restart) cycle.

An inserted primary input stream can be processed either prior to or instead of the resumption of checkpointed input stream processing as described above. An inserted primary input stream may not contain a CKPNT control statement.

The value of logical system parameter JRSTRT upon completion of the inserted input stream (ENDCS) processing determines whether the run will terminate or resume processing the checkpointed input stream. The JRSTRT is set to .TRUE. during restart-run initialization. If the user has changed its value to .FALSE., and the ENDCS is processed, the run will be terminated at that point. If JRSTRT is equal to .TRUE. at ENDCS process time, run processing will continue with the execution of the control statement that follows the applicable CKPNT CS within the checkpointed input streams.

3.10.2 CKPNT CS

Purpose: The CKPNT control statement establishes a potential checkpoint within a user's run. A CKPNT CS is executed only when system parameter JCKPNT=.TRUE.; each execution generates a new cycle on a predesignated checkpoint file. The checkpoint cycle contains the internal system and run-life data base information required to preserve the current operating environment for use in subsequent restart runs.

Format: **[label␢]CKPNT[␢STOP] $**

label label name

STOP optional keyword. If specified on an executed CKPNT CS, the run will terminate following the generation of the checkpoint file cycle preserving the current-run operating environment.

Examples:

LABEL1 CKPNT $

CKPNT STOP $

Restriction: The CKPNT CS is valid within any input stream except for a primary input stream inserted in an ANOPP restart run.

3.10.3 RSTRT CS

Purpose: The RSTRT control statement identifies a restart run and reestablishes a checkpointed operating environment. It identifies, through the specification or omission of keyword parameters, the name of the external restart file and the checkpoint cycle on the file where the operating environment to be reestablished is preserved.

The RSTRT CS also allows the user to specify new values for initialization and system parameters. Table 5 in this section presents those parameters subject to RSTRT CS specification.

Format:

RSTRT[␢RSTRT=sefn][␢RSCYCL=cn][␢paramname$_1$=value$_1$,...,␢paramname$_n$=value$_n$] $

RSTRT=sefn optional keyword specification of the external file name (sefn) by which the restart file must be currently assigned to the user's job. If omitted, RSTRT=RSFILE is assumed.

RSCYCL=cn optional keyword specification of the integer checkpoint cycle number on the restart file from which the operating environment is to be reestablished at the initiation of this run. If omitted, the last cycle on the restart file is used.

paramname optional specification of the name of a run-control initialization or system parameter whose checkpointed value is to be changed. (See table 5.)

value the new value to be assigned the specified paramname at initialization of the restart run

Examples:

```
RSTRT $
RSTRT RSTRT=RFILEIN LENGL=30000 CKPNT=NEWRFLE $
RSTRT RSCYCLE=3, JECHO=.TRUE. $
```

Restrictions:

RSTRT CS is valid only as the first CS and, optionally, the only CS in a restart run input deck.

The specified or assumed external file name (sefn) of the restart file may not be the same as the external file name of any sequential library file or run-life data-unit file used within the run.

3.10.4 Examples

Checkpoint run example 1: For the following example, the run is initialized with checkpoint processing active via the initialization parameter CKPNT specification on the ANOPP CS; system parameter JCKPNT is automatically initialized as .TRUE.; and external sequential file JETFILE is identified as the checkpoint file:

```
       ANOPP CKPNT=JETFILE $
       STARTCS $
         .
CP1    CKPNT $
         .
       IF (KEYITEM .EQ. .TRUE.) GOTO CP2 $
       SETSYS JCKPNT=.FALSE. $
CP2    CKPTN STOP $
         .
       SETSYS JCKPNT=.TRUE. $
CP3    CKPNT STOP $
         .
       ENDCS $
```

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 5.- RSTRT CS INITIALIZATION AND SYSTEM PARAMETERS

Parameter name	Parameter type	Description	Value		Default value
			Type	Range	
ACCOUNT	System[a]	Controls accumulation of cost-accounting data associated with each functional module or executive processor ACCOUNT=.TRUE. - accumulate accounting data ACCOUNT=.FALSE. - do not accumulate accounting data	Logical	.TRUE. .FALSE.	.FALSE.
CKPNT	Initialization	Identifies name of external checkpoint file to be generated during run. Its specification also results in system parameter JCKPNT being initialized as .TRUE., automatically activating checkpoint processing in the run	Alphanumeric name of seven characters or less		CPFILE
JECHO	System[a]	Controls printing of CS card image upon validation in primary edit phase (primary input stream editing) and secondary edit phases (procedure member-editing) JECHO=.TRUE. - print CS card images JECHO=.FALSE. - do not print CS card images	Logical	.TRUE. .FALSE.	.FALSE.
JLOG	System[a]	Control printing of CS card images upon execution in executive processing phases JLOG=.TRUE. - print CS card images JLOG=.FALSE. - do not print CS card images	Logical	.TRUE. .FALSE.	.TRUE.
LENGL	Initialization	Controls size (number of computer words within user's job field length) assigned to global dynamic stores (GDS) for this ANOPP run	Integer	≥3000	12000
NAETD	Initialization	Controls number of table-directory entries initially allocated for this ANOPP run	Integer	≥1	10
NAEUD	Initialization	Controls number of data unit directory entries initially allocated for this ANOPP run	Integer	≥5	25
NLPPM	Initialization	Controls number of lines per page to be used for ANOPP printed output during this run	Integer	>15	48
NOGO	Initialization	Determine if ANOPP run is to be limited to primary-edit-phase execution only NOGO=.TRUE. - primary-edit-phase-only run NOGO=.FALSE. - normal ANOPP run	Logical	.TRUE. .FALSE.	.FALSE.

[a]Usage discussions for each of the initialization parameters are presented in section 3.1.3 and for each of the system parameters in section 3.2.

Checkpoint CP1 will be executed. However, the next checkpoint to be executed depends on the result of the IF CS test. Checkpoint CP2 will be executed if logical user param KEYITEM=.TRUE. followed by run termination via the STOP keyword specification on the CKPNT CS. Otherwise, the SETSYS CS preceding checkpoint CP2 will deactivate checkpoint processing and CP2 CKPNT will be ignored. In this case, checkpoint CP3 would subsequently be executed and terminate the run.

Checkpoint run example 2: For the following example, the run would be initialized with checkpoint processing inactive (system param JCKPNT set to default value .FALSE.). Checkpoint CKP1 would not be executed.

```
        STARTCS $
           :
CKP1    CKPNT $
           :
        SETSYS JCKPNT=.TRUE. $
CKP2    CKPNT $
           :
        ENDCS $
```

The SETSYS CS specification of JCKPNT=.TRUE. would initialize checkpoint processing in the run. The checkpoint file used will have the default external name of CPFILE. Checkpoint CKP2 will be executed, followed by continued run processing since the CKPNT CS keyword STOP was not specified.

Restart run example 1: Assume that the following CS is the only card image in the user's input stream, that this is a restart of the run shown in checkpoint run example 1 above, and that the CKPNT CS labeled CP2 was the last checkpoint executed in that run:

```
RSTRT  RSTRT=JETFILE  CKPNT=NEWCPF $
```

The operating environment preserved by CKPNT CP2 will be reestablished since RSTRT CS keyword RSCYCL was not specified and the last cycle found on the restart file is used by default. Since the RSTRT CS also specifies initialization parameter CKPNT, checkpoint processing will also be initially active during this restart run. Processing will resume with the execution of the control statement in the input stream of the checkpointed run that follows the CKPNT CS labeled CP2. Subsequent execution of CKPNT control statements will result in new checkpoint cycles being generated on checkpoint file NEWCPF.

Restart run example 2: Again, assume that the following is a restart of the run shown in checkpoint run example 1 above. Since the RSTRT CS keyword RSTRT= is not specified, the system assumes that the user's restart file is assigned to this job by the default external file name RSFILE. Keyword RSCYCL specified that cycle 1 on the file, generated by the CKPNT CS labeled CP1 in the checkpointed run, will be used to reestablish the operating environment. The user has also chosen to reset the checkpointed value of system-parameter JECHO to .FALSE..

```
        RSTRT RSCYCL=1 JECHO=.FALSE. $
        STARTCS $
          :
        EXECUTE FMJET $
        IF (FMRESULT .EQ. .TRUE.) GOTO ENDIT $
        SETSYS JRSTRT=.FALSE. $
ENDIT ENDCS $
```

Following initialization, processing will resume with the execution of the inserted primary input stream above. Following the execution of the ENDCS ending the inserted stream, the system will check the current value of system parameter JRSTRT:

If JRSTRT=.FALSE., restart run processing terminates.

If JRSTRT is still set at restart-run initial value .TRUE., processing will continue with the execution of the control statement following the CKPNT CS labeled CP1 in the input stream of the checkpointed run.

4 ANOPP FUNCTIONAL MODULES

This section describes the functional modules, provides the instructions for their use, documents the inputs required for execution, and describes the resulting outputs. Each functional module is implemented using the EXECUTE CS as documented in section 3.4.2.

The functional modules described in this section are documented with internal documentation that is maintained in the FORTRAN source code. For ease of use, it is arranged in accordance with the format described in figure 1. When questions arise concerning the operation of a functional module, the source code should be consulted for the exact requirements and results. This is done to ensure that the correct documentation be available for each version of the program in existence. The level described in this manual is Level 03/00/00.

4.1 OVERVIEW

The functional modules installed in ANOPP fall into seven categories:

1. Aircraft flight dynamics
2. Propagation effects
3. Source noise parameters
4. Propagation
5. Received noise
6. Utilities
7. Noise sources

The ANOPP User's Manual documents modules in the same order as the ANOPP Theoretical Manual (ref. 1) documents prediction methods, since these manuals are meant to be companion volumes.

ORIGINAL PAGE IS
OF POOR QUALITY

```
COLUMN
0   0   1 1 1 1
1   5   0 2 4 6
***
*       PURPOSE - short description of the functional module (1 - 2 sentences)
*       AUTHOR -  initials and level number, such as L01/00/00
*       INPUT
*         USER PARAMETERS
*           Name_1 - description - default value
*           .
*           .
*           .
*           Name_n - description - default value
*         MEMBERS AND TABLES
*           DATA UNIT(DATA MEMBER) - short description of data requirement
*       OUTPUT
*         SYSTEM PARAMETERS
*           Name - description
*         USER PARAMETER - same as for INPUT
*         MEMBERS AND TABLES
*           DATA UNIT(DATA MEMBER) - short description of data requirements
*       DATA BASE STRUCTURES
*         DATA UNIT(DATA MEMBER) - complete description of data and required
*                                  format
*       ERRORS
*         NON-FATAL - description of errors that are possible within the
*                     functional module.
*         FATAL - functional modules are requested to not use fatal errors.
*                 In the event this is not followed they must be documented
*                 in this section.
*       LDS REQUIREMENTS - describes the amount of local dynamic storage
*                          required by this module.
*       GDS REQUIREMENTS - describes the amount of global dynamic storage
*                          required for this module.
```

Figure 1.- ANOPP functional module prologue format.

ORIGINAL PAGE IS
OF POOR QUALITY

4.2 AIRCRAFT FLIGHT DYNAMICS

4.2.1 Atmospheric Module (ATM)

```
PURPOSE - BUILD TABLE OF ATMOSPHERIC MODEL DATA AS FUNCTIONS
          OF ALTITUDE

AUTHOR  - SWP(L03/00/00)
          MMF(L03/00/09)

INPUT
  USER PARAMETERS                                      TYPE DEFAULT
    DELH        ALTITUDE INCREMENT FOR OUTPUT           RS    100.
                M (FT)
    H1          GROUND LEVEL ALTITUDE                   RS    0.
                REFERENCED TO SEA LEVEL
                M (FT)
    IUNITS      INPUT UNITS CODE                        A     2HSI
                =2HSI     , INPUTS ARE IN SI
                            UNITS
                =7HENGLISH, INPUTS ARE IN ENGLISH
                            UNITS
    NHO         NUMBER OF ALTITUDES FOR OUTPUT          I     1
                ATMOSPHERIC FUNCTIONS
    P1          ATMOSPHERIC PRESSURE AT                 RS    101325.
                GROUND LEVEL     N/M**2 (LBF/FT**2)
    IPRINT      PRINT CODE FOR FORTRAN WRITE            I     3
                  0   NO PRINT DESIRED
                  1   INPUT PARAMETER PRINT ONLY
                  2   OUTPUT PRINT ONLY
                  3   BOTH INPUT PARAMETER AND OUTPUT PRINT
  MEMBER
    ATM( IN )

TEMPORARIES
  MEMBER
    SCRATCH( TAB1 )

OUTPUT
  SYSTEM PARAMETER
    NERR        EXECUTIVE SYSTEM PARAMETER FOR ERROR ENCOUNTERED
                DURING EXECUTION OF A FUNCTIONAL MODULE. NERR
                SET TO .TRUE. IF ERROR ENCOUNTERED.
  MEMBER
    ATM( TMOD )
```

**ORIGINAL PAGE IS
OF POOR QUALITY**

Atmospheric Module (ATM)

DATA BASE STRUCTURES

ATM(IN) CONTAINS DATA INPUT TO ATM IN FOLLOWING FORMAT

RECORD	FORMAT	DESCRIPTION
1	3RS	ALT, TEMP, RELATIVE HUMIDITY (ALTITUDE, "ALT", IS REFERENCED TO SEA LEVEL AND SHOULD NOT BE LESS THAN USER PARAMETER H1.)
.	.	
.	.	

ALTITUDE UNITS	M(FT)
TEMPERATURE UNITS	KELVIN(RANKINE)
RELATIVE HUMIDITY	PERCENT

SCRATCH([illegible])

TEMPORARY TWO-DIMENSIONAL TYPE 1 DATA TABLE
- INDEPENDENT VARIABLES
  1. ALTITUDE
  2. ORDERED POSITION
- DEPENDENT VARIABLES IN FOLLOWING ORDER
  - TEMPERATURE
  - HUMIDITY

ATM(TMOD) OUTPUT TWO-DIMENSIONAL TYPE 1 DATA TABLE OF ATMOSPHERIC MODEL VALUES IN DIMENSIONLESS UNITS
- INDEPENDENT VARIABLES
  1. ALTITUDE (REFERENCED TO GROUND LEVEL)
  2. ORDERED POSITION
- DEPENDENT VARIABLES IN FOLLOWING ORDER
  - PRESSURE
  - DENSITY
  - TEMPERATURE
  - SPEED OF SOUND
  - AVERAGE SPEED OF SOUND
  - HUMIDITY
  - COEFFICIENT OF VISCOSITY
  - COEFFICIENT OF THERMAL CONDUCTIVITY
  - CHARACTERISTIC IMPEDANCE(RHO*C)

ERRORS

NON-FATAL
1. USER PARAMETER NHO IS OUT OF RANGE
2. MEMBER CONTAINING INPUT DATA NOT AVAILABLE
3. LOCAL DYNAMIC STORAGE INSUFFICIENT
4. ERROR OCCURRED IN TABLE BUILD ROUTINE WHICH PREVENTED THE BUILDING OF A TABLE.
5. MEMBER CONTAINING INPUT DATA INVALID

FATAL - NONE

ORIGINAL PAGE IS
OF POOR QUALITY

4.2.2 Geometry Module (GEO)

PURPOSE - TO CALCULATE THE SOURCE TO OBSERVER GEOMETRY

AUTHOR REG(L03/00/00)
WKB(L03/00/01)
WKB(L03/00/03)
MMF(L03/00/09)

INPUT

USER PARAMETERS

		DEFAULT SI UNITS
AW	REFERENCE AREA OF THE AIRCRAFT (EXAMPLE, WING AREA), M**2 (FT**2)	1.00
CTK	CHARACTERISTIC TIME CONSTANT	1.00
DELDB	LIMITING NOISE LEVEL, DOWN FROM THE PEAK(DB)	20.
MASSAC	REFERENCE MASS OF THE AIRCRAFT, KG (SLUGS)	416.8
START	INITIAL FLIGHT TIME TO BE CONSIDERED, S	0.
STOP	FINAL FLIGHT TIME TO BE CONSIDERED, S	9999.
DELT	RECEPTION TIME INCREMENT, S	.5
DELTH	MAXIMUM POLAR DIRECTIVITY ANGLE LIMIT, DEG	10.
ICOORD	OUTPUT OPTION 0 NEITHER BODY NOR WIND OUTPUT REQUIRED 1 BODY AXES OUTPUT REQUIRED 2 WIND AXES OUTPUT REQUIRED 3 BOTH WIND AND BODY AXES OUTPUT REQUIRED	3
DIRECT	=.FALSE., INTERPOLATE FROM FLI(PATH) OBSERVER RECEPTION TIMES AND GEOMETRY BASED ON USER PARAMETERS START, STOP, DELTH, AND DELT =.TRUE. , CALCULATE OBSERVER RECEPTION TIMES AND GEOMETRY BASED ON THE EXACT FLIGHT TIMES AS READ FROM UNIT MEMBER FLI(PATH)	.FALSE.
IPRINT	OUTPUT PRINT OPTION 0 NO PRINT DESIRED 1 INPUT PRINT ONLY 2 OUTPUT PRINT ONLY 3 BOTH INPUT AND OUTPUT PRINT	3
IUNITS	UNITS FLAG FOR BOTH INPUT AND OUTPUT 2HSI METRIC UNITS 7HENGLISH ENGLISH UNITS	2HSI

MEMBERS
FLI(PATH)

ORIGINAL PAGE IS
OF POOR QUALITY

Geometry Module (GEO)

GEO(SOURCE) - NEEDED IF OTHER THAN BODY AND WIND AXES SYSTEMS ARE TO BE OUTPUT.
OBSERV(COORD) - GEOMETRY OF OBSERVERS

OUTPUT
USER PARAMETERS
GEOERR GEOMETRY ERROR FLAG
0 NO OUTPUT WITHIN RANGE
1 SUCCESSFUL COMPLETION

MEMBERS
GEO(BODY) - GEOMETRY ASSOCIATED WITH BODY AXIS SYSTEM
GEO(WIND) - GEOMETRY ASSOCIATED WITH WIND AXIS SYSTEM
GEO(XXXX) - GEOMETRY ASSOCIATED WITH USER SUPPLIED SOURCE COORDINATE SYSTEM(S). THE MEMBER NAME(S) XXXX IS GIVEN BY THE SECOND ELEMENT OF EACH RECORD FROM UNIT MEMBER GEO(SOURCE). THE NUMBER OF RECORDS IN GEO(SOURCE) DETERMINES THE NUMBER OF GEO(XXXX) UNIT MEMBERS. IT IS IMPORTANT THAT THE SECOND ELEMENT OF EACH RECORD IN GEO(SOURCE) BE UNIQUE TO ALLOW EACH COORDINATE SYSTEM TO HAVE ITS OWN MEMBER NAME ON UNIT GEO.

TEMPORARY
MEMBERS
SCRATCH(FLITAB) - TABLE OF FLIGHT DATA CREATED AND USED BY GEO
SCRATCH(FLIGHT) - TEMPORARY MEMBER CONTAINING OUTPUT DATA PRIOR TO ARRANGING IT ONTO THE OUTPUT DATA MEMBERS

DATA STRUCTURES
ATM(TMOD) - TYPE 1 DATA TABLE OF ATMOSPHERIC MODEL CONTAINING EIGHT FUNCTIONS OF ALTITUDE. DEPENDENT VARIABLES ARE IN THE FOLLOWING ORDER BY ORDERED POSITION

1. ATMOSPHERIC PRESSURE
2. DENSITY
3. TEMPERATURE
4. SPEED OF SOUND
5. AVERAGE SPEED OF SOUND
6. RELATIVE HUMIDITY
7. COEFFICIENT OF VISCOSITY
8. COEFFICIENT OF THERMAL CONDUCTIVITY
9. CHARACTERISTIC IMPEDANCE

ORIGINAL PAGE IS
OF POOR QUALITY

Geometry Module (GEO)

FLI(PATH) - INPUT MEMBER IN 10RS FORMAT, EACH RECORD CONTAINS THE A/C TRAJECTORY DATA IN THE FOLLOWING ORDER

- WORD 1 - FLIGHT TIME
- WORD 2 - A/C X LOCATION (EARTH FIXED)
- WORD 3 - Y LOCATION (EARTH FIXED)
- WORD 4 - Z LOCATION (EARTH FIXED)
- WORD 5 - PSI EULER ANGLE (EARTH FIXED - BODY)
- WORD 6 - TH EULER ANGLE (EARTH FIXED - BODY)
- WORD 7 - PHI EULER ANGLE (EARTH FIXED - BODY)
- WORD 8 - PSI EULER ANGLE (BODY - WIND)
- WORD 9 - TH EULER ANGLE (BODY - WIND)
- WORD 10 - PHI EULER ANGLE (BODY - WIND)

GEO(SOURCE)- INPUT MEMBER CONTAINING DESCRIPTIONS OF SOURCE COORDINATE SYSTEMS FOR WHICH OUTPUT IS REQUIRED THE MEMBER CONSISTS OF RECORDS IN I,A8,6RS,A80 FORMAT WHERE

- ELEMENT 1 - SOURCE COORDINATE INDEX
- ELEMENT 2 - SOURCE COORDINATE NAME (THIS NAME WILL BE THE MEMBER NAME XXXX USED ON UNIT MEMBER GEO(XXXX))
- ELEMENT 3 - X OFFSET FROM THE BODY ORIGIN
- ELEMENT 4 - Y OFFSET FROM THE BODY ORIGIN
- ELEMENT 5 - Z OFFSET FROM THE BODY ORIGIN
- ELEMENT 6 - PSI EULER ANGLE(BODY - SOURCE)
- ELEMENT 7 - TH EULER ANGLE(BODY - SOURCE)
- ELEMENT 8 - PHI
- ELEMENT 9 - SOURCE COORDINATE DESCRIPTION

GEO(XXXX) -
THE FORMAT OF GEO(XXXX) IS AS FOLLOWS:

RECORD	WORD	DESCRIPTION
1		RECORD FORMAT IS I,3RS,I,RS
	1	OBSERVER INDEX FOR FIRST OBSERVER
	2	X COORDINATE OF OBSERVER
	3	Y COORDINATE OF OBSERVER
	4	Z COORDINATE OF OBSERVER
	5	NUMBER OF RECEPTION TIMES ASSOCIATED WITH THIS OBSERVER (ASSUME VALUE IS N)
	6	OBSERVER'S HEIGHT
2		RECORD FORMAT IS *RS
	1 . . N	RECEPTION TIMES FOR CURRENT OBSERVER INDEX

ORIGINAL PAGE IS
OF POOR QUALITY

Geometry Module (GEO)

RECORDS 3 THROUGH N+2 CONTAIN GEOMETRY DATA FOR EACH RECEPTION TIME. RECORD 3 CONTAINS GEOMETRY DATA FOR THE FIRST RECEPTION TIME, RECORD 4 FOR THE SECOND RECEPTION TIME,... RECORD N+2 FOR THE N TH RECEPTION TIME.

```
 3               RECORD FORMAT IS *RS
           1     DISTANCE OF SOURCE FROM OBSERVER
           2     EMISSION TIME
           3     DIRECTIVITY ANGLE
           4     ELEVATION ANGLE
           5     AZIMUTH ANGLE

 4         REPEAT OF RECORD 3 FOR SECOND RECEPTION TIME
 .         .
 .         .
N+3              RECORD FORMAT IS *RS
           1     OBSERVER INDEX FOR SECOND OBSERVER
           2     X COORDINATE OF OBSERVER
           3     Y COORDINATE OF OBSERVER
           Y     Z COORDINATE OF OBSERVER
           5     NUMBER OF RECEPTION TIMES ASSOCIATED WITH
                 THIS OBSERVER (ASSUME VALUE IS M)

N+4              RECORD FORMAT IS *RS
           1
           .     RECEPTION TIMES FOR CURRENT OBSERVER
           .     INDEX
           M
```

RECORD N+5 THROUGH RECORD N+M+4 CONTAIN GEOMETRY DATA FOR EACH RECEPTION TIME STORED IN THE SAME MANNER AS DESCRIBED ABOVE IN RECORDS 3 THROUGH N+2.

THE PATTERN AS SEEN IN RECORDS 1 THROUGH N+2 AND RECORDS N+3 THROUGH N+M+4 CONTINUES FOR ALL OBSERVERS

OBSERV(COORD) - MULTI-RECORD MEMBER WITH FORMAT = 4H3RS$, CONTAINING ONE RECORD FOR EACH OBSERVER WITH VALUES OF THE X, Y, Z COORDINATES

GEO(BODY) - GEOMETRY ASSOCIATED WITH BODY AXES - SAME FORMAT AS FOR GEO(XXXX)

GEO(WIND) - GEOMETRY ASSOCIATED WITH WIND AXES - SAME FORMAT AS FOR GEO(XXXX)

ORIGINAL PAGE IS
OF POOR QUALITY

Geometry Module (GEO)

SCRATCH(FLITAB)-TYPE ONE DATA TABLE OF NINE FUNCTIONS OF FLIGHT TIME ARRANGED IN THE FOLLOWING ORDER

1. FLIGHT TIME
2. A/C X (EARTH FIXED)
3. A/C Y (EARTH FIXED)
4. A/C Z (EARTH FIXED)
5. PSI EULER ANGLE (EARTH FIXED - BODY)
6. TH EULER ANGLE (EARTH FIXED - BODY)
7. PHI EULER ANGLE (EARTH FIXED - BODY)
8. PSI EULER ANGLE (BODY - WIND)
9. TH EULER ANGLE (BODY - WIND)
10. PHI EULER ANGLE (BODY - WIND)

SCRATCH(FLIGHT)-TEMPORARY DATA UNIT(DATA MEMBER) TO HOLD OUTPUT DATA PRIOR TO TRANSLATION INTO OUTPUT COORDINATE SYSTEMS. THE MEMBER CONSISTS OF RECORDS IN 11RS FORMAT IN THE FOLLOWING ORDER

1. RECEIVED TIME
2. FLIGHT TIME
3. A/C X (EARTH FIXED)
4. A/C Y (EARTH FIXED)
5. A/C Z (EARTH FIXED)
6. PSI EULER ANGLE (EARTH FIXED - BODY)
7. TH EULER ANGLE (EARTH FIXED - BODY)
8. PHI EULER ANGLE (EARTH FIXED - BODY)
9. PSI EULER ANGLE (BODY - WIND)
10. TH EULER ANGLE (BODY - WIND)
11. PHI EULER ANGLE (BODY - WIND)

ERRORS

NON-FATAL

1. INSUFFICIENT LDS FOR EXECUTION
2. MORE THAN 18 SOURCE COORDINATE SYSTEMS DEFINED
3. UNABLE TO INTERPOLATE TABLE ATM(TMOD)
4. USER PARAMETER VALUE OUT OF RANGE, DEFAULT TO BE USED
5. NO FLIGHT TIMES THAT CORRESPOND TO USER SUPPLIED INPUT PARAMETERS START AND STOP

LDS REQUIREMENTS

LENGTH = 8*(NSOURCE+2) + 3*NUMOBS + MAXTIMES + 10*NTIMES

WHERE

NSOURCE = NUMBER OF COORDINATE SYSTEMS
NUMOBS = NUMBER OF OBSERVERS
MAXTIMES = MAXIMUM NUMBER OF TIMES FOR ANY OBSERVER
NTIMES = NUMBER OF FLIGHT PATH TIMES

GDS REQUIREMENTS

ALLOCATION REQUIRED FOR TABLES ATM(TMOD) AND SCRATCH(FLITAB)

ORIGINAL PAGE IS
OF POOR QUALITY

4.2.3 Flight Dynamics Module (FLI)

PURPOSE - COMPUTES THE TWO-DIMENSIONAL TRAJECTORY OF AN AIRCRAFT AS A FUNCTION OF TIME.

AUTHOR - SLP(L03/00/00)
CBF(L03/00/05)
DSW(L03/00/07)
SLP(L03/00/07)
MMF(L03/00/09)

INPUT

USER PARAMETERS

		DEFAULT SI UNITS
IUNITS	SYSTEM OF UNITS 2HSI SI 7HENGLISH ENGLISH	2HSI
NJO	NO. OF INTEGRATION STEPS COMPLETED	0
IPRINT	INTEGER PRINT OPTION 0 NO PRINT DESIRED 1 INPUT PRINT ONLY 2 OUTPUT PRINT ONLY 3 INPUT AND OUTPUT PRINT	3
IPTAPE	INTEGER TAPE OUTPUT OPTION 0 NO PLOT FILE GENERATED 1 TT,VA,XA,ZA,THW,ANGLE-OF ATTACK AND POWER SETTING WRITTEN TO TAPE7 AT EACH TIME STEP. FORMAT(7F15.4) HEADER RECORD FORMAT(2A15,2F15.4,A15)	
IOUT	OUTPUT MEMBER OPTION 0 NO OUTPUT MEMBER 1 OUTPUT IN ORIGINAL UNITS	0
J	INITIAL INTEGRATION STEP NUMBER	1
TSTEP	MAXIMUM ACCEPTABLE TIME INTERVAL BETWEEN INTEGRATION STEPS, SEC	1.0
MSTEP	MAXIMUM MACH NUMBER INCREMENT BETWEEN ENGINE TABLE ENTRIES	0.1
XI	DAMPING COEFFICIENT FOR PITCH DYNAMICS	.95
PITCH	PITCH ANGLE, DEG	0.0
TLG	LANDING GEAR RETRACTION TIME, SEC	0.
	** REFERENCE VALUES **	
AE	ENGINE INLET REFERENCE AREA, M**2 (FT**2)	PI/4
AW	WING REFERENCE AREA, M**2 (FT**2)	100.

ORIGINAL PAGE IS
OF POOR QUALITY

Flight Dynamics Module (FLI)

B	WING SPAN, M (FT)	20.
M1	FULLY LOADED REFERENCE MASS, KG (SLUGS)	10000.
Z1	ALTITUDE AT BRAKE RELEASE, M (FT)	0.
TAU	COEFFICIENT OF ROLLING FRICTION	0.01
NENG	NUMBER OF ENGINES IN EACH GROUP	1,0,0,0
E	ENGINE INCLINATION ANGLE WRT BODY AXES FOR EACH ENGINE GROUP, DEG	0.,0.,0.,0.
ENGNUM	ENGINE IDENTIFIER ARRAY FOR ENGINE GROUP	3HEN1,... 3HEN4
	** CONDITIONS AT START OF FLIGHT SEGMENT **	
TT	INITIAL TIME, SEC	0.0
VA	INITIAL VELOCITY, M/S (FT/S)	0.0
XA	INITIAL DISTANCE FROM ORIGIN, M (FT)	0.0
YA	INITIAL SIDELINE DISTANCE, M (FT)	0.0
ZA	INITIAL ALTITUDE (ZA=Z1 IMPLIES AIRCRAFT IS ON GROUND), M (FT)	0.0
MASS	INITIAL MASS	10000.
THW	INITIAL FLIGHT PATH ANGLE, DEG	0.0
	** STOPPING CRITERIA **	
JF	FINAL INTEGRATION STEP NUMBER LIMIT	50
TF	FINAL TIME LIMIT, SEC	100.
VF	FINAL VELOCITY LIMIT, M/S (FT/S)	155.
XF	FINAL DISTANCE LIMIT, M (FT)	10000.
ZF	FINAL ALTITUDE LIMIT, M (FT)	1000.

NOTE: NORMAL TERMINATION OCCURS IF ANY OF THE ABOVE LIMITS IS REACHED OR IF THE LAST RECORD OF ANY OF THE PILOT CONTROL MEMBERS IS SATISFIED.

TABLES/MEMBERS
NOTE: THE FOLLOWING ARE UNIT (MEMBER) NAMES

```
ATM(TMOD)
AERO(CL)
AERO(CD)
AERO(CDLG)
ENG(DYN)
FLI(TEMP)    *NOT READ IF NJO=0*
FLI(FLIXXX)    *NOT OPENED IF IOUT=0*
```

ORIGINAL PAGE IS
OF POOR QUALITY

Flight Dynamics Module (FLI)

```
    FLI(PATH)     *NOT OPENED IF IOUT=0*
    PILOT(CONTROL)
    PILOT(FLAP)
    PILOT(POWER1)
    PILOT(POWER2)   * READ ONLY IF NENG HAS 2 OR MORE ELEMENTS
    PILOT(POWER3)   * READ ONLY IF NENG HAS 3 OR MORE ELEMENTS
    PILOT(POWER4)   * READ ONLY IF NENG HAS 4 ELEMENTS

OUTPUT
  USER PARAMETERS
    NJO         NO. OF INTEGRATION STEPS COMPLETED
    TT          ACTUAL FINAL TIME
    VA          ACTUAL FINAL VELOCITY
    XA          ACTUAL FINAL DISTANCE
    ZA          ACTUAL FINAL ALTITUDE
    MASS        FINAL MASS
    THW         ACTUAL FINAL FLIGHT PATH ANGLE
    PITCH       ACTUAL FINAL PITCH ANGLE
    HSTEP       FINAL TIME STEP SIZE

  MEMBER        DESCRIBED UNDER DATA BASE STRUCTURES
    FLI(PATH) - FLIGHT DATA AT EACH INTEGRATION STEP
                IN ORIGINAL UNITS.  OUTPUT IF IOUT>0
    FLI(TEMP) - TEMPORARY INFORMATION USED BY THE
                INTEGRATOR TO RESTART.
                MEMBER IS WRITTEN ONCE FOR EACH CALL
                TO THE FLI MODULE.
    FLI(FLIXXX) FLIGHT DATA AT A FEW SELECTED TIMES.  OUTPUT ONLY
                IF IOUT>0.
                ONE MEMBER CREATED FOR EACH ENGINE GROUP;  XXX
                REPLACED BY ENGNUM(I), I = 1,N, WHERE N =
                NUMBER OF ELEMENTS ASSOCIATED WITH PARAMETER, NENG

  SYSTEM PARAMETER
    NERR        SYSTEM ERROR PARAMETER
                .TRUE. - ERROR ENCOUNTERED DURING MODULE
                         EXECUTION
                .FALSE.- NO ERROR

DATA BASE STRUCTURES
  ATM( TMOD )    - TYPE 1 DATA TABLE OF ATMOSPHERIC MODEL OF
                   NINE FUNCTIONS OF ALTITUDE.
                   VARIABLES IN THE FOLLOWING ORDER:
                   (PRESSURE,DENSITY,TEMP.,SPEED OF SOUND,
                   AVERAGE SPEED OF SOUND,
                   ABSOLUTE HUMIDITY,COEFF. OF VISCOSITY,
                   COEFF. OF THERMAL CONDUCTIVITY,
                   CHARACTERISTIC IMPEDANCE)
```

ORIGINAL PAGE IS
OF POOR QUALITY

Flight Dynamics Module (FLI)

AERO(CL) - TYPE 1 DATA TABLE OF LIFT COEFF. AS A FUNCTION OF ANGLE-OF-ATTACK, FLAP SETTING AND NONDIMENSIONAL ALTITUDE.
ANGLES IN DEGREES AND ALTITUDE DIVIDED BY WING SPAN.

AERO(CD) - TYPE 1 DATA TABLE OF DRAG COEFF. AS A FUNCTION OF ANGLE-OF-ATTACK, FLAP SETTING AND NONDIMENSIONAL ALTITUDE
ANGLES IN DEGREES AND ALTITUDE DIVIDED BY WING SPAN

AERO(CDLG) - TYPE 1 DATA TABLE OF LANDING GEAR DRAG COEFF. AS A FUNCTION OF LIFT COEFF.

ENG(DYN) - TYPE 1 DATA TABLE OF ENGINE PERFORMANCE DATA; THREE FUNCTIONS OF MACH NUMBER AND POWER SETTING. DEPENDENT VARIABLES IN THE FOLLOWING ORDER:
SPECIFIC THRUST, RE CA
AIR FLOWRATE, RE RHOA*CA*AE
FUEL FLOWRATE, RE RHOA*CA*AE

PILOT(CONTROL)- INPUT UNIT MEMBER WITH FORMAT=2HCI.
EACH "CI" (IE, CARD IMAGE) RECORD HAS THE SAME FIELDS. THE FIELDS ARE AS FOLLOWS :

FIELD	CONTENTS
1	ONE OF THE FOLLOWING CONDITION PARAMETER NAMES : MACH — MACH NUMBER TIME — TIME RANGE — DOWNRANGE POSITION ATTACK — ANGLE OF ATTACK ALTITUDE — ALTITUDE PITCH — PITCH ANGLE PATH — FLIGHT PATH ANGLE
2	ONE OF THE FOLLOWING LOGICAL OPERATORS : .EQ. .NE. .LE. .LT. .GE. .GT.
3	CONDITION REFERENCE VALUE GIVEN BY EITHER A USER DEFINED PARAMETER NAME OR A "RS" VALUE

ORIGINAL PAGE IS
OF POOR QUALITY

4 ONE OF THE FOLLOWING GOAL PARAMETER NAMES :

MACH	MACH NUMBER
ATTACK	ANGLE OF ATTACK
PITCH	PITCH ANGLE
PITCHDT	PITCH ANGLE RATE OF CHANGE
PATH	FLIGHT PATH
PATHDT	FLIGHT PATH RATE OF CHANGE

5 GOAL REFERENCE VALUE GIVEN BY EITHER A USER DEFINED PARAMETER NAME OR A "RS" VALUE

6 FORCING FUNCTION CONSTANT GIVEN BY EITHER A USER DEFINED PARAMETER NAME OR A REAL SINGLE (RS) PRECISION VALUE

THE VALUES SPECIFIED BY FIELDS 4, 5, AND 6 IN RECORD 1 REMAIN IN EFFECT AS LONG AS THE CONDITION AS REFLECTED BY FIELDS 1, 2 AND 3 REMAINS .TRUE. ONCE THIS CONDITION BECOMES .FALSE., FIELDS 1, 2 AND 3 OF THE NEXT RECORD ESTABLISH THE NEXT CONDITION TO BE TESTED. IF THIS CONDITION IS .TRUE., THEN FIELDS 4, 5 AND 6 SPECIFY VALUES TO BE USED. WHEN THE CONDITION IS .FALSE., FIELDS 1, 2 AND 3 OF THE NEXT RECORD ESTABLISH THE NEXT CONDITION TO BE TESTED. THIS PATTERN CONTINUES UNTIL THE LAST RECORD IS PROCESSED.

PILOT(FLAP) - INPUT UNIT MEMBER WITH FORMAT=2HCI. EACH RECORD HAS THE SAME FIELDS. THE FIELDS ARE AS FOLLOWS:

FIELD CONTENTS

1 ONE OF THE FOLLOWING CONDITION PARAMETER NAMES :

MACH	MACH NUMBER
TIME	TIME
RANGE	DOWNRANGE POSITION
ALTITUDE	ALTITUDE
FLAP	FLAP ANGLE

2 ONE OF THE FOLLOWING RELATIONAL OPERATORS :
.EQ. .NE. .LT. .LE. .GT. .GE.

3 CONDITION REFERENCE VALUE GIVEN BY EITHER A USER DEFINED PARAMETER NAME OR A "RS" VALUE

ORIGINAL PAGE IS
OF POOR QUALITY

4	ONE OF THE FOLLOWING GOAL PARAMETER NAMES : FLAP FLAP ANGLE FLAPDT FLAP ANGLE RATE OF CHANGE
5	GOAL REFERENCE VALUE GIVEN BY EITHER A USER DEFINED PARAMETER NAME OR A "RS" VALUE

THESE RECORDS ARE PROCESSED IN THE SAME MANNER AS DESCRIBED FOR UNIT MEMBER PILOT(CONTROL).

PILOT(POWER) - INPUT UNIT MEMBER WITH FORMAT=2HCI. EACH RECORD HAS THE SAME FIELDS. THE FIELDS ARE AS FOLLOWS:

FIELD	CONTENTS
1	ONE OF THE FOLLOWING CONDITION PARAMETER NAMES : MACH MACH NUMBER TIME TIME RANGE DOWNRANGE POSITION ALTITUDE ALTITUDE POWER POWER SETTING
2	ONE OF THE FOLLOWING RELATIONAL OPERATORS : .EQ. .NE. .LT. .LE. .GT. .GE.
3	CONDITION REFERENCE VALUE GIVEN BY EITHER A USER DEFINED PARAMETER NAME OR AS A "RS" VALUE
4	ONE OF THE FOLLOWING GOAL PARAMETER NAMES : POWER POWER SETTING POWERDT POWER SETTING RATE OF CHANGE
5	GOAL REFERENCE VALUE GIVEN BY EITHER A USER DEFINED PARAMETER NAME OR A "RS" VALUE

THESE RECORDS ARE PROCESSED IN THE SAME MANNER AS DESCRIBED FOR UNIT MEMBER PILOT(CONTROL).

FLI(PATH) - OUTPUT MEMBER IN *RS FORMAT CONTAINING THE FOLLOWING TRAJECTORY DATA IN EACH OF THE NJO RECORDS:
TIME,AIRCRAFT POSITION(X,Y,Z),EULER ANGLES FROM VEHICLE CARRIED TO BODY AXIS, AND EULER ANGLES FROM BODY TO WIND AXIS.

ORIGINAL PAGE IS
OF POOR QUALITY

Flight Dynamics Module (FLI)

EACH IS IN DIMENSIONAL UNITS.
INPUT MEMBER OF THIS FORM IS OPTIONAL.
USE ONLY IF FLIGHT PATH IS BUILT IN PIECES.

FLI(FLIXXX) - OUTPUT MEMBER IN 6RS,A4,2RS FORMAT. EACH RECORD CONTAINS FLIGHT TIME,MACH NUMBER,POWER SETTING, AMBIENT SPEED OF SOUND,DENSITY,VISCOSITY, LANDING GEAR POSITION,FLAP SETTING AND ABSOLUTE HUMIDITY.(LANDING GEAR POSITION VALUES ARE EITHER 4HDOWN OR 4HUP .)
ENGNUM(I) REPLACES XXX IN MEMBER NAME.
ONE RECORD IS WRITTEN FOR EACH CHANGE IN POWER SETTING, FLAP SETTING, LANDING GEAR POSITION, AND FOR EACH TIME MACH NUMBER CHANGES BY PARAMETER, MSTEP.
INPUT MEMBER IN THIS FORMAT IS OPTIONAL. USE ONLY IF FLIGHT PATH IS BUILT IN PIECES.

FLI(TEMP) - OUTPUT MEMBER IN *RS FORMAT CONTAINING ONE RECORD OF INTEGRATOR RESTART INFO. THIS MEMBER IS READ ONLY BY THE FLI MODULE ITSELF AND ONLY WHEN A FLIGHT PATH IS BUILT BY SUCCESSIVE CALLS TO FLI.

ERRORS
NON-FATAL
1. INSUFFICIENT SPACE IN LDS
2. INPUTS INVALID
3. MEMBER MANAGER ERROR ON OPENING DATA UNITS
4. ERROR IN INTEGRATION OR INTERPOLATION ROUTINE
5. ILLEGAL TRAJECTORY DETECTED

FATAL
NONE

LDS REQUIREMENTS

```
LENGTH = NEQ * ( 22 + NEQ ) + ( FLPMAX * 5 ) + ( CTLMAX * 6 ) +
    ( POWMAX(1) * 95 )
  + ( POWMAX(2) * 95 ) (* IF NENG HAS 2 OR MORE ELEMENTS *)
  + ( POWMAX(3) * 95 ) (* IF NENG HAS 3 OR MORE ELEMENTS *)
  + ( POWMAX(4) * 95 ) (* IF NEMG HAS 4 ELEMENTS         *)
  WHERE
    NEQ       = NUMBER OF EQUATIONS PASSED TO THE INTEGRATION
                ROUTINE
    FLPMAX    = NUMBER OF RECORDS ON THE PILOT'S FLAP CONTROL
                MEMBER
    CTLMAX    = NUMBER OF RECORDS ON THE PILOT'S CONTROL MEMBER
    POWMAX(1) = NUMBER OF RECORDS ON THE PILOT'S POWER SETTING
                ARRAY, FOR FIRST ENGINE GROUP
```

ORIGINAL PAGE IS
OF POOR QUALITY

Flight Dynamics Module (FLI)

```
POWMAX(2) = NUMBER OF RECORDS ON THE PILOT'S POWER SETTING
            ARRAY, FOR SECOND ENGINE GROUP
POWMAX(3) = NUMBER OF RECORDS ON THE PILOT'S POWER SETTING
            ARRAY, FOR THIRD ENGINE GROUP
POWMAX(4) = NUMBER OF RECORDS ON THE PILOT'S POWER SETTING
            ARRAY, FOR FOURTH ENGINE GROUP
```

```
GDS REQUIREMENTS
  SUFFICIENT ALLOCATION FOR THE FOLLOWING TABLES :
    ATM(TMOD)
    AERO(CL)
    AERO(CD)
    AERO(CDLG)
    ENG(DYN)
```

ORIGINAL PAGE IS
OF POOR QUALITY

4.3 PROPAGATION EFFECTS

4.3.1 Atmospheric Absorption Module (ABS)

```
PURPOSE. COMPUTE AVERAGE ABSORPTION/WAVELENGTH AS FUNCTION
         OF ALTITUDE AND FREQUENCY AND BUILD TABLE

AUTHOR   SWP(L03/00/00)
         MMF(L03/00/09)

INPUT
  USER PARAMETERS                                        DEFAULT
                                                       IN SI UNITS

    IUNITS     INPUT UNITS CODE                          2HSI
               =2HSI     , INPUTS IN SI UNITS
               =7HENGLISH, INPUTS IN ENGLISH UNITS
    ABSINT     NUMBER OF INTEGRATION STEPS                5
    IPRINT     INTEGER PRINT OPTION                       3
               0    NO PRINT DESIRED
               1    INPUT PRINT ONLY
               2    OUTPUT PRINT ONLY
               3    INPUT AND OUTPUT PRINT DESIRED
    SAE        METHOD OPTION (L)                          F
               .TRUE.  - SAE ARP 866 METHOD
               .FALSE. - ANSI STANDARD METHOD
  MEMBERS      DESCRIBED UNDER DATA BASE STRUCTURES
    SFIELD( FREQ )
    NOTE: MEMBERS ARE SPECIFIED BY UNIT( MEMBER ) NAME
  TABLES       DESCRIBED UNDER DATA BASE STRUCTURES
    ATM( TMOD )

OUTPUT
  SYSTEM PARAMETER
    NERR       EXECUTIVE SYSTEM ERROR INDICATOR (L)
               .TRUE. - ERROR ENCOUNTERED DURING MODULE
                        EXECUTION
               .FALSE. - NO ERROR ENCOUNTERED
  TABLES       DESCRIBED UNDER DATA BASE STRUCTURES
    ATM( AAC )

DATA BASE STRUCTURES
  ATM( TMOD )  TWO-DIMENSIONAL TYPE ONE DATA TABLE OF
               ATMOSPHERIC MODEL OF NINE FUNCTIONS
                  INDEPENDENT VARIABLES
                    1. ALTITUDE
                    2. ORDERED POSITION
                  DEPENDENT VARIABLES
                  WITH VALUES IN DIMENSIONLESS UNITS
                  ARRANGED IN FOLLOWING ORDER -
                    1. PRESSURE
                    2. DENSITY
                    3. TEMPERATURE
```

ORIGINAL PAGE IS
OF POOR QUALITY

4. SPEED OF SOUND
5. AVERAGE SPEED OF SOUND
6. HUMIDITY
7. COEFFICIENT OF VISCOSITY
8. COEFFICIENT OF THERMAL CONDUCTIVITY
9. CHARACTERISTIC IMPEDANCE

LINEAR INTERPOLATION AND CLOSEST VALUE EXTRAPOLATION

ATM(AAC) TYPE ONE DATA TABLE OF ATMOSPHERIC ABSORPTION COEFFICIENT AS FUNCTION OF ALTITUDE AND FREQUENCIES. NO EXTRAPOLATION AND LINEAR INTERPOLATION

SFIELD(FREQ) 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF 1/3 OCTAVE BAND CENTER FREQUENCIES IN HERTZ

ERRORS

NON-FATAL

1. UNABLE TO INTERPOLATE TABLE ATM(TMOD)
2. UNABLE TO BUILD TABLE ATM(AAC)
3. INSUFFICIENT SPACE IN LDS FOR REQUIRED ALLOCATION

FATAL - NONE

ORIGINAL PAGE IS
OF POOR QUALITY

4.3.2 Ground Reflection and Attenuation Module (GRA)

PURPOSE - COMPUTE GROUND EFFECTS FACTOR AND BUILD TABLE

AUTHOR - SWP(L03/00/00)
MMF(L03/00/09)

```
INPUT
  USER PARAMETERS                                                DEFAULT
    COH        INCOHERENCE COEFFICIENT                            .01
    C          AMBIENT SPEED OF SOUND AT THE GROUND              340.294
               M/S,(FT/S)
    RHO        AMBIENT AIR DENSITY AT GROUND,                     1.225
               KG/M**3 (SLUG/FT**3)
    SIGMA      GROUND FLOW RESISTANCE,                           2.5*10**5
               KG/(S M**3) OR (SLUG/(S FT**3))
    GRARF      TWO WORD ARRAY CONTAINING FIRST AND                50.
               LAST VALUE OF FREQUENCY, HZ                        2000.
    GRARD      TWO WORD ARRAY CONTAINING FIRST AND                100.
               LAST VALUE OF DISTANCE,M(FT)                       1000.
    GRART      TWO WORD ARRAY CONTAINING FIRST AND                0.
               LAST VALUES OF THETA, DEG                          90.
    GRAID      INDEX FOR GROUND EFFECT OPTION                     1
                 ID = 0 IMPLIES HARD SURFACE SUCH AS CEMENT
                 ID = 1 IMPLIES TYPICAL SOFT GROUND
    GRAND      NUMBER OF GROUND DIPS TO BE INCLUDED               5
    GRANP      NUMBER OF POINTS IN EACH GROUND DIP                5
    GRANT      NUMBER OF INCIDENCE ANGLES                         10
    GRANF      NUMBER OF FREQUENCIES                              5
    GRANR2     NUMBER OF IMAGE DISTANCES                          5
    NSUBB      NUMBER OF SUBBANDS PER 1/3 OCTAVE BAND             1
    IPRINT     PRINT OPTION                                       3
                  0    NO PRINT
                  1    INPUT PARAMETER PRINT ONLY
                  2    OUTPUT PRINT ONLY
                  3    BOTH INPUT PARAMETER AND OUTPUT PRINT
 DATA BASE MEMBERS
   NONE
 OUTPUT
   SYSTEM PARAMETER
     NERR      SET TO .TRUE. IF ERROR ENCOUNTERED
 DATA BASE MEMBERS
   TAB(GE)     TYPE 1 DATA TABLE OF GROUND EFECT FACTOR AS
               FUNCTION OF PATH DIFFERENCE, COSINE OF INCIDENCE
               ANGLE BAND WIDTH FACTOR, AND IMAGE DISTANCE
                NOTE:  FOR CASE WHERE GRAID = 0
                GRANT,GRANF AND GRANR2 ARE SET = 1
                COSINE OF INCIDENCE ANGLE IS SET = 1
                BAND WIDTH FACTOR IS SET = 0
                IMAGE DISTANCE IS SET = 1
                SO THAT GROUND EFFECT FACTOR IS IN EFFECT A
```

ORIGINAL PAGE IS
OF POOR QUALITY

FUNCTION OF PATH DIFFERENCE ONLY,ALTHOUGH TABLE
IS FOUR-DIMENSIONAL

ERRORS

1. INSUFFICIENT SPACE IN LDS
2. ERROR IN ATTEMPT TO BUILD TABLE
3. USER PARAMETERS INVALID

ORIGINAL PAGE IS OF POOR QUALITY

4.4 SOURCE NOISE PARAMETERS

4.4.1 Fan Noise Parameters Module (PREFAN)

```
PURPOSE - TO GENERATE THE PHYSICAL PARAMETERS REQUIRED BY
          HEIDMAN'S METHOD FOR FAN NOISE PREDICTION FOR
          TURBOJET AND TURBOFAN ENGINES

AUTHOR  - CBF(L03/00/00)
          MMF(L03/00/09)

INPUT                                                    DEFAULT
  USER PARAMETERS
    IPRINT - PRINTED OUTPUT OPTION                          3
             0 - NO PRINT DESIRED
             1 - INPUT PRINT ONLY
             2 - OUTPUT PRINT ONLY
             3 - BOTH INPUT AND OUTPUT PRINT
    IUNITS - INPUT UNIT OPTION                            2HSI
             2HSI,      SI UNITS
             7HENGLISH, ENGLISH UNITS
    ITYPE  - TYPE OF INTERPOLATION TO BE USED ON THE        1
             INPUT TABLES
             1 - LINEAR INTERPOLATION
             2 - CUBIC SPLINE INTERPOLATION
    SCRXXX - THREE LETTER CODE XXX USED TO FORM THE      3HXXX
             UNIT MEMBER NAME FLI(FLIXXX).  IF THE
             VALUE OF USER PARAMETER SCRXXX
             IS 3HABC, THE FLI UNIT MEMBER NAME IS
             FLI(FLIABC).

  MEMBERS AND TABLES - DESCRIBED UNDER DATA BASE STRUCTURES
    FLI(FLIXXX) - (SEE USER PARAMETER SCRXXX)
    ENG(FAN1)
    ENG(FAN2)

OUTPUT
  SYSTEM PARAMETER
    NERR   - EXECUTIVE SYSTEM PARAMETER FOR ERROR ENCOUNTERED
             DURING EXECUTION OF A FUNCTIONAL MODULE
             =.TRUE.,  ERROR ENCOUNTERED
             =.FALSE., NO ERRORS

  USER PARAMETERS
    NTIMES - NUMBER OF SOURCE TIME VALUES
    TIMES  - MULTI-ELEMENT PARAMETER CONTAINING THE SOURCE
             TIMES
    MT1    - MULTI-ELEMENT PARAMETER CONTAINING THE FAN
             ENTRANCE MASS FLOW RATE AT EACH SOURCE TIME
    MT2    - MULTI-ELEMENT PARAMETER CONTAINING THE FAN
             EXIT MASS FLOW RATE FOR EACH SOURCE TIME
    MTA    - MULTI-ELEMENT PARAMETER CONTAINING THE AIRCRAFT
             MACH NUMBER FOR EACH TIME
```

ORIGINAL PAGE IS
OF POOR QUALITY

Fan Noise Parameters Module (PREFAN)

NT1 - MULTI-ELEMENT PARAMETER CONTAINING THE ROTATIONAL SPEED FOR EACH TIME

DT1 - MULTI-ELEMENT PARAMETER CONTAINING THE TOTAL TEMPERATURE RISE ACROSS FAN FOR EACH TIME

CTA - MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT SPEED OF SOUND FOR EACH TIME

RHOTA - MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DENSITY FOR EACH TIME

DATA BASE STRUCTURES

FLI(FLIXXX) MULTI-ELEMENT MEMBER WITH FORMAT=6RS,A4,2RS, WITH ONE RECORD PER SOURCE TIME CONTAINING THE FOLLOWING VALUES : (1) SOURCE TIME, (2) AIRCRAFT MACH NUMBER, (3) ENGINE POWER SETTING, (4) AMBIENT SPEED OF SOUND, (5) AMBIENT DENSITY, (6) AMBIENT DYNAMIC VISCOSITY, (7) LANDING GEAR POSITION, (8) FLAP SETTING AND (9) ABSOLUTE HUMIDITY. ONLY THE FIRST FIVE ENTRIES ARE USED.

ENG(FAN1) THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE FAN ENTRANCE FLOW STATE AS A FUNCTION OF (1) ENGINE POWER SETTING, (2) AIRCRAFT MACH NUMBER AND (3) ORDERED POSITION. THE FLOW STATE DEPENDENT VARIABLES IN ORDERED POSITION ARE : AREA, FUEL-TO-AIR RATIO, MASS FLOW RATE, TOTAL PRESSURE, TOTAL TEMPERATURE, AND ROTATIONAL SPEED.
(ONLY MASS FLOW RATE, ROTATIONAL SPEED, AND TOTAL TEMPERATURE ARE REQUIRED)

ENG(FAN2) THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE FAN EXIT FLOW STATE AS A FUNCTION OF (1) ENGINE POWER SETTING, (2) AIRCRAFT MACH NUMBER AND (3) ORDERED POSITION. THE FLOW STATE DEPENDENT VARIABLES IN ORDERED POSITION ARE : AREA, FUEL-TO-AIR RATIO, MASS FLOW RATE, TOTAL PRESSURE, TOTAL TEMPERATURE AND ROTATIONAL SPEED.
(ONLY MASS FLOW RATE AND TOTAL TEMPERATURE ARE REQUIRED)

ERRORS

NON-FATAL

1. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.
2. INSUFFICIENT LOCAL DYNAMIC STORAGE.
3. REQUIRED UNIT MEMBER NOT AVAILABLE.
4. UNABLE TO INTERPOLATE SPECIFIED TABLE.

FATAL - NONE

ORIGINAL PAGE IS
OF POOR QUALITY

Fan Noise Parameters Module (PREFAN)

```
LDS REQUIREMENTS
  LENGTH = ( NTIMES * ( NWORDS + ( NP - 1 ) } ) + 1
    WHERE NTIMES = NUMBER OF SOURCE TIMES
          NWORDS = NUMBER OF WORDS PER RECORD ON THE MEMBER
                   CONTAINING THE ENGINE VARIABLE DATA
          NP     = NUMBER OF USER PARAMETERS OUTPUT FROM
                   THIS MODULE

GDS REQUIREMENTS
  ALLOCATION REQUIRED FOR THE FOLLOWING TABLES
    1.  ENG(FAN1)
    2.  ENG(FAN2)
```

ORIGINAL PAGE IS
OF POOR QUALITY

4.4.2 Core Noise Parameters Module (PRECOR)

PURPOSE - TO GENERATE THE PHYSICAL PARAMETERS REQUIRED FOR A CORE NOISE PREDICTION MODULE

AUTHOR - CBF(L03/00/00)
MMF(L03/00/09)

INPUT		DEFAULT
USER PARAMETERS		
IPRINT	- PRINTED OUTPUT OPTION 0 - NO PRINT DESIRED 1 - INPUT PRINT ONLY 2 - OUTPUT PRINT ONLY 3 - BOTH INPUT AND OUTPUT PRINT	3
IUNITS	- INPUT UNIT OPTION 2HSI, SI UNITS 7HENGLISH, ENGLISH UNITS	2HSI
ITYPE	- TYPE OF INTERPOLATION TO BE USED ON THE INPUT TABLES 1 - LINEAR INTERPOLATION 2 - CUBIC SPLINE INTERPOLATION	1
SCRXXX	- THREE LETTER CODE XXX USED TO FORM THE UNIT MEMBER NAME FLI(FLIXXX). IF THE VALUE OF USER PARAMETER SCRXXX IS 3HABC, THE FLI UNIT MEMBER NAME IS FLI(FLIABC).	3HXXX

MEMBERS AND TABLES - DESCRIBED UNDER DATA BASE STRUCTURES
FLI(FLIXXX) - (SEE USER PARAMETER SCRXXX)
ENG(CORE1)
ENG(CORE2)

OUTPUT

SYSTEM PARAMETER
- NERR - EXECUTIVE SYSTEM PARAMETER FOR ERROR ENCOUNTERED DURING EXECUTION OF A FUNCTIONAL MODULE

USER PARAMETERS
- NTIMES - NUMBER OF SOURCE TIME VALUES
- TIMES - MULTI-ELEMENT PARAMETER CONTAINING THE SOURCE TIMES
- MT1 - MULTI-ELEMENT PARAMETER CONTAINING THE COMBUSTOR ENTRANCE MASS FLOW RATE VALUES FOR EACH TIME
- MTA - MULTI-ELEMENT PARAMETER CONTAINING THE AIRCRAFT MACH NUMBER FOR EACH TIME
- PT1 - MULTI-ELEMENT PARAMETER CONTAINING THE COMBUSTOR ENTRANCE TOTAL PRESSURE FOR EACH TIME
- TT1 - MULTI-ELEMENT PARAMETER CONTAINING THE COMBUSTOR ENTRANCE TOTAL TEMPERATURE FOR EACH TIME

ORIGINAL PAGE IS
OF POOR QUALITY

TT2 - MULTI-ELEMENT PARAMETER CONTAINING THE COMBUSTOR EXIT TOTAL TEMPERATURE FOR EACH TIME

CTA - MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT SPEED OF SOUND FOR EACH TIME

RHOTA - MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DENSITY FOR EACH TIME

DATA BASE STRUCTURES

FLI(FLIXXX) MULTI-ELEMENT MEMBER WITH FORMAT = 6RS,A4,2RS, WITH ONE RECORD PER SOURCE TIME CONTAINING THE FOLLOWING VALUES: (1) SOURCE TIME, (2) AIRCRAFT MACH NUMBER, (3) ENGINE POWER SETTING, (4) AMBIENT SPEED OF SOUND, (5) AMBIENT DENSITY, (6) AMBIENT DYNAMIC VISCOSITY, (7) LANDING GEAR POSITION, (8) FLAP SETTING AND (9) ABSOLUTE HUMIDITY. ONLY THE FIRST FIVE VALUES ARE USED.

ENG(CORE1) THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE CORE ENTRANCE FLOW STATE AS A FUNCTION OF (1) ENGINE POWER SETTING, (2) AIRCRAFT MACH NUMBER AND (3) ORDERED POSITION. THE FLOW STATE DEPENDENT VARIABLES IN ORDERED POSITION ARE : AREA, FUEL-TO-AIR RATIO, MASS FLOW RATE, TOTAL PRESSURE, TOTAL TEMPERATURE, AND ROTATIONAL SPEED.
(ONLY MASS FLOW RATE, TOTAL PRESSURE, AND TOTAL TEMPERATURE ARE REQUIRED)

ENG(CORE2) THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE CORE EXIT FLOW STATE AS A FUNCTION OF (1) ENGINE POWER SETTING, (2) AIRCRAFT MACH NUMBER AND (1) ORDERED POSITION. THE FLOW STATE DEPENDENT VARIABLES IN ORDERED POSITION ARE : AREA, FUEL-TO-AIR RATIO, MASS FLOW RATE, TOTAL PRESSURE, TOTAL TEMPERATURE AND ROTATIONAL SPEED.
(ONLY TOTAL TEMPERATURE IS REQUIRED)

ERRORS

NON-FATAL

1. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.
2. INSUFFICIENT LOCAL DYNAMIC STORAGE.
3. REQUIRED UNIT MEMBER NOT AVAILABLE.
4. UNABLE TO INTERPOLATE SPECIFIED TABLE.

FATAL - NONE

LDS REQUIREMENTS

(13 * NTIMES) + 1

WHERE

NTIMES - NUMBER OF SOURCE TIMES

ORIGINAL PAGE IS
OF POOR QUALITY

4.4.3 Turbine Noise Parameters Module (PRETUR)

PURPOSE - TO GENERATE THE PHYSICAL PARAMETERS REQUIRED FOR A TURBINE NOISE PREDICTION MODULE FOR TURBOJET AND TURBOFAN ENGINES

AUTHOR - CBF(L03/00/00)
MMF(L03/00/09)

INPUT — DEFAULT

USER PARAMETERS

IPRINT - PRINTED OUTPUT OPTION — 3
- 0 - NO PRINT DESIRED
- 1 - INPUT PRINT ONLY
- 2 - OUTPUT PRINT ONLY
- 3 - BOTH INPUT AND OUTPUT PRINT

IUNITS - INPUT UNIT OPTION — 2HSI
- 2HSI, SI UNITS
- 7HENGLISH, ENGLISH UNITS

ITYPE - TYPE OF INTERPOLATION TO BE USED ON THE INPUT TABLES — 1
- 1 - LINEAR INTERPOLATION
- 2 - CUBIC SPLINE INTERPOLATION

IRATIO - SPECIFIC HEATS OPTION — 8HCONSTANT
- 8HCONSTANT - CONSTANT SPECIFIC HEATS
- 8HVARIABLE - VARIABLE SPECIFIC HEATS

SCRXXX - THREE LETTER CODE XXX USED TO FORM THE UNIT MEMBER NAME FLI(FLIXXX). IF THE VALUE OF USER PARAMETER SCRXXX IS 3HABC, THE FLI UNIT MEMBER NAME IS FLI(FLIABC). — 3HXXX

MEMBERS AND TABLES - DESCRIBED UNDER DATA BASE STRUCTURES
- FLI(FLIXXX) - (SEE USER PARAMETER SCRXXX)
- ENG(TURBINE1)
- ENG(TURBINE2)

OUTPUT

SYSTEM PARAMETER

NERR - EXECUTIVE SYSTEM PARAMETER FOR ERROR ENCOUNTERED DURING EXECUTION OF A FUNCTIONAL MODULE
- =.TRUE., AN ERROR WAS ENCOUNTERED
- =.FALSE., NO ERRORS

USER PARAMETERS

NTIMES - NUMBER OF SOURCE TIME VALUES

TIMES - MULTI-ELEMENT PARAMETER CONTAINING THE SOURCE TIMES

MTA - MULTI-ELEMENT PARAMETER CONTAINING THE AIRCRAFT MACH NUMBER FOR EACH TIME

ORIGINAL PAGE IS
OF POOR QUALITY

Turbine Noise Parameters Module (PRETUR)

NT1 - MULTI-ELEMENT PARAMETER CONTAINING THE ROTATIONAL SPEED FOR EACH TIME

TT1 - MULTI-ELEMENT PARAMETER CONTAINING THE TURBINE ENTRANCE TOTAL TEMPERATURE FOR EACH TIME

TT2 - MULTI-ELEMENT PARAMETER CONTAINING THE TURBINE EXIT STATIC TEMPERATURE FOR EACH TIME

MT1 - MULTI-ELEMENT PARAMETER CONTAINING THE TURBINE MASS FLOW RATE FOR EACH TIME

FTA - MULTI-ELEMENT PARAMETER CONTAINING THE FUEL-TO-AIR RATIO FOR EACH TIME

CTA - MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT SPEED OF SOUND FOR EACH TIME

RHOTA - MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DENSITY FOR EACH TIME

HTA - MULTI-ELEMENT PARAMETER CONTAINING THE ABSOLUTE HUMIDITY FOR EACH TIME

DATA BASE STRUCTURES

FLI(FLIXXX) MULTI-ELEMENT MEMBER WITH FORMAT=6RS,A4,2RS, WITH ONE RECORD PER SOURCE TIME CONTAINING THE FOLLOWING VALUES : (1) SOURCE TIME, (2) AIRCRAFT MACH NUMBER, (3) ENGINE POWER SETTING, (4) AMBIENT SPEED OF SOUND, (5) AMBIENT DENSITY, (6) AMBIENT DYNAMIC VISCOSITY, (7) LANDING GEAR POSITION, (8) FLAP SETTING AND (9) ABSOLUTE HUMIDITY. ONLY THE VALUES OF THE FOLLOWING ENTRIES ARE USED: 1, 2, 3, 4, 5, AND 9

ENG(TURBINE1) THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE TURBINE ENTRANCE FLOW STATE AS A FUNCTION OF (1) ENGINE POWER SETTING, (2) AIRCRAFT MACH NUMBER AND (3) ORDERED POSITION. THE FLOW STATE DEPENDENT VARIABLES IN ORDERED POSITION ARE : AREA, FUEL-TO-AIR RATIO, MASS FLOW RATE, TOTAL PRESSURE, TOTAL TEMPERATURE, AND ROTATIONAL SPEED.
(ONLY MASS FLOW RATE, ROTATIONAL SPEED, AND TOTAL TEMPERATURE ARE REQUIRED)

ENG(TURBINE2) THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE TURBINE EXIT FLOW STATE AS A FUNCTION OF (1) ENGINE POWER SETTING, (2) AIRCRAFT MACH NUMBER AND (1) ORDERED POSITION. THE FLOW STATE DEPENDENT VARIABLES IN ORDERED POSITION ARE : AREA, FUEL-TO-AIR RATIO, MASS FLOW RATE, TOTAL PRESSURE, TOTAL TEMPERATURE, AND ROTATIONAL SPEED.
(ALL BUT ROTATIONAL SPEED ARE REQUIRED)

ORIGINAL PAGE IS
OF POOR QUALITY

Turbine Noise Parameters Module (PRETUR)

```
ERRORS
  NON-FATAL
    1.  MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.
    2.  INSUFFICIENT LOCAL DYNAMIC STORAGE.
    3.  REQUIRED UNIT MEMBER NOT AVAILABLE.
    4.  UNABLE TO INTERPOLATE SPECIFIED TABLE.
  FATAL - NONE

LDS REQUIREMENTS
  LENGTH = ( NTIMES * ( NWORDS + ( NP - 1 ) ) + 1 )
    WHERE
      NTIMES = NUMBER OF SOURCE TIMES
      NWORDS = NUMBER OF WORDS READ PER RECORD ON MEMBER
               CONTAINING ENGINE VARIABLE DATA
      NP     = NUMBER OF USER PARAMETERS OUTPUT FROM THIS
               MODULE

GDS REQUIREMENTS
  ALLOCATION REQUIRED FOR THE FOLLOWING TABLES
    1.  ENG(TURBINE1)
    2.  ENG(TURBINE2)
```

ORIGINAL PAGE IS
OF POOR QUALITY

4.4.4 Jet Noise Parameters Module (PREJET)

```
PURPOSE - GENERATE THE PHYSICAL PARAMETERS REQUIRED FOR THE
          EXHAUST JET NOISE PREDICTION MODULES FOR TURBOJET AND
          TURBOFAN ENGINES

AUTHOR  - CBF(L03/00/00)
          MMF(L03/00/09)

INPUT                                                      DEFAULT
  USER PARAMETERS
    AE     - ENGINE REFERENCE AREA, M**2 (FT**2)           PI/4.
    AP     - PRIMARY NOZZLE PLUG AREA, RE AE               0.
    SCRXXX - THREE LETTER CODE XXX USED TO FORM THE        3HXXX
             UNIT MEMBER NAME FLI(FLIXXX).  IF THE
             VALUE OF THE SCRXXX USER PARAMETER
             IS 3HABC, THE FLI UNIT MEMBER NAME IS
             FLI(FLIABC).
    IRATIO - SPECIFIC HEATS OPTION                       8HCONSTANT
             8HCONSTANT - CONSTANT RATIO OF SPECIFIC
                          HEATS
             8HVARIABLE - VARIABLE RATIO OF SPECIFIC
                          HEATS
    ITYPE  - TYPE OF INTERPOLATION TO BE USED ON THE       1
             INPUT TABLE(S)
             1 - LINEAR INTERPOLATION
             2 - CUBIC SPLINE INTERPOLATION
    IUNITS - INPUT UNIT OPTION                             2HSI
             2HSI,      SI UNITS
             7HENGLISH, ENGLISH UNITS
    IPRINT - PRINTED OUTPUT OPTION                         3
             0 - NO PRINT DESIRED
             1 - INPUT PRINT ONLY
             2 - OUTPUT PRINT ONLY
             3 - BOTH INPUT AND OUTPUT PRINT

  MEMBERS AND TABLES - DESCRIBED UNDER DATA BASE STRUCTURES
      FLI(FLIXXX)
      ENG(PRIM)
      ENG(SEC)

OUTPUT
  SYSTEM PARAMETER
    NERR   EXECUTIVE SYSTEM PARAMETER FOR ERROR ENCOUNTERED
           DURING EXECUTION OF A FUNCTIONAL MODULE.

  USER PARAMETERS
    NTIMES - NUMBER OF SOURCE TIME VALUES
    TIMES  - MULTI-ELEMENT PARAMETER CONTAINING THE SOURCE
             TIMES
```

Jet Noise Parameters Module (PREJET)

AT1 - MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET AREA FOR EACH TIME
MTA - MULTI-ELEMENT PARAMETER CONTAINING THE AIRCRAFT MACH NUMBER FOR EACH TIME
DET1 - MULTI-ELEMENT PARAMETER CONTAINING THE ACTUAL PRIMARY JET EQUIVALENT DIAMETER FOR EACH TIME
DHT1 - MULTI-ELEMENT PARAMETER CONTAINING THE ACTUAL PRIMARY JET HYDRAULIC DIAMETER FOR EACH TIME
MT1 - MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET MACH NUMBER FOR EACH TIME
TT1 - MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET TOTAL TEMPERATURE FOR EACH TIME
VT1 - MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET VELOCITY FOR EACH TIME
RSHT1 - MULTI-ELEMENT PARAMETER CONTAINING THE RATIO OF SPECIFIC HEATS FOR THE PRIMARY JET FOR EACH TIME
RHOT1 - MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET DENSITY FOR EACH TIME
CTA - MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT SPEED OF SOUND FOR EACH TIME
RHOTA - MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DENSITY FOR EACH TIME
(THE FOLLOWING USER PARAMETERS ARE COMPUTED ONLY FOR A DOUBLE STREAM TURBOFAN ENGINE.)
AT2 - MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET AREA FOR EACH TIME
MT2 - MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET MACH NUMBER FOR EACH TIME
TT2 - MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET TOTAL TEMPERATURE FOR EACH TIME
VT2 - MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET VELOCITY FOR EACH TIME
RHOT2 - MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET DENSITY FOR EACH TIME
RSHT2 - MULTI-ELEMENT PARAMETER CONTAINING THE RATIO OF SPECIFIC HEATS FOR THE SECONDARY JET FOR EACH TIME
DET2 - MULTI-ELEMENT PARAMETER CONTAINING THE ACTUAL SECONDARY JET EQUIVALENT DIAMETER FOR EACH TIME
DHT2 - MULTI-ELEMENT PARAMETER CONTAINING THE ACTUAL SECONDARY JET HYDRAULIC DIAMETER FOR EACH TIME

DATA BASE STRUCTURES
FLI(FLIXXX) MULTI-ELEMENT MEMBER WITH FORMAT - 6RS,A4,2RS, WITH ONE RECORD PER SOURCE TIME CONTAINING THE FOLLOWING VALUES: (1) SOURCE TIME, (2) AIRCRAFT MACH NUMBER, (3) ENGINE POWER SETTING, (4) AMBIENT SPEED OF SOUND, (5) AMBIENT DENSITY, (6) AMBIENT DYNAMIC

ORIGINAL PAGE IS
OF POOR QUALITY

	VISCOSITY, (7) LANDING GEAR POSITION, (8) FLAP SETTING, AND (9) ABSOLUTE HUMIDITY. ONLY THE FOLLOWING VALUES ARE USED : (1) TIME, (2) AIRCRAFT MACH NUMBER, (3) ENGINE POWER SETTING, (4) AMBIENT SPEED OF SOUND, (5) AMBIENT DENSITY, AND (9) ABSOLUTE HUMIDITY.
ENG(PRIM)	THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE PRIMARY NOZZLE FLOW STATE AS A FUNCTION OF (1)ENGINE POWER SETTING, (2)AIRCRAFT MACH NUMBER, AND (3) ORDERED POSITION. THE FLOW STATE DEPENDENT VARIABLES IN ORDERED POSITION ARE : AREA, FUEL-TO-AIR RATIO, MASS FLOW RATE, TOTAL PRESSURE, TOTAL TEMPERATURE, AND ROTATIONAL SPEED. (ROTATIONAL SPEED IS NOT REQUIRED)
ENG(SEC)	THREE-DIMENSIONAL TYPE ONE DATA TABLE CONTAINING THE SECONDARY NOZZLE FLOW STATE AS A FUNCTION OF (1)ENGINE POWER SETTING, (2)AIRCRAFT MACH NUMBER, AND (3)ORDERED POSITION. THE FLOW STATE DEPENDENT VARIABLES IN ORDERED POSITION ARE : AREA, FUEL-TO-AIR RATIO, MASS FLOW RATE, TOTAL PRESSURE, TOTAL TEMPERATURE, AND ROTATIONAL SPEED. (ROTATIONAL SPEED IS NOT REQUIRED)

ERRORS

NON-FATAL

1. REQUIRED UNIT MEMBER NOT AVAILABLE.
2. INSUFFICIENT LOCAL DYNAMIC STORAGE.
3. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.
4. UNABLE TO INTERPOLATE SPECIFIED TABLE.
5. SPECIFIED USER PARAMETER OUT OF RANGE. DEFAULT VALUE WILL BE USED.

FATAL - NONE

LDS REQUIREMENTS

LENGTH = NTIMES * (21 + NWORDS) + 1

WHERE

NTIMES - NUMBER OF SOURCE TIMES

NWORDS - NUMBER OF WORDS PER RECORD ON MEMBER CONTAINING ENGINE VARIABLE DATA

GDS REQUIREMENTS

ALLOCATION REQUIRED FOR THE FOLLOWING TABLE(S)

1. ENG(PRIM)
2. ENG(SEC) (IF COMPUTING FOR A DUAL STREAM JET)

ORIGINAL PAGE IS
OF POOR QUALITY

4.4.5 Airframe Noise Parameters Module (PREAFM)

```
PURPOSE - TO GENERATE THE PARAMETERS REQUIRED TO
          EXECUTE THE AIRFRAME NOISE MODULE

AUTHOR  - CBF(L03/00/00)
          MMF(L03/00/09)

INPUT                                                    DEFAULT
  USER PARAMETERS
    IPRINT - PRINTED OUTPUT OPTION                          3
             0 - NO PRINT DESIRED
             1 - INPUT PRINT ONLY
             2 - OUTPUT PRINT ONLY
             3 - BOTH INPUT AND OUTPUT PRINT
    IUNITS - INPUT UNIT OPTION                             2HSI
             2HSI,      SI UNITS
             7HENGLISH, ENGLISH UNITS
    SCRXXX - THREE LETTER CODE XXX USED TO FORM THE        3HXXX
             UNIT MEMBER NAME FLI(FLIXXX).  IF THE
             VALUE OF USER PARAMETER, SCRXXX
             IS 3HABC, THE FLI UNIT MEMBER NAME IS
             FLI(FLIABC).

  MEMBERS AND TABLES - DESCRIBED UNDER DATA BASE STRUCTURES
    FLI(FLIXXX) - (SEE USER PARAMETER SCRXXX)

OUTPUT
  SYSTEM PARAMETER
    NERR    - EXECUTIVE SYSTEM PARAMETER FOR ERROR ENCOUNTERED
              DURING EXECUTION OF A FUNCTIONAL MODULE
              =.TRUE.,  ERROR ENCOUNTERED
              =.FALSE., NO ERRORS ENCOUNTERED

  USER PARAMETERS
    NTIMES - NUMBER OF SOURCE TIME VALUES
    TIMES  - MULTI-ELEMENT PARAMETER CONTAINING THE SOURCE
             TIMES
    LANDTG - MULTI-ELEMENT PARAMETER CONTAINING THE
             LANDING GEAR POSITION FOR EACH TIME
             ( 4HDOWN OR 4HUP   )
    MTA    - MULTI-ELEMENT PARAMETER CONTAINING THE AIRCRAFT
             MACH NUMBER FOR EACH TIME
    DELTATF- MULTI-ELEMENT PARAMETER CONTAINING THE
             FLAP SETTING, IN DEGREES, FOR EACH TIME
    CTA    - MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT
             SPEED OF SOUND FOR EACH TIME
    RHOTA  - MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT
             DENSITY FOR EACH TIME
```

ORIGINAL PAGE IS
OF POOR QUALITY

Airframe Noise Parameters Module (PREAFM)

MUTA - MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DYNAMIC VISCOSITY FOR EACH TIME

DATA BASE STRUCTURES

FLI(FLIXXX) MULTI=ELEMENT MEMBER WITH FORMAT=6RS,A4,2RS, WITH ONE RECORD PER SOURCE TIME CONTAINING THE FOLLOWING VALUES : (1) SOURCE TIME, (2) AIRCRAFT MACH NUMBER, (3) ENGINE POWER SETTING, (4) AMBIENT SPEED OF SOUND, (5) AMBIENT DENSITY, (6) AMBIENT DYNAMIC VISCOSITY, (7) LANDING GEAR POSITION, (8) FLAP SETTING, AND (9) ABSOLUTE HUMIDITY. THE FOLLOWING WORDS ARE USED IN THIS MODULE : (1) SOURCE TIME, (2) AIRCRAFT MACH NUMBER, (4) AMBIENT SPEED OF SOUND, (5) AMBIENT DENSITY, (6) AMBIENT DYNAMIC VISCOSITY, (7) LANDING GEAR POSITION, AND (8) FLAP SETTING, IN DEGREES.

ERRORS

NON-FATAL

1. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.
2. INSUFFICIENT LOCAL DYNAMIC STORAGE.

FATAL - NONE

LDS REQUIREMENTS

LENGTH = (NTIMES * (NWORDS + (NP - 1)) + 1)

WHERE

NTIMES = NUMBER OF SOURCE TIMES

NWORDS = NUMBER OF WORDS PER RECORD USED FROM MEMBER CONTAINING ENGINE VARIABLE DATA

NP = NUMBER OF USER PARAMETERS OUTPUT FROM THIS MODULE

ORIGINAL PAGE IS
OF POOR QUALITY

4.5 PROPAGATION

4.5.1 Propagation Module (PRO)

PURPOSE - PRO IS THE ENTRY POINT FOR THE PROPAGATION MODULE. PRO TAKES NOISE DATA WHICH IS GENERATED BY THE NOISE MODULES IN THE SOURCE FRAME OF REFERENCE AND APPLIES ALL OF THE APPROPIATE COMPUTATIONS TO TRANSFER IT TO THE OBSERVER FRAME OF REFERENCE.

AUTHOR - WKB(L03/00/00)
WKB(L03/00/04)
CBF(L03/00/05)
MMF(L03/00/09)

INPUT
USER PARAMETERS

IPRINT	OUTPUT PRINT OPTION CODE (INTEGER) =0 NO PRINTED OUTPUT =1 PRINT INPUT DATA ONLY =2 PRINT OUTPUT DATA ONLY =3 PRINT BOTH INPUT AND OUTPUT DATA (DEFAULT)
IOUT	=1 PRINT OUTPUT AS SOUND PRESSURE LEVEL, SPL, IN DECIBELS (INTEGER) =2 PRINT OUTPUT AS DIMENSIONLESS MEAN-SQUARE PRESSURE (DEFAULT) =3 PRINT OUTPUT IN BOTH FORMS
SIGMA	SPECIFIC FLOW RESISTANCE OF THE GROUND, KG/(S M**3) OR (SLUG/(S FT**3)) (REAL SG) (DEFAULT IS 2.5 * 10**5 KG/(S M**3))
IUNITS	=2HSI , INPUTS ARE IN SI UNITS (DEFAULT) =7HENGLISH, INPUTS ARE IN ENGLISH UNITS
NBAND	NUMBER OF SUBBANDS PER ONE-THIRD OCTAVE BAND (INTEGER) (DEFAULT IS 5) (NOTE, NBAND MUST BE ODD - E.G., 1,3,5,...)
RS	SOURCE RADIUS, M (FT) (REAL SINGLE) (DEFAULT IS 1.0 M) USER SHOULD NOTE THAT RS VALUE MUST BE THE SAME USED BY THE NOISE MODULES IN GENERATING THE NOISE TABLES.
SURFACE	TYPE OF SURFACE TO BE USED IN CALCULATING GROUND EFFECTS =4HSOFT, NON HARD GROUND SURFACE (DEFAULT) =4HHARD, HARD GROUND SURFACE
COH	INCOHERENCE COEFFICIENT - DEFAULT IS .01
ABSORP	=.TRUE. INCLUDE THE EFFECTS OF ATMOSPHERIC ABSORPTION =.FALSE. DO NOT INCLUDE THE EFFECTS OF ATMOSPHERIC ABSORPTION (DEFAULT)
GROUND	=.TRUE. INCLUDE GROUND EFFECTS =.FALSE. DO NOT INCLUDE GROUND EFFECTS (DEFAULT)

ORIGINAL PAGE IS
OF POOR QUALITY

Propagation Module (PRO)

PROTIME THREE LETTER ID (XXX) TO APPEND TO 3HFLI TO DEFINE THE UNIT MEMBER FLI(FLIXXX) THAT WAS GENERATED BY THE FLIGHT MODULE CONTAINING THE SOURCE TIMES TO BE USED BY THE NOISE MODULES. THIS ID ALSO BECOMES THE FIRST THREE CHARACTERS OF EACH TABLE (ONE TABLE FOR EACH SOURCE TIME) MEMBER THAT IS GENERATED BY A NOISE MODULE. FOR EXAMPLE, IF THE VALUE OF PARAMETER PROTIME IS 3HF01, THEN THE SOURCE TIMES ARE WRITTEN ON UNIT MEMBER FLI(FLIF01) BY THE FLIGHT MODULE. IF THERE ARE 20 SOURCE TIMES ON UNIT MEMBER FLI(FLIF01), THEN THERE ARE 20 TABLES GENERATED FOR EACH NOISE MODULE CALLED AND THE MEMBER NAMES FOR THESE TABLES ARE THE SAME FOR ALL NOISE SOURCES WITH ONLY THE UNIT NAME BEING DIFFERENT. FOR THIS EXAMPLE, IF NOISE MODULES FAN AND CORE ARE CALLED, THE TABLES ARE ON UNIT MEMBERS FAN(F01001), FAN(F01002), ... FAN(F01020), CORE(F01001), CORE(F01002),.. CORE(F01020). DEFAULT IS 3HXXX.

PROSUM CONTAINS THE NAMES OF SOURCE UNITS THAT ARE TO BE SUMMED BEFORE PROPAGATION. IF THE NOISE DATA FROM THE CORE AND FAN MODULES ARE TO BE SUMMED, THEN PROSUM WILL CONTAIN TWO ELEMENTS - 4HCORE AND 4HFAN .

DATA BASE UNITS AND MEMBERS

ATM(TMOD) TYPE 1 TABLE CONTAINING ATMOSPHERIC PROPERTIES (SEE MODULE ATM)

INDEPENDENT VARIABLES
1. ALTITUDE
2. ORDERED POSITION

DEPENDENT VARIABLES IN THE FOLLOWING ORDERED POSITION
1. PRESSURE
2. DENSITY
3. TEMPERATURE
4. SPEED OF SOUND
5. AVERAGE SPEED OF SOUND
6. HUMIDITY
7. COEFFICIENT OF VISCOSITY
8. COEFFICIENT OF THERMAL CONDUCTIVITY
9. CHARACTERISTIC IMPEDANCE (RHO*C)

ATM(AAC) TYPE 1 TABLE CONTAINING ATMOSPHERIC ABSORPTION COEFFICIENTS (SEE MODULE ABS)

INDEPENDENT VARIABLES
1. ALTITUDE
2. FREQUENCY

ORIGINAL PAGE IS
OF POOR QUALITY

Propagation Module (PRO)

GEO(GEOM) — GEOMETRY DATA FOR ALL OBSERVERS RELATIVE TO ONE NOISE SOURCE COORDINATE SYSTEM
SEE DESCRIPTION IN DATA BASE STRUCTURES.
(SEE MODULE GEO)

FLI(FLIXXX) — FLIGHT DATA CONTAINING SOURCE TIMES USED BY THE NOISE MODULES. THE XXX IS REPLACED BY THE ID FOUND IN PARAMETER PROTIME.
(SEE DESCRIPTION IN DATA BASE STRUCTURES)

YYYYYY(XXXNNN) — TYPE 1 TABLE CONTAINING DIMENSIONLESS MEAN SOUARE PRESSURE. THE YYYYYY IS THE UNIT NAME ASSOCIATED WITH A NOISE MODULE (E.G., FAN OR CORE). THE XXX IS THE ID FOUND IN PARAMETER PROTIME. THE NNN IS A COUNTER STARTING AT 001 AND CONTINUING FOR THE NUMBER OF SOURCE TIMES. THERE IS ONE TABLE PER SOURCE TIME. IF THERE ARE TWENTY SOURCE TIMES, THEN THERE ARE TWENTY TABLES FOR EACH NOISE MODULE.
INDEPENDENT VARIABLES
1. FREQUENCY, HZ
2. POLAR DIRECTIVITY ANGLE, DEG
3. AZIMUTHAL DIRECTIVITY ANGLE, DEG

OUTPUT

USER PARAMETERS

NERR — =.TRUE. , ERROR ENCOUNTERED, PRO TERMINATED ABNORMALLY
=.FALSE., NO ERRORS ENCOUNTERED, PRO TERMINATED SUCCESSFULLY

DATA BASE UNITS AND MEMBERS

PRO(PRES) — DIMENSIONLESS MEAN SQUARE PRESSURE AT THE OBSERVER AS A FUNCTION OF FREQUENCY AND TIME. (SEE DESCRIPTION IN DATA BASE STRUCTURES.)

SCRATCH(XXXNNN) — UNIT SCRATCH CONTAINS THE RESULT OF SUMMING NOISE TABLES

DATA BASE STRUCTURES

THE FORMAT OF GEO(GEOM) IS AS FOLLOWS:

RECORD	WORD	DESCRIPTION
1		RECORD FORMAT IS I,3RS,I,RS
	1	OBSERVER INDEX FOR FIRST OBSERVER
	2	X COORDINATE OF OBSERVER
	3	Y COORDINATE OF OBSERVER
	4	Z COORDINATE OF OBSERVER
	5	NUMBER OF RECEPTION TIMES ASSOCIATED WITH THIS OBSERVER (ASSUME VALUE IS N)
	6	OBSERVER'S HEIGHT

ORIGINAL PAGE IS
OF POOR QUALITY

Propagation Module (PRO)

```
2          RECORD FORMAT IS *RS
      1
      .    RECEPTION TIMES FOR CURRENT OBSERVER
      .    INDEX
      N
```

RECORDS 3 THROUGH N+2 CONTAIN GEOMETRY DATA FOR EACH RECEPTION TIME. RECORD 3 CONTAINS GEOMETRY DATA FOR THE FIRST RECEPTION TIME, RECORD 4 FOR THE SECOND RECEPTION TIME,... RECORD N+2 FOR THE N TH RECEPTION TIME.

```
3          RECORD FORMAT IS *RS
      1    DISTANCE OF SOURCE FROM OBSERVER
      2    EMISSION TIME, SEC
      3    DIRECTIVITY ANGLE, DEG
      4    ELEVATION ANGLE, DEG
      5    AZIMUTH ANGLE, DEG

4     REPEAT OF RECORD 3 FOR SECOND RECEPTION TIME
.     .
.     .
N+3        RECORD FORMAT IS I,3RS,I,RS
      1    OBSERVER INDEX FOR SECOND OBSERVER
      2    X COORDINATE OF OBSERVER
      3    Y COORDINATE OF OBSERVER
      4    Z COORDINATE OF OBSERVER
      5    NUMBER OF RECEPTION TIMES ASSOCIATED WITH
           THIS OBSERVER (ASSUME VALUE IS M)

N+4        RECORD FORMAT IS *RS
      1
      .    RECEPTION TIMES FOR CURRENT OBSERVER
      .    INDEX
      M
```

RECORD N+5 THROUGH RECORD N+M+4 CONTAIN GEOMETRY DATA FOR EACH RECEPTION TIME STORED IN THE SAME MANNER AS DESCRIBED ABOVE IN RECORDS 3 THROUGH N+2.

THE PATTERN AS SEEN IN RECORDS 1 THROUGH N+2 AND RECORDS N+3 THROUGH N+M+4 CONTINUES FOR ALL OBSERVERS

Propagation Module (PRO)

THE FORMAT OF FLI(FLIXXX) IS AS FOLLOWS:

RECORD	WORD	DESCRIPTION
1		RECORD FORMAT IS *RS
	1	VALUE OF FIRST SOURCE TIME
	2	MACH NUMBER
	3	POWER SETTING
	4	AMBIENT SPEED OF SOUND
	5	DENSITY
	6	VISCOSITY
	7	LANDING GEAR INDICATOR
	8	FLAP SETTING
	9	HUMIDITY
2		RECORD FORMAT IS *RS
	.	SAME INFORMATION AS IN RECORD 1 EXCEPT FOR THE SECOND SOURCE TIME
3	.	
.	.	
.	.	
.	.	

THE FORMAT OF PRO(PRES) IS AS FOLLOWS:

RECORD	WORD	DESCRIPTION
1		RECORD FORMAT IS I,*A8
	1	NUMBER OF NOISE SOURCES PROPAGATED TO THE OBSERVERS, NS.
	2-(NS+1)	MODULE NAMES OF NOISE SOURCES PROPAGATED TO THE OBSERVERS
2		RECORD FORMAT IS 2I,2RS
	1	OBSERVER INDEX FOR THE FIRST OBSERVER
	2	NUMBER OF RECEPTION TIMES ASSOCIATED WITH THIS OBSERVER (ASSUME VALUE IS N)
	3	AIR DENSITY AT THE OBSERVER (RE RHO_R)
	4	SPEED OF SOUND AT THE OBSERVER (RE C_R)
3		RECORD FORMAT IS *RS
	1	
	.	RECEPTION TIMES FOR CURRENT OBSERVER INDEX
	.	
	N	

```
4              RECORD FORMAT IS *RS
          1    DIMENSIONLESS MEAN SQUARE PRESSURE FOR
               THE FIRST FREQUENCY AND THE FIRST
               RECEPTION TIME
          2    DIMENSIONLESS MEAN SQUARE PRESSURE FOR
               THE SECOND FREQUENCY AND THE FIRST
               RECEPTION TIME
          .
          .
         NF    DIMENSIONLESS MEAN SQUARE PRESSURE FOR
               THE LAST FREQUENCY AND THE FIRST
               RECEPTION TIME

5              RECORD FORMAT IS *RS
          1    DIMENSIONLESS MEAN SQUARE PRESSURE FOR
          .    ALL FREQUENCIES FOR THE SECOND
          .    RECEPTION TIME
         NF

6              RECORD FORMAT IS *RS
          1    DIMENSIONLESS MEAN SQUARE PRESSURE FOR
          .    ALL FREQUENCIES FOR THE THIRD
          .    RECEPTION TIME
         NF
.
.
N+4       SAME AS RECORD 2 BUT DATA IS FOR SECOND
          OBSERVER

N+5       SAME AS RECORD 3 BUT DATA IS FOR SECOND
          OBSERVER
```

RECORDS 2 THROUGH N+3 REPEAT FOR ALL OBSERVERS. THE VALUE OF N DIFFERS FOR EACH OBSERVER.

ERRORS
NON-FATAL

FUNCTIONAL MODULE ERRORS
1. REQUIRED UNIT MEMBER NOT AVAILABLE
2. INSUFFICIENT LDS DYNAMIC STORAGE
3. UNIT MEMBER NOT OF CORRECT FORMAT
4. MEMBER MANAGER ERROR OCCURRED ON READING OR OPENING A UNIT MEMBER
5. REQUIRED USER PARAMETER PROSUM IS TYPE --- WITH --- - EXPECTED TYPE ALPHA AND THE NUMBER OF ELEMENTS .LE. 10.

6. UNABLE TO INTERPOLATE TABLE ---- FOR VALUES -, -, -, -
7. ERROR ENCOUNTERED IN BUILDING TABLE

PRO MODULE ERRORS

1. TABLES OF NOISE SOURCES TO BE SUMMED ARE NOT COMPATIBLE

FATAL - NONE

LDS REQUIREMENTS

LENGTH = 3*NFREQ + NB + 6*NB*NFREQ

WHERE NFREQ = NUMBER OF FREQUENCIES
NB = VALUE OF USER PARAMETER "NBAND"

GDS REQUIREMENTS

ALLOCATION REQUIRED FOR THE FOLLOWING TABLES:

1. ATM(TMOD)
2. ATM(AAC) (IF USER PARAMETER "ABSORP" IS .TRUE.)
3. NUMBER OF YYYYYY(XXXNNN) NOISE TABLES THAT ARE TO BE SUMMED PLUS ONE

4.5.2 General Suppression Module (GENSUP)

```
PURPOSE - TO APPLY A NOISE SUPPRESSION FACTOR TO A NOISE TABLE
          PRODUCED BY ANY ANOPP NOISE SOURCE MODULE

AUTHOR  - CBF(L03/00/00)
          MMF(L03/00/09)

INPUT
  USER PARAMETERS                                             DEFAULT
    SCRXXX - THREE LETTER CODE, XXX, USED TO FORM              3HXXX
             OUTPUT TABLE MEMBER NAME, NOISES (XXXNNN)
    SCRNNN - INTEGER VALUE, NNN, .GT. 0 USED TO FORM
             OUTPUT TABLE MEMBER NAME, NOISES (XXXNNN)          001
    IPRINT - PRINT FLAG (I)                                       3
             =0, NO PRINT DESIRED
             =1, INPUT PRINT ONLY
             =2, OUTPUT PRINT ONLY
             =3, BOTH INPUT AND OUTPUT PRINT
    RHOA   - AMBIENT DENSITY (RS), KG/M**3                    1.225
             (SLUG/FT**3)
    CA     - AMBIENT SPEED OF SOUND (RS), M/S               340.294
             (FT/SEC)
    IOUT   - OUTPUT CODE ( FOR TABLE AND/OR                       3
             PRINTED OUTPUT )
              0   NO PRINT, BUT GENERATE TABLE
                  NOISES(XXXNNN)
             -1   PRINT OUTPUT IN DB UNITS, BUT DO
                  NOT GENERATE TABLE NOISES(XXXNNN)
             -2   PRINT OUTPUT IN DIMENSIONLESS UNITS,
                  BUT DO NOT GENERATE TABLE
                  NOISES(XXXNNN)
             -3   BOTH OPTIONS -1 AND -2
              1   PRINT OUTPUT IN DB UNITS AND
                  GENERATE TABLE NOISES(XXXNNN)
              2   PRINT OUTPUT IN DIMENSIONLESS UNITS
                  AND GENERATE TABLE NOISES(XXXNNN)
              3   BOTH OPTIONS 1 AND 2
    IUNITS - INPUT UNITS FLAG                                    SI
             2HSI,      SI UNITS
             7HENGLISH, ENGLISH UNITS

  DATA BASE UNIT MEMBERS
    (DESCRIBED UNDER DATA BASE STRUCTURES)
    NOISE   (XXXNNN) NOTE MEMBER NAME XXXNNN IS FORMED FROM
                     USER PARAMETERS SCRXXX AND SCRNNN.
    SUPPRESS(FACTOR)
```

```
OUTPUT
  SYSTEM PARAMETERS
    NERR    - ERROR FLAG
             .TRUE.,  IMPLIES AN ERROR WAS ENCOUNTERED
                      DURING MODULE EXECUTION
             .FALSE., NO ERROR ENCOUNTERED

  DATA BASE UNIT MEMBERS
    NOISES (XXXNNN) SEE FORMAT UNDER DATA BASE STRUCTURES.
                    NOTE MEMBER NAME XXXNNN IS FORMED FROM
                    USER PARAMETERS SCRXXX AND SCRNNN.

DATA BASE STRUCTURES
  NOISE    (XXXNNN)  TYPE 1 TABLE CONTAINING MEAN SQUARE
                     ACOUSTIC PRESSURE AS A FUNCTION OF
                     (1) FREQUENCY, (2) DIRECTIVITY ANGLE
                     AND (3) AZIMUTHAL ANGLE.  SEE INFORMATION
                     ON OVERRIDING THIS UNIT NAME UNDER
                     DESCRIPTION OF NOISES(XXXNNN).
  SUPPRESS(FACTOR)   TYPE 1 DATA TABLE CONTAINING A SUPPRESSION
                     FACTOR AS A FUNCTION OF (1) FREQUENCY,
                     (2) DIRECTIVITY ANGLE AND (3) AZIMUTHAL
                     ANGLE
  NOISES   (XXXNNN)  TYPE 1 TABLE CONTAINING SUPPRESSED MEAN
                     SQUARE ACOUSTIC PRESSURE AS A FUNCTION
                     OF (1) FREQUENCY, (2) DIRECTIVITY ANGLE,
                     AND (3) AZIMUTH ANGLE.  WHEN OVERRIDING
                     THIS UNIT NAME, THE NAME OF THE UNIT MUST
                     BE THE SAME AS THE NAME OF THE UNIT CONTAIN-
                     ING THE INPUT NOISE TABLE WITH AN 'S' ON
                     THE END. EXAMPLE (1) : SUPPOSE THE UNIT
                     CONTAINING THE INPUT NOISE TABLE IS 'DATE'.
                     THE NAME OF THE UNIT CONTAINING THE OUTPUT
                     NOISE TABLE WILL BECOME 'DATES'.  EXAMPLE
                     (2): SUPPOSE THE UNIT CONTAINING THE OUTPUT
                     NOISE TABLE IS OVERRIDEN TO BECOME 'STORES'
                     THEN THE UNIT CONTAINING THE INPUT NOISE
                     TABLE MUST BE OVERRIDEN TO BE 'STORE'.
                     EXAMPLE (3) : IF THE INPUT UNIT IS TO BE
                     'TABLE', THEN TO OVERRIDE THIS OUTPUT UNIT
                     TO BE SOMETHING OTHER THAN 'TABLES', THE
                     FOLLOWING MUST BE DONE.  FOR EXAMPLE, THE
                     INPUT UNIT IS 'TABLE'; THE OUTPUT UNIT IS
                     TO BE 'STORE'.  OVERRIDE 'NOISE' TO BE
                     'TABLE' AND OVERRIDE 'TABLES' TO BE 'STORE'
```

ERRORS
  NON-FATAL
    1. INSUFFICIENT DYNAMIC STORAGE.
    2. INTERPOLATION ERROR ON INPUT NOISE TABLE.
    3. INVALID VALUE FOR INPUT USER PARAMETER. DEFAULT
       VALUE WILL BE USED.
    4. ERROR IN BUILDING OUTPUT TABLE OF SUPPRESSED NOISE.
  FATAL - NONE

LDS REQUIREMENTS
```
  LENGTH =  ( 2 * NFREQ ) + NTHETA + NPHI + NINT +
            ( NTHETA * NPHI ) * ( 1 + NFREQ )
      WHERE :
        NFREQ  = NUMBER OF FREQUENCY VALUES ON INPUT AND OUTPUT
                 NOISE TABLES
        NTHETA = NUMBER OF DIRECTIVITY ANGLE VALUES ON INPUT
                 AND OUTPUT NOISE TABLES
        NPHI   = NUMBER OF AZIMUTH ANGLE VALUES ON INPUT AND
                 OUTPUT NOISE TABLES
        NINT   = NUMBER OF VALID INTERPOLATION PROCEDURES FOR
                 THE INPUT AND OUTPUT NOISE TABLES
```

GDS REQUIREMENTS
  SUFFICIENT ALLOCATION FOR THE FOLLOWING TABLES
    1. NOISE   (XXXNNN)
    2. SUPPRESS(FACTOR)
    3. NOISES   (XXXNNN)

4.6 RECEIVED NOISE

4.6.1 Noise Levels Module (LEV)

PURPOSE - TO COMPUTE OVERALL SOUND PRESSURE LEVEL, A-WEIGHTED SOUND PRESSURE LEVEL, D-WEIGHTED SOUND PRESSURE LEVEL, PERCEIVED NOISE LEVEL, AND TONE CORRECTED PERCEIVED NOISE LEVEL AS FUNCTIONS OF TIME AND OBSERVER

AUTHOR - CBF(L03/00/00)
CBF(L03/00/04)
MMF(L03/00/09)

```
INPUT
  USER PARAMETERS                                              DEFAULT
    IAWT   - A-WEIGHTED SOUND PRESSURE LEVEL OPTION               F
    IDWT   - D-WEIGHTED SOUND PRESSURE LEVEL OPTION               F
    IOSPL  - OVERALL SOUND PRESSURE LEVEL OPTION                  F
    IPNL   - PERCEIVED NOISE LEVEL OPTION                         F
    IPNLT  - TONE-CORRECTED PERCEIVED NOISE LEVEL OPTION          T
    ( THE ABOVE OPTION CODES HAVE THE FOLLOWING VALUES )
    (        .TRUE.  - COMPUTE                         )
    (        .FALSE. - DO NOT COMPUTE                  )
    IUNITS - INPUT UNIT OPTION                                  2HSI
             =2HSI,      SI UNITS
             =7HENGLISH, ENGLISH UNITS
    IPRINT - PRINTED OUTPUT OPTION                                 3
             0 - NO PRINT DESIRED
             1 - INPUT PRINT ONLY
             2 - OUTPUT PRINT ONLY
             3 - BOTH INPUT AND OUTPUT PRINT
    MEMSUM - CONTAINS THE UNIT NAME AND MEMBER NAME OF THE
             NOISE MEMBERS TO BE SUMMED ( CONTAINS .LE. 20
             ELEMENTS, IE. .LE. 10 NOISE MEMBERS )

  MEMBERS - DESCRIBED UNDER DATA BASE STRUCTURES
  NOTE : MEMBERS ARE SPECIFIED BY UNIT(MEMBER) NAME
      SFIELD(FREQ)
      OBSERV(COORD)
      XXXXXX(YYYYYY)

OUTPUT
  SYSTEM PARAMETER
    NERR     EXECUTIVE SYSTEM PARAMETER FOR ERROR ENCOUNTERED
             DURING EXECUTION OF A FUNCTIONAL MODULE. NERR SET
             TO .TRUE. IF ERROR ENCOUNTERED
    MEMBERS
      LEV(PNLT)
```

Noise Levels Module (LEV)

DATA BASE STRUCTURES

OBSERV(COORD)	MULTI-RECORD MEMBER WITH FORMAT = 3RS, CONTAINING ONE RECORD FOR EACH OBSERVER WITH VALUES OF THE X,Y,Z COORDINATES
SFIELD(FREQ)	1 RECORD MEMBER IN *RS FORMAT CONTAINING REAL SINGLE VALUES OF FREQUENCIES IN HZ
XXXXXX(YYYYYY)	XXXXXX IS THE UNIT NAME AND YYYYYY IS THE MEMBER NAME OF AN UNFORMATTED MULTI-RECORD NOISE MEMBER. THESE UNIT AND MEMBER NAMES ARE ELEMENTS OF USER PARAMETER, MEMSUM. ONE OR MORE NOISE MEMBERS WILL BE INPUT TO LEV TO BE SUMMED. THE NOISE MEMBERS HAVE THE FOLLOWING FORMAT:

RECORD	FORMAT	DESCRIPTION
1	I,*A8	NO. OF NOISE SOURCES, ARRAY OF SOURCES
2	2I,2RS	OBSERVER INDEX, NUMBER OF OBSERVER TIMES, AMBIENT DENSITY, AMBIENT SPEED OF SOUND
3	*RS	ARRAY OF OBSERVER TIMES
4	*RS	MEAN SQUARE PRESSURE FOR EACH FREQUENCY FOR FIRST TIME
5	*RS	MEAN SQUARE PRESSURE FOR EACH FREQUENCY FOR SECOND TIME .
.	.	.
.	.	.
.	.	.
1+M*(2+N)		(WHERE M IS THE NUMBER OF OBSERVERS AND N IS THE NUMBER OF OBSERVER TIMES. NOTE: N IS DIFFERENT FOR EACH OBSERVER)

LEV(PNLT)	UNFORMATTED MULTI-RECORD OUTPUT MEMBER IN THE FOLLOWING FORMAT OF A HEADER RECORD AND THREE RECORDS PER OBSERVER

RECORD	FORMAT	DESCRIPTION
1	I,*A8	NO. OF NOISE SOURCES, ARRAY OF SOURCES
2	2I	OBSERVER INDEX, NO. OF TIMES
3	*RS	ARRAY OF TIMES
4	*RS	ARRAY OF TONE-CORRECTED PERCEIVED NOISE LEVEL FOR ALL TIMES
.	.	.
.	.	.
.	.	.
1+3*M		(WHERE M IS THE NUMBER OF OBSERVERS)

Noise Levels Module (LEV)

ERRORS

NON-FATAL

1. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER (FMNMSG ERROR 4)
2. INSUFFICIENT DYNAMIC STORAGE. (FMNMSG ERROR 2)
3. REQUIRED USER PARAMETER NOT AVAILABLE. (FMNMSG ERROR 5)
4. REQUIRED USER PARAMETER TYPE OR NUMBER OF ELEMENTS IS INVALID. (FMNMSG ERROR 6)

FATAL - NONE

LDS REQUIREMENTS

LENGTH = NFREQ * (2+NMEM) + LONGESTREC + NS

WHERE NFREQ = NUMBER OF FREQUENCY VALUES

NMEM = NUMBER OF NOISE MEMBERS TO BE SUMMED

LONGESTREC = LENGTH OF THE LONGEST RECORD OF THE NOISE MEMBERS

NS = TOTAL NUMBER OF NOISE SOURCES READ FROM THE FIRST RECORD OF THE INPUT NOISE MEMBERS

4.6.2 Effective Noise Module-~~(EFF)~~

```
PURPOSE - TO COMPUTE EFFECTIVE PERCEIVED NOISE LEVEL (EPNL)
          AS A FUNCTION OF OBSERVER POSITION

AUTHOR  - CBF(L03/00/00)
          CBF(L03/00/04)

INPUT
  USER PARAMETERS                                           DEFAULT
    DTIME  - RECEPTION TIME INCREMENT (S)                      .5
    IPRINT - PRINTED OUTPUT OPTION                              3
             0 - NO PRINT DESIRED
             1 - INPUT PRINT ONLY
             2 - OUTPUT PRINT ONLY
             3 - BOTH INPUT AND OUTPUT PRINT

  MEMBERS  - DESCRIBED UNDER DATA BASE STRUCTURES
  NOTE : MEMBERS ARE SPECIFIED BY UNIT(MEMBER) NAME
      OBSERV(COORD)
      LEV(PNLT)


OUTPUT
  SYSTEM PARAMETER
    NERR   - EXECUTIVE SYSTEM PARAMETER FOR ERROR ENCOUNTERED
             DURING EXECUTION OF A FUNCTIONAL MODULE. NERR SET
             TO .TRUE. IF ERROR ENCOUNTERED
  MEMBERS
    EFF(EPNL)

  DATA BASE STRUCTURES
    OBSERV(COORD)     MULTI-RECORD MEMBER WITH FORMAT = 3RS,
                      CONTAINING ONE RECORD FOR EACH OBSERVER
                      WITH VALUES OF THE X,Y,Z COORDINATES
    LEV(PLNT)         UNFORMATTED MULTI-RECORD MEMBER IN
                      THE FOLLOWING FORMAT OF A HEADER RECORD
                      FOLLOWED BY THREE RECORDS PER OBSERVER
        RECORD  FORMAT                 DESCRIPTION
        -------------------------------------------------------
          1     I,*A8  NO. OF NOISE SOURCES, ARRAY OF NOISE
                       SOURCES
          2     2I     OBSERVER INDEX, NO. OF TIMES
          3     *RS    ARRAY OF TIMES
          4     *RS    ARRAY OF TONE-CORRECTED PERCEIVED NOISE
                         LEVEL FOR ALL TIMES
          .      .        .
          .      .        .
          .      .        .
          1+3*M        (WHERE M IS THE NUMBER OF OBSERVERS)
```

Effective Noise Module (EFF)

EFF(EPNL) UNFORMATTED MULTI-RECORD OUTPUT MEMBER IN THE FOLLOWING FORMAT OF A HEADER RECORD FOLLOWED BY A RECORD FOR EACH OBSERVER CONTAINING THE OBSERVER INDEX AND EPNL VALUE

RECORD	FORMAT	DESCRIPTION
1	I,*A8	NO. OF SOURCES, ARRAY OF NOISE SOURCES
2	I,RS	OBSERVER INDEX, EFFECTIVE PERCEIVED NOISE LEVEL (EPNL)
.	.	.
.	.	.
.	.	.
N+1		(WHERE N IS THE NUMBER OF OBSERVERS)

ERRORS

NON-FATAL

1. INSUFFICIENT DYNAMIC STORAGE (FMNMSG ERROR 2)
2. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER. (FMNMSG ERROR 4)

LDS REQUIREMENTS

LENGTH = 12 * NOBS + NS

WHERE NOBS = NUMBER OF OBSERVERS
NS = NUMBER OF NOISE SOURCES

4.7 UTILITIES

The modules for the ANOPP utilities were not available at Level 03/00/00.

4.8 NOISE SOURCES

4.8.1 Fan Noise Module (HDNFAN)

PURPOSE - PREDICT THE BROADBAND NOISE AND PURE TONES FOR AN AXIAL FLOW COMPRESSOR OR FAN BY THE HEIDMAN METHOD REFERENCE - NASA TM X-71763, INTERIM PREDICTION METHOD FOR FAN AND COMPRESSOR SOURCE NOISE, M. F. HEIDMAN

AUTHOR - CBF(L03/00/00)
CBF(L03/00/04)
MMF(L03/00/09)

INPUT

USER PARAMETERS

		DEFAULT SI UNITS
AE	ENGINE REFERENCE AREA (RS), M**2 (FT**2)	PI/4
RS	DISTANCE FROM SOURCE TO OBSERVER (RS), M (FT)	SQRT(AE)
AFAN	FAN INLET CROSS-SECTIONAL AREA (RS), RE AE	1.
DIAM	FAN ROTOR DIAMETER (RS), RE SQRT(AE)	1.128
MD	FAN ROTOR RELATIVE TIP MACH NUMBER AT DESIGN POINT (RS)	1.0
RSS	ROTOR-STATOR SPACING (RS), RE MEAN ROTOR BLADE CHORD	1.0
MDOT	MASS FLOWRATE (RS), RE RHOA * CA * AE	.2
MA	AIRCRAFT MACH NUMBER (RS)	0.
N	ROTATIONAL SPEED (RS), RE CA/DIAM	.3
DELTAT	TOTAL TEMPERATURE RISE ACROSS FAN (RS), RE TA	.2
CA	AMBIENT SPEED OF SOUND (RS), M/S (FT/S)	340.294
RHOA	AMBIENT DENSITY (RS), KG/M**3 (SLUG/FT**3)	1.225
NBANDS	NUMBER OF 1/3 OCTAVE BANDS FOR TONE FREQUENCY SHIFT (I)	0
NENG	NUMBER OF ENGINES (I)	1
NB	NUMBER OF ROTOR BLADES (I)	20
NV	NUMBER OF STATOR VANES (I)	50
IGV	INLET GUIDE VANE INDEX (I) 1, FOR A FAN WITH NO INLET GUIDE VANES 2, FOR A FAN WITH INLET GUIDE VANES	1
DIS	INLET FLOW DISTORTION INDEX (I) 1, IF THERE IS NO INLET FLOW DISTORTION 2, IF THERE IS INLET FLOW DISTORTION	1

Fan Noise Module (HDNFAN)

STIME	SOURCE TIME (RS)	0.
IOUT	TABLE OUTPUT AND PRINT OUTPUT OPTION (I) 0 NO PRINT BUT GENERATE TABLE HDNFAN(XXXNNN) -1 PRINT OUTPUT IN DB UNITS, BUT DO NOT GENERATE TABLE HDNFAN(XXXNNN) -2 PRINT OUTPUT IN DIMENSIONLESS FORM BUT DO NOT GENERATE TABLE HDNFAN (XXXNNN) -3 BOTH OPTIONS -1 AND -2 1 PRINT OUTPUT IN DB UNITS AND GENERATE TABLE HDNFAN(XXXNNN) 2 PRINT OUTPUT IN DIMENSIONLESS FORM AND GENERATE TABLE HDNFAN(XXXNNN) 3 BOTH OPTIONS 1 AND 2	3
IPRINT	PRINT FLAG (I) 0 NO PRINT DESIRED 1 INPUT PRINT ONLY 2 OUTPUT PRINT ONLY 3 BOTH INPUT AND OUTPUT PRINT	3
SCRNNN	INTEGER VALUE, NNN, .GT. 0 USED TO FORM TABLE UNIT MEMBER NAME HDNFAN(XXXNNN)	001
SCRXXX	THREE LETTER CODE XXX USED TO FORM TABLE UNIT MEMBER NAME HDNFAN(XXXNNN)	3HXXX
IUNITS	INPUT UNITS FLAG 7HENGLISH, ENGLISH UNITS 2HSI, SI UNITS	2HSI

(THE NEXT SIX CODES HAVE THE FOLLOWING VALUES)
(.FALSE. - DO NOT INCLUDE)
(.TRUE. - INCLUDE IN TOTAL PREDICTION)

INRS	INLET ROTOR-STATOR INTERACTION TONES	T
INCT	COMBINATION TONE NOISE	T
INDIS	INLET FLOW DISTORTION TONES	T
IDBB	DISCHARGE BROADBAND NOISE	T
IDRS	DISCHARGE ROTOR-STATOR INTERACTION TONES	T
INBB	INLET BROADBAND NOISE	T

DATA BASE UNIT MEMBERS
(DESCRIBED UNDER DATA BASE STRUCTURES)
SFIELD(FREQ)
SFIELD(THETA)
SFIELD(PHI)

OUTPUT
USER PARAMETERS

RS	DISTANCE FROM SOURCE TO OBSERVER

Fan Noise Module (HDNFAN)

SYSTEM PARAMETERS

NERR — .TRUE., IMPLIES AN ERROR WAS ENCOUNTERED DURING MODULE EXECUTION
.FALSE., NO ERROR ENCOUNTERED

DATA BASE UNIT MEMBERS

HDNFAN(XXXNNN) — SEE FORMAT UNDER DATA BASE STRUCTURES. NOTE MEMBER NAME XXXNNN IS FORMED FROM USER PARAMETERS SCRXXX AND SCRNNN. OUTPUT OF THIS TABLE IS CONTROLLED BY USER PARAMETER IOUT.

DATA BASE STRUCTURES

SFIELD(FREQ) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF 1/3 OCTAVE BAND CENTER FREQUENCIES IN HZ

SFIELD(THETA) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF THE POLAR DIRECTIVITY ANGLE IN DEG

SFIELD(PHI) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF THE AZIMUTHAL DIRECTIVITY ANGLE IN DEG

HDNFAN(XXXNNN) - TYPE 1 TABLE CONTAINING MEAN SQUARE ACOUSTIC PRESSURE AS A FUNCTION OF (1) FREQUENCY, (2) DIRECTIVITY ANGLE AND (3) AZIMUTHAL ANGLE

ERRORS

NON-FATAL

1. INSUFFICIENT LOCAL DYNAMIC STORAGE.
2. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.

FATAL - NONE

REMARKS

REFERENCES

HEIDMAN, M. F., INTERIM PREDICTION METHOD FOR FAN AND COMPRESSOR SOURCE NOISE, NASA TM X-71763, JUNE 1975.

LDS REQUIREMENTS

LENGTH = (NFREQ * NTHETA * NPHI) + (NTHETA * NPHI) + NFREQ + NTHETA + NPHI + 3 * (NFREQ * NTHETA) + NTHETA

WHERE

NFREQ = NUMBER OF FREQUENCY VALUES
NTHETA = NUMBER OF DIRECTIVITY ANGLES
NPHI = NUMBER OF AZIMUTHAL ANGLES

Fan Noise Module (HDNFAN)

GDS REQUIREMENTS
SUFFICIENT ALLOCATION FOR TABLE HDNFAN(XXXNNN)

4.8.2 Combustion Noise Module (GECOR)

PURPOSE - GECOR PREDICTS THE NOISE FROM CONVENTIONAL COMBUSTORS INSTALLED IN GAS TURBINE ENGINES. THE METHOD IS BASED ON A PROPOSED APPENDIX TO THE SOCIETY OF AUTOMOTIVE ENGINEERS (SAE) AEROSPACE RECOMMENDED PRACTICE (ARP) NUMBER 876.

AUTHOR - WKB(L03/00/00)
MMF(L03/00/09)

INPUT

USER PARAMETERS

		TYPE	DEFAULT (SI)
AE	ENGINE REFERENCE AREA. M**2 (FT**2)	RS	PI/4
A	COMBUSTOR ENTRANCE AREA DIMENSIONLESS RE ENGINE REFERENCE AREA	RS	1.0
NENG	NUMBER OF ENGINES	I	1
RS	DISTANCE FROM SOURCE TO OBSERVER	RS	SQRT(AE)
MDOT	COMBUSTOR ENTRANCE MASS FLOWRATE DIMENSIONLESS RE RHOA*CA*AE	RS	.2
MA	AIRCRAFT MACH NUMBER	RS	0.0
PI	COMBUSTOR ENTRANCE TOTAL PRESSURE DIMENSIONLESS RE AMBIENT PRESSURE	RS	1.0
TI	COMBUSTOR ENTRANCE TOTAL TEMPERATURE DIMENSIONLESS RE AMBIENT TEMPERATURE	RS	1.0
TJ	COMBUSTOR EXIT TOTAL TEMPERATURE DIMENSIONLESS RE AMBIENT TEMPERATURE	RS	2.0
TDDELT	DESIGN TURBINE TEMPERATURE RISE DIMENSIONLESS RE AMBIENT TEMPERATURE	RS	0.5
CA	AMBIENT SPEED OF SOUND, M/S (FT/SEC)	RS	340.294
RHOA	AMBIENT DENSITY, KG/M**3 (SLUG/FT**3)	RS	1.225
STIME	SOURCE TIME AT WHICH NOISE IS CALCULATED	RS	0.0
ICA078	IF .TRUE., USE THE SAE ARP 876 ENVELOPE SPECTRUM FUNCTION AS A REPLACEMENT FOR THE SPECTRAL DISTRIBUTION FUNCTION. IF .FALSE., THE REGULAR SPECTRAL DISTRIBUTION FUNCTION IS USED	L	.FALSE.

SCRXXX	THREE LETTER CODE XXX USED TO FORM TABLE UNIT MEMBER NAME GECOR(XXXNNN). SEE USER PARAMETER SCRNNN	A	3HXXX
SCRNNN	INTEGER VALUE, NNN .GT. 0, USED TO FORM TABLE UNIT MEMBER NAME GECOR(XXXNNN). SEE USER PARAMETER SCRXXX.	I	001
IOUT	OUTPUT CODE (FOR TABLE AND/OR PRINTED OUTPUT) 0 NO PRINT BUT GENERATE TABLE GECOR(XXXNNN) -1 PRINT OUTPUT IN DB UNITS BUT DO NOT GENERATE TABLE GECOR(XXXNNN) -2 PRINT OUTPUT IN DIMENSIONLESS UNITS BUT DO NOT GENERATE TABLE GECOR(XXXNNN) -3 BOTH OPTIONS -1 AND -2 1 PRINT OUTPUT IN DB UNITS AND GENERATE TABLE GECOR(XXXNNN) 2 PRINT OUTPUT IN DIMENSIONLESS UNITS AND GENERATE TABLE GECOR(XXXNNN) 3 BOTH OPTIONS 1 AND 2	I	3
IPRINT	PRINTED OUTPUT OPTION CODE 0 NO PRINT DESIRED 1 INPUT PRINT ONLY 2 OUTPUT PRINT ONLY 3 BOTH INPUT AND OUTPUT PRINT	I	3
IUNITS	=2HSI , INPUTS ARE IN SI UNITS =7HENGLISH, INPUTS ARE IN ENGLISH UNITS	A	2HSI

DATA BASE UNITS AND MEMBERS

SFIELD(FREQ) SFIELD(THETA) SFIELD(PHI)
(FORMATS ARE DESCRIBED UNDER DATA BASE STRUCTURES)

OUTPUT

USER PARAMETERS

RS	RADIAL DISTANCE FROM SOURCE TO OBSERVER (RS)
GECERR	=0, NO ERRORS OCCURRED =1, MEMBER MANAGER ERROR OCCURRED ON OPEN =2, INSUFFICIENT LDS AVAILABLE =3, MEMBER MANAGER ERROR OCCURRED ON READ =4, AT LEAST ONE USER PARAMETER HAS AN INVALID VALUE

```
                =5, UNABLE TO BUILD TABLE GECOR(XXXNNN)
                =6, AN INVALID VALUE WAS CALCULATED DURING
                    EXECUTION
  SYSTEM PARAMETERS
    NERR                  SYSTEM ERROR PARAMETER
                          .TRUE.   ERROR ENCOUNTERED DURING
                                   EXECUTION (GECERR=1,2,3,5)
                          .FALSE.  NO ERROR ENCOUNTERED

  DATA BASE UNITS AND MEMBERS

    GECOR(XXXNNN)         SEE FORMAT DESCRIPTION UNDER DATA BASE
                          STRUCTURES. NOTE MEMBER NAME XXXNNN
                          IS FORMED FROM USER PARAMETERS SCRXXX
                          AND SCRNNN. OUTPUT OF THIS TABLE IS
                          CONTROLLED BY USER PARAMETER IOUT.
```

DATA BASE STRUCTURES

```
  SFIELD( FREQ ) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING
                   VALUES OF 1/3 OCTAVE BAND CENTER FREQUENCIES
                   IN HZ
  SFIELD( THETA) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING
                   VALUES OF POLAR DIRECTIVITY ANGLE IN DEGREES
  SFIELD( PHI  ) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING
                   VALUES OF AZIMUTHAL DIRECTIVITY ANGLE IN
                   DEG
  GECOR( XXXNNN) - TYPE 1 TABLE OF MEAN SQUARE ACOUSTIC
                   PRESSURE DATA AS A FUNCTION OF (1) FREQUENCY,
                   (2) DIRECTIVITY ANGLE, AND (3) AZIMUTH ANGLE
```

ERRORS

NON-FATAL

1. INSUFFICIENT SPACE IN LDS
2. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER
3. USER-PARAMETER VALUE OUT OF RANGE - DEFAULT TO BE USED

REFERENCES

EMMERLING, J.J., CORE ENGINE NOISE CONTROL PROGRAM, VOLUME III, SUPPLEMENT I, PREDICTION METHODS. REPORT NO. FAA-RD-74-125, III-I, MARCH, 1976.

MATTA, RAM K., PROPOSED APPENDIX TO ARP876-COMBUSTION NOISE PREDICTION. SUBMITTED TO SAE A-21 JET NOISE SUBCOMMITTEE, JULY, 1977.

Combustion Noise Module (GECOR)

```
LDS REQUIREMENTS
  LENGTH = NFREQ*NTHETA*NPHI + NTHETA*NPHI +
           NFREQ + NTHETA + NPHI
      WHERE
           NFREQ  = NUMBER OF FREQUENCIES
           NTHETA = NUMBER OF DIRECTIVITY ANGLES
           NPHI   = NUMBER OF AZIMUT'' ANGLES

GDS REQUIREMENTS
  ALLOCATION REQUIRED FOR TABLE GECOR(XXXNNN)
  AND UNIT MEMBERS SFIELD( FREQ, THETA, PHI )
```

4.8.3 Turbine Noise Module (GETUR)

PURPOSE - GETUR PREDICTS THE BROADBAND NOISE AND PURE TONES FOR AN AXIAL FLOW TURBINE

AUTHOR - WKB(L03/00/00)
MMF(L03/00/09)

INPUT

USER PARAMETERS

		TYPE	DEFAULT (SI)
AE	ENGINE REFERENCE AREA, M**2 (FT**2)	RS	PI/4
NENG	NUMBER OF ENGINES	I	1
RS	DISTANCE FROM SOURCE TO OBSERVER	RS	SQRT(AE)
STIME	SOURCE TIME AT WHICH NOISE IS CALCULATED	RS	0.0
AREA	TURBINE INLET CROSS-SECTIONAL AREA DIMENSIONLESS RE ENGINE REFERENCE AREA	RS	1.0
F	FUEL TO AIR RATIO	RS	.0
HA	ABSOLUTE HUMIDITY	RS	.0
NBLADE	NUMBER OF ROTOR BLADES	I	20
D	TURBINE ROTOR DIAMETER DIMENSIONLESS RE SQRT(ENGINE REFERENCE AREA)	RS	1.0
MA	AIRCRAFT MACH NUMBER	RS	0.0
ROTSPD	ROTATIONAL SPEED DIMENSIONLESS RE (CA/D)	RS	.3
TTI	ENTRANCE TOTAL TEMPERATURE DIMENSIONLESS RE AMBIENT TEMPERATURE	RS	3.0
TSJ	EXIT STATIC TEMPERATURE DIMENSIONLESS RE AMBIENT TEMPERATURE	RS	2.0
CA	AMBIENT SPEED OF SOUND, M/S (FT/S)	RS	340.294
RHOA	AMBIENT DENSITY	RS	1.225
SCRXXX	THREE LETTER CODE XXX USED TO FORM TABLE UNIT MEMBER NAME GETUR(XXXNNN). SEE USER PARAMETER SCRNNN	A	3HXXX
SCRNNN	INTEGER VALUE, NNN .GT. 0, USED TO FORM TABLE UNIT MEMBER NAME GETUR(XXXNNN). SEE USER PARAMETER SCRXXX	I	001
BROAD	COMPUTE NOISE DATA FOR BROADBAND SPECTRA	L	.TRUE.
PURE	COMPUTE NOISE DATA FOR PURE TONE SPECTRA	L	.TRUE.

```
  IOUT       OUTPUT CODE FOR TABLE AND/OR PRINTED  I       3
             OUTPUT
              0   NO PRINT BUT GENERATE TABLE
                  GETUR(XXXNNN)
             -1   PRINT OUTPUT IN DB UNITS BUT DO
                  NOT GENERATE TABLE GETUR(XXXNNN)
             -2   PRINT OUTPUT IN DIMENSIONLESS
                  UNITS BUT DO NOT GENERATE
                  TABLE GETUR(XXXNNN)
             -3   BOTH OPTIONS -1 AND -2
              1   PRINT OUTPUT IN DB UNITS AND
                  GENERATE TABLE GETUR(XXXNNN)
              2   PRINT OUTPUT IN DIMENSIONLESS
                  UNITS AND GENERATE TABLE
                  GETUR(XXXNNN)
              3   BOTH OPTIONS 1 AND 2
  IPRINT     OUTPUT PRINT OPTION CODE              I       3
              0   NO PRINT DESIRED
              1   PRINT INPUTS ONLY
              2   PRINT OUTPUTS ONLY
              3   BOTH OPTIONS 1 AND 2
  IUNITS     =2HSI     , INPUTS ARE IN SI UNITS    A    2HSI
             =7HENGLISH, INPUTS ARE IN ENGLISH
                         UNITS
 DATA BASE UNITS AND MEMBERS
  SFIELD(FREQ)    SFIELD(THETA)      SFIELD(PHI)
  ( FORMATS ARE DESCRIBED UNDER DATA BASE STRUCTURES )

OUTPUT

 USER PARAMETERS
  RS         DISTANCE FROM SOURCE TO OBSERVER,
             M (FT)
  GETERR     =0, NO ERRORS OCCURRED
             =1, MEMBER MANAGER ERROR OCCURRED ON OPEN
             =2, INSUFFICIENT LDS AVAILABLE
             =3, MEMBER MANAGER ERROR OCCURRED ON READ
             =4, AT LEAST ONE USER PARAMETER HAS AN INVALID
                 VALUE
             =5, UNABLE TO BUILD TABLE GETUR(XXXNNN)
             =6, AN INVALID VALUE WAS CALCULATED DURING
                 EXECUTION

 SYSTEM PARAMETERS
  NERR       SYSTEM ERROR PARAMETER
             =.TRUE. , ERROR OCCURRED DURING EXECUTION
                       (GETERR = 1, 2, 3, 5)
             =.FALSE., NO ERROR OCCURRED
```

Turbine Noise Module (GETUR)

DATA BASE UNITS AND MEMBERS

GETUR(XXXNNN) SEE FORMAT DESCRIPTIONS UNDER DATA BASE STRUCTURES. NOTE MEMBER NAME XXXNNN IS FORMED FROM USER PARAMETERS SCRNNN AND SCRXXX. OUTPUT OF THIS TABLE IS CONTROLLED BY USER PARAMETER IOUT

DATA BASE STRUCTURES

SFIELD(FREQ) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF 1/3 OCTAVE BAND CENTER FREQUENCIES IN HZ

SFIELD(THETA) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF POLAR DIRECTIVITY ANGLE IN DEG

SFIELD(PHI) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF AZIMUTHAL DIRECTIVITY ANGLE IN DEG

GETUR(XXXXNNN) - TYPE 1 TABLE OF MEAN SQUARE ACOUSTIC PRESSURE DATA AS A FUNCTION OF (1) FREQUENCY, (2) DIRECTIVITY ANGLE, AND (3) AZIMUTH ANGLE

ERRORS

NON-FATAL

1. INSUFFICIENT SPACE IN LDS
2. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER
3. USER PARAMETER VALUE OUT OF RANGE - DEFAULT TO BE USED
4. UNABLE TO CALCULATE THE MASS FRACTION OF EACH AIR CONSTITUENT
5. UNABLE TO CALCULATE TOTAL ENTRANCE AND EXIT ENTHALPY

FATAL - NONE

LDS REQUIREMENTS

```
LENGTH = NFREQ*NTHETA*NPHI + NTHETA*NPHI +
         NFREQ + NTHETA + NPHI
      WHERE
         NFREQ  = NUMBER OF FREQUENCIES
         NTHETA = NUMBER OF DIRECTIVITY ANGLES
         NPHI   = NUMBER OF AZIMUTH ANGLES
```

GDS REQUIREMENTS

SUFFICIENT ALLOCATION FOR TABLE GETUR(XXXNNN) AND UNIT MEMBERS SFIELD(FREQ, THETA, PHI).

4.8.4 Single Stream Circular Jet Noise Module (SGLJET)

PURPOSE - PREDICT 1/3 OCTAVE BAND CIRCULAR JET NOISE INCORPORATING FORWARD FLIGHT EFFECTS WITH METHODS DEVELOPED BY THE SAE A-21 JET NOISE SUB-COMMITTEE.

AUTHOR - WKB(L03/00/00)
MMF(L03/00/09)

INPUT
USER PARAMETERS

		TYPE	DEFAULT IN SI UNITS
AJ	AREA OF FULLY EXPANDED PRIMARY JET DIMENSIONLESS UNITS, RE ENGINE REFERENCE AREA	RS	1.0
RHOJ	DENSITY OF PRIMARY JET DIMENSIONLESS UNITS, RE AMBIENT DENSITY	RS	1.0
TJ	ABSOLUTE TOTAL TEMPERATURE OF THE PRIMARY JET - DIMENSIONLESS UNITS, RE AMBIENT TEMPERATURE	RS	1.0
VJ	VELOCITY OF PRIMARY JET RELATIVE TO NOZZLE EXIT - DIMENSIONLESS UNITS, RE AMBIENT SPEED OF SOUND	RS	1.0
RS	RADIAL DISTANCE FROM NOZZLE EXIT TO OBSERVER, M (FT)	RS	SQRT(AE)
RHOA	AMBIENT DENSITY, KG/M**3 (SLUGS/FT**3)	RS	1.225
CA	AMBIENT SPEED OF SOUND, M/SEC (FT/SEC)	RS	340.294
STIME	SOURCE TIME AT WHICH NOISE IS CALCULATED	RS	0.0
MA	AIRCRAFT MACH NUMBER	RS	0.
AE	ENGINE REFERENCE AREA, M**2 (FT**2)	RS	.7854
NENG	NO. OF ENGINES	I	1
SCRXXX	THREE LETTER CODE XXX USED TO FORM TABLE UNIT MEMBER NAME SGLJET(XXXNNN). SEE USER PARAMETER SCRNNN.	A	3HXXX
SCRNNN	INTEGER VALUE, NNN .GT. 0, USED TO FORM TABLE UNIT MEMBER NAME SGLJET(XXXNNN). SEE USER PARAMETER SCRXXX.	I	1
IUNITS	=2HSI , INPUTS ARE SI UNITS =7HENGLISH, INPUTS ARE ENGLISH UNITS	A	2HSI

```
IOUT        OUTPUT CODE FOR TABLE AND                I   3
            PRINTED OUTPUT
            0    NO PRINT BUT GENERATE TABLE
                 SGLJET(XXXNNN).
           -1    PRINT OUTPUT IN DB UNITS BUT
                 DO NOT GENERATE TABLE
                 SGLJET(XXXNNN).
           -2    PRINT OUTPUT IN DIMENSIONLESS
                 FORM BUT DO NOT GENERATE
                 TABLE SGLJET(XXXNNN).
           -3    BOTH OPTIONS -1 AND -2
            1    PRINT OUTPUT IN DB UNITS AND
                 GENERATE TABLE SGLJET(XXXNNN).
            2    PRINT OUTPUT IN DIMENSIONLESS
                 FORM AND GENERATE TABLE
                 SGLJET(XXXNNN).
            3    BOTH OPTIONS 1 AND 2
IPRINT      PRINTED OUTPUT OPTION CODE               I   3
            0    NO PRINT DESIRED
            1    INPUT PRINT ONLY
            2    OUTPUT PRINT ONLY
            3    BOTH INPUT AND OUTPUT PRINT
            ( MEMBER OUTPUT ALWAYS GENERATED
              ACCORDING TO IOUT VALUE )

DATA BASE UNITS AND MEMBERS
  ( FORMATS ARE DESCRIBED UNDER DATA BASE STRUCTURES )

  SFIELD( FREQ )   SFIELD( PHI )   SFIELD( THETA )
  SAE( MTH )       SAE( NDF )      SAE( OM )
  SAE( PDF )       SAE( SCF )      SAE( SJC )

OUTPUT
  USER PARAMETERS
    RS          RADIAL DISTANCE FROM NOZZLE EXIT TO OBSERVER

  DATA BASE UNITS AND MEMBERS

    SGLJET( XXXNNN )      SEE FORMAT UNDER DATA BASE STRUCTURES.
                          NOTE MEMBER NAME XXXNNN IS FORMED FROM
                          USER PARAMETERS SRCXXX AND SRCNNN.
                          OUTPUT OF THIS UNIT MEMBER TABLE IS
                          CONTROLED BY USER PARAMETER IOUT.
  SYSTEM PARAMETERS
    NERR        SYSTEM ERROR PARAMETER
                .TRUE.     ERROR ENCOUNTERED DURING MODULE
                           EXECUTION
                .FALSE.    NO ERROR ENCOUNTERED
```

Single Stream Circular Jet Noise Module (SGLJET)

DATA BASE STRUCTURES

SFIELD(FREQ) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF 1/3 OCTAVE BAND CENTER FREQUENCIES IN HZ

SFIELD(PHI) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF AZIMUTHAL DIRECTIVITY ANGLE IN DEG

SFIELD(THETA) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF POLAR DIRECTIVITY ANGLE IN DEG

TABLES

UNIT(TABLE)	TYPE	DESCRIPTION
SAE(OM)	1	JET VARIABLE DENSITY EXPONENT AS FUNCTION OF LOG10 OF RATIO OF JET VELOCITY TO SPEED OF SOUND
SAE(PDF)	1	LOG10 OF POWER DEVIATION FACTOR FROM V**8 LAW AS FUNCTION OF LOG10 OF RATIO OF JET VELOCITY TO SPEED OF SOUND
SAE(NDF)	1	LOG10 OF NORMALIZED DIRECTIVITY FUNCTION FOR CIRCULAR JETS AS FUNCTION OF (1) DIR. ANGLE IN DEG AND (2) LOG10 OF RATIO OF JET VELOCITY TO SPEED OF SOUND
SAE(MTH)	1	FORWARD VELOCITY INDEX AS A FUNCTION OF DIRECTIVITY ANGLE IN DEG
SAE(SJC)	1	NORMALIZED SPECTRAL DISTRIBUTION FACTOR AS FUNCTION OF (1) LOG10 OF CORRECTED STROUHAL NUMBER, (2) RATIO OF JET VELOCITY TO SPEED OF SOUND, AND (3) DIRECTIVITY ANGLE FROM JET INLET IN DEG, (4) RATIO OF JET TEMPERATURE TO AMBIENT TEMPERATURE
SAE(SCF)	1	STROUHAL NUMBER CORRECTION FACTOR AS FUNCTION OF RATIO OF JET VELOCITY TO SPEED OF SOUND AND DIRECTIVITY ANGLE
SGLJET(XXXNNN)	1	MEAN SQUARE ACOUSTIC PRESSURE AS A FUNCTION OF (1) FREQUENCY, (2) DIRECTIVITY ANGLE AND (3) AZIMUTH ANGLE

ERRORS

NON-FATAL

1. INSUFFICIENT SPACE IN LDS
2. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER

3. INTERPOLATION REQUEST NOT COMPLETE
4. USER PARAMETER VALUE OUT OF RANGE - DEFAULT TO BE USED
5. USER PARAMETER MA MAY NOT BE GREATER THAN USER PARAMETER VJ

FATAL - NONE

LDS REQUIREMENTS

LENGTH = NFREQ*NTHETA*NPHI + NTHETA*NPHI + NFREQ + NTHETA + NPHI

WHERE

NFREQ = NUMBER OF FREQUENCIES
NTHETA = NUMBER OF DIRECTIVITY ANGLES
NPHI = NUMBER OF AZIMUTH ANGLES

GDS REQUIREMENTS

ALLOCATION REQUIRED FOR TABLES SAE(OM), SAE(PDF), SAE(NDF), SAE(MTH), SAE(SCJ), SAE(SCF), AND SGLJET(XXXNNN) AND FOR UNIT MEMBERS SFIELD(FREQ, THETA, AND PHI).

4.8.5 Circular Jet Shock Cell Noise Module (SAESHK)

PURPOSE - PREDICT THE BROADBAND SHOCK ASSOCIATED NOISE FROM A SINGLE CONVERGENT NOZZLE OPERATING AT SUPERCRITICAL PRESSURE RATIOS.

AUTHOR - WKB(L03/00/00)
MMF(L03/00/09)

INPUT

USER PARAMETERS

		TYPE	DEFAULT SI VAL.
AE	ENGINE REFERENCE AREA, M**2 (FT**2)	RS	PI/4
NENG	NUMBER OF ENGINES	I	1
NSHK	NUMBER OF SHOCKS	I	8
DELTA	ANGLE BETWEEN FLIGHT VECTOR AND ENGINE INLET AXIS IN DEG. THIS IS THE ENGINE INCLINATION ANGLE + ANGLE OF ATTACK.	RS	0.0
RS	DISTANCE FROM NOZZLE EXIT TO A PSEUDO OBSERVER, M (FT)	RS	SQRT(AE)
AJ	AREA OF THE JET DIMENSIONLESS UNITS, (RE ENGINE REFERENCE AREA)	RS	1.0
MA	AIRCRAFT MACH NUMBER	RS	0.0
MJ	FULLY EXPANDED JET MACH NUMBER	RS	1.414
TJ	JET TOTAL TEMPERATURE DIMENSIONLESS UNITS, (RE AMBIENT TEMPERATURE)	RS	1.0
VJ	FULLY EXPANDED JET VELOCITY DIMENSIONLESS UNITS, (RE AMBIENT SPEED OF SOUND)	RS	1.0
CA	AMBIENT SPEED OF SOUND, M/S (FT/S)	RS	340.294
RHOA	AMBIENT DENSITY, KG/M**3 (LB/FT**3)	RS	1.225
STIME	SOURCE TIME AT WHICH NOISE IS CALCULATED	RS	0.0
ICA078	IF .TRUE., CALCULATE SHOCK NOISE FOR THE ICAO REFERENCE PROCEDURE (1978) WHICH IS BASED ON SAE ARP 876. THE ONLY DIFFERENCE IS FOR ICA078=.TRUE., AN ADDITIONAL DIRECTIVITY FUNCTION IS APPLIED TO THE NOISE DATA.	L	.FALSE.

Circular Jet Shock Cell Noise Module (SAESHK)

IPRINT	PRINT FLAG 0 NO PRINT DESIRED 1 INPUT PRINT ONLY 2 OUTPUT PRINT ONLY 3 BOTH INPUT AND OUTPUT PRINT	I	3
IOUT	TABLE OUTPUT AND PRINT OUTPUT OPTION 0 NO PRINT BUT GENERATE TABLE SAESHK(XXXNNN) -1 PRINT OUTPUT IN DB UNITS BUT DO NOT GENERATE TABLE SAESHK(XXXNNN) -2 PRINT OUTPUT IN DIMENSIONLESS FORM BUT DO NOT GENERATE TABLE SAESHK(XXXNNN) -3 BOTH OPTIONS -1 AND -2 1 PRINT OUTPUT IN DB UNITS AND GENERATE TABLE SAESHK(XXXNNN) 2 PRINT OUTPUT IN DIMENSIONLESS FORM AND GENERATE TABLE SAESHK(XXXNNN) 3 BOTH OPTIONS 1 AND 2	I	3
IUNITS	=2HSI , INPUTS ARE IN SI UNITS =7HENGLISH, INPUTS ARE IN ENGLISH UNITS	A	2HSI
SCRXXX	THREE LETTER CODE XXX USED TO FORM TABLE UNIT MEMBER NAME SAESHK(XXXNNN). SEE USER PARAMETER SCRNNN	A	3HXXX
SCRNNN	INTEGER VALUE, NNN .GT. 0, USED TO FORM TABLE UNIT MEMBER SAESHK(XXXNNN). SEE USER PARAMETER SCRXXX	I	001

DATA BASE UNITS AND MEMBERS
(FORMATS ARE DESCRIBED UNDER DATA BASE STRUCTURES)
SFIELD(FREQ) SFIELD(THETA) SFIELD(PHI)

OUTPUT
USER PARAMETERS

RS	DISTANCE FROM NOZZLE EXIT TO OBSERVER
SAEERR	=0. NO ERRORS OCCURRED =1, MEMBER MANAGER ERROR OCCURRED ON OPEN =2, INSUFFICIENT LDS AVAILABLE =3, MEMBER MANAGER ERROR OCCURRED ON READ =4, AT LEAST ONE USER PARAMETER HAS AN INVALID VALUE =5, UNABLE TO BUILD OUTPUT TABLE SAESHK(XXXNNN) =6, AN INVALID VALUE WAS CALCULATED DURING EXECUTION

Circular Jet Shock Cell Noise Module (SAESHK)

```
SYSTEM PARAMETERS
  NERR        .TRUE., IMPLIES ERROR ENCOUNTERED DURING MODULE
                      EXECUTION ( SAEERR=1,2,3,5 )
              .FALSE., NO ERRORS ENCOUNTERED

DATA BASE UNIT MEMBERS
  SAESHK(XXXNNN)      SEE FORMAT UNDER DATA BASE STRUCTURES.
                      NOTE MEMBER NAME XXXNNN IS FORMED FROM
                      USER PARAMETERS SCRXXX AND SCRNNN.
                      OUTPUT OF THIS TABLE IS CONTROLLED
                      BY USER PARAMETER IOUT.

DATA BASE STRUCTURES
  SFIELD( FREQ ) - 1 RECORD MEMBER IN *RS FORMAT CONTAINING
                   VALUES OF 1/3 OCTAVE BAND CENTER
                   FREQUENCIES IN HZ
  SFIELD( THETA ) -1 RECORD MEMBER IN *RS FORMAT CONTAINING
                   VALUES OF THE POLAR DIRECTIVITY ANGLE IN
                   DEG
  SFIELD( PHI )  - 1 RECORD MEMBER IN *RS FORMAT CONTAINING
                   VALUES OF AZIMUTHAL DIRECTIVITY ANGLE IN
                   DEG
  SAESHK(XXXNNN) - TYPE 1 TABLE CONTAINING MEAN SQUARE ACOUSTIC
                   PRESSURE AS A FUNCTION OF (1) FREQUENCY,
                   (2) DIRECTIVITY ANGLE, AND (3) AZIMUTH
                   ANGLE

ERRORS
  NON-FATAL
    1. INSUFFICIENT LDS SPACE.
    2. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.
    3. USER PARAMETER VALUE OUT OF RANGE - DEFAULT TO BE USED.

REMARKS
  REFERENCES
    TANNA, H. K., PREDICTION METHOD FOR SHOCK ASSOCIATED NOISE
          FROM CONVERGENT NOZZLES AT SUPERCRITICAL CONDITIONS,
          PROPOSAL TO SAE-A21 COMMITTEE, JULY 1977.
    HARPER-BOURNE, M., AND FISHER, M. J., THE NOISE FROM SHOCK
          WAVES IN SUPERSONIC JETS, AGARD REPORT, CP-131,
          1973.
    TANNA, H. K., DEAN, P. D., AND BURRIN, R. H., THE
          GENERATION AND RADIATION OF SUPERSONIC JET NOISE, VOL.
          4 - SHOCK ASSOCIATED NOISE DATA, AIR FORCE AERO-
          PROPULSION LABORATORY TECHNICAL REPORT, AFAPL-TR-
          76-65, 1976.
```

Circular Jet Shock Cell Noise Module (SAESHK)

RANGE OF APPLICATION

THIS METHOD IS APPLICABLE TO BOTH HEATED AND UNHEATED JET. THE OBSERVER ANGLE SHOULD BE RESTRICTED TO THE RANGE OF 0 - 130 DEGREES FOR UNHEATED JETS AND 0 - 150 DEGREES FOR HEATED JETS. WITHIN THE CONE OF DIRECTIONS BEYOND THE UPPER LIMIT FOR THE OBSERVER ANGLE, THE JET MIXING NOISE DOMINATES AND THEREFORE THE SHOCK NOISE COMPONENT CAN BE NEGLECTED

LDS REQUIREMENTS

```
LENGTH = NFREQ*NTHETA*NPHI + NTHETA*NPHI +
         NFREQ + NTHETA + NPHI
         WHERE
           NFREQ  = NUMBER OF FREQUENCIES
           NTHETA = NUMBER OF DIRECTIVITY ANGLES
           NPHI   = NUMBER OF AZIMUTH ANGLES
```

GDS REQUIREMENTS

SUFFICIENT ALLOCATION FOR TABLE SAESHK(XXXNNN) AND UNIT MEMBERS SFIELD (FREQ, THETA, PHI)

4.8.6 Stone Jet Noise Module (STNJET)

PURPOSE - PREDICT THE FAR-FIELD MEAN SQUARE ACOUSTIC PRESSURE FOR SINGLE STREAM AND COAXIAL CIRCULAR JETS BY THE STONE METHOD

AUTHOR - CBF(L03/00/04)
MMF(L03/00/09)

INPUT

USER PARAMETERS		DEFAULT SI UNITS
CA	AMBIENT SPEED OF SOUND (RS), M/S (FT/S)	340.294
RHOA	AMBIENT DENSITY (RS), KG/M**3 (SLUG/FT**3)	1.225
MA	AIRCRAFT MACH NUMBER (RS)	0.
RS	DISTANCE FROM NOZZLE EXIT TO PSEUDO-OBSERVER (RS), M (FT)	SQRT (PI/4.)
AE	ENGINE REFERENCE AREA (RS), M**2 (FT**2)	PI/4.
A1	PRIMARY FULLY EXPANDED JET AREA (RS), RE AE	1.
DE1	ACTUAL PRIMARY STREAM EQUIVALENT DIAMETER (RS), RE SQRT(AE)	2/ SQRT(PI)
DH1	ACTUAL PRIMARY STREAM HYDRAULIC DIAMETER (RS), RE SQRT(AE)	2/ SQRT(PI)
V1	PRIMARY STREAM JET VELOCITY (RS), RE CA	1.
RHO1	PRIMARY STREAM JET DENSITY (RS), RE RHOA	1.
T1	PRIMARY STREAM TOTAL TEMPERATURE (RS), RE TA	1.
M1	PRIMARY STREAM MACH NUMBER (RS)	1.
A2	SECONDARY FULLY EXPANDED JET AREA (RS), RE AE	0.
V2	SECONDARY STREAM JET VELOCITY (RS), RE CA	0.
RHO2	SECONDARY STREAM JET DENSITY (RS), RE RHOA	1.
T2	SECONDARY STREAM TOTAL TEMPERATURE (RS), RE TA	1.
M2	SECONDARY STREAM MACH NUMBER (RS)	0.
STIME	SOURCE TIME (RS), SEC	0.
NENG	NUMBER OF ENGINES (I)	1
SCRNNN	INTEGER VALUE, NNN, .GT. 0 USED TO FORM TABLE UNIT MEMBER NAME STNJET(XXXNNN)	1
SCRXXX	THREE LETTER CODE, XXX, USED TO FORM TABLE UNIT MEMBER NAME STNJET(XXXNNN)	3HXXX
IUNITS	INPUT UNITS FLAG (I) 7HENGLISH, ENGLISH UNITS 2HSI, SI UNITS	2HSI

```
PLUG      .TRUE.,  IMPLIES NOZZLE WITH CONICAL      .FALSE.
                   PLUG
          .FALSE., IMPLIES NOZZLE WITHOUT PLUG
CIRCLE    .TRUE.,  IMPLIES CIRCULAR NOZZLE          .FALSE.
          .FALSE., IMPLIES COAXIAL NOZZLE
SUPER     .TRUE.,  IMPLIES SUPERSONIC CIRCULAR      .FALSE.
                   NOZZLE OR SUPERSONIC PRIMARY JET
                   OF A COAXIAL NOZZLE
          .FALSE., IMPLIES SUBSONIC CIRCULAR
                   NOZZLE OR SUBSONIC PRIMARY AND
                   SECONDARY JETS OF A COAXIAL NOZZLE
IOUT      TABLE OUTPUT AND PRINT OUTPUT OPTION (I)       3
           0, NO PRINT, BUT GENERATE TABLE
              STNJET(XXXNNN)
          -1, PRINT OUTPUT IN DB UNITS, BUT DO NOT
              GENERATE TABLE STNJET(XXXNNN)
          -2, PRINT OUTPUT IN DIMENSIONLESS FORM,
              BUT DO NOT GENERATE TABLE
              STNJET(XXXNNN)
          -3, BOTH OPTIONS -1 AND -2
           1, PRINT OUTPUT IN DB UNITS AND GENERATE
              TABLE STNJET(XXXNNN)
           2, PRINT OUTPUT IN DIMENSIONLESS FORM
              AND GENERATE TABLE STNJET(XXXNNN)
           3, BOTH OPTIONS 1 AND 2
IPRINT    PRINT OPTION CODE (I)                          3
          0    NO PRINT DESIRED
          1    INPUT PRINT ONLY
          2    OUTPUT PRINT ONLY
          3    BOTH INPUT AND OUTPUT PRINT
```

DATA BASE UNIT MEMBERS
(DESCRIBED UNDER DATA BASE STRUCTURES)
SFIELD(FREQ)
SFIELD(THETA)
SFIELD(PHI)
STNTBL(SDF)
STNTBL(JDF)
STNTBL(FSP)

OUTPUT

USER PARAMETERS

RS   DISTANCE FROM NOZZLE EXIT TO PSUEDO-OBSERVER

SYSTEM PARAMETER

```
NERR      .TRUE.  - IMPLIES AN ERROR WAS ENCOUNTERED
                    DURING MODULE EXECUTION
          .FALSE. - NO ERROR ENCOUNTERED
```

Stone Jet Noise Module (STNJET)

DATA BASE UNIT MEMBERS

```
  STNJET(XXXNNN)  SEE FORMAT UNDER DATA BASE STRUCTURES.
                  NOTE MEMBER NAME XXXNNN IS FORMED FROM
                  USER PARAMETERS SCRXXX AND SCRNNN.
                  OUTPUT OF THIS TABLE IS CONTROLLED BY
                  USER PARAMETER IOUT.
```

DATA BASE STRUCTURES

```
  SFIELD( FREQ )  1 RECORD MEMBER IN *RS FORMAT CONTAINING
                  VALUES OF 1/3 OCTAVE BAND CENTER FREQUENCIES
                  IN HZ
  SFIELD( PHI )   1 RECORD MEMBER IN *RS FORMAT CONTAINING
                  VALUES OF AZIMUTHAL DIRECTIVITY ANGLE
                  IN DEG
  SFIELD(THETA)   1 RECORD MEMBER IN *RS FORMAT CONTAINING
                  VALUES OF POLAR DIRECTIVITY ANGLE
                  IN DEG
  STNJET(XXXNNN)  TYPE 1 TABLE CONTAINING MEAN
                  SQUARE PRESSURE AS A FUNCTION OF
                  (1) FREQUENCY, (2) DIRECTIVITY ANGLE,
                  AND (3) AZIMUTHAL ANGLE
  STNTBL(JDF)     TYPE 1 TABLE CONTAINING THE JET MIXING
                  NOISE SPECTRAL DISTRIBUTION FACTOR AS A
                  FUNCTION OF LOG10 OF THE STROUHAL NUMBER AND
                  THE MODIFIED DIRECTIVITY ANGLE
  STNTBL(SDF)     TYPE 1 TABLE CONTAINING SHOCK NOISE SPECTRAL
                  DISTRIBUTION FACTOR AS A FUNCTION OF LOG10
                  OF THE SHOCK NOISE FREQUENCY PARAMETER
  STNTBL(FSP)     TYPE 1 TABLE CONTAINING THE FREQUENCY SHIFT
                  PARAMETER AS A FUNCTION OF THE LOG OF THE
                  AREA RATIO PARAMETER AND OF THE VELOCITY
                  RATIO
```

ERRORS

NON-FATAL

1. INSUFFICIENT LOCAL DYNAMIC STORAGE.
2. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER.
3. SHOCK CELL NOISE CANNOT BE COMPUTED WITH PRIMARY STREAM MACH NUMBER .LE. ONE.
4. COAXIAL JET NOISE CANNOT BE COMPUTED WHEN SECONDARY JET VELOCITY IS GREATER THAN PRIMARY JET VELOCITY.
5. INTERPOLATION ERROR ENCOUNTERED BY SUBPROGRAM TMTERP.
6. USER PARAMETER VALUE OUT OF RANGE. DEFAULT VALUE WILL BE USED.

FATAL - NONE

Stone Jet Noise Module (STNJET)

```
REMARKS
  REFERENCES
    1.  STONE, J.R.: "INTERIM PREDICTION METHOD FOR JET
        NOISE", NASA TM X-71618, 1974.
    2.  STONE, J.R., AND MONTEGANI, F.J.:  "AN IMPROVED
        PREDICTION METHOD FOR THE NOISE GENERATED
        IN FLIGHT BY CIRCULAR JETS", NASA TM-81470,
        APRIL, 1980.

LDS REQUIREMENTS
  LENGTH = ( NFREQ * NTHETA * NPHI ) + ( NTHETA * NPHI ) +
             NFREQ + NTHETA + NPHI

GDS REQUIREMENTS
  SUFFICIENT ALLOCATION FOR THE FOLLOWING TABLES
    STNJET(XXXNNN)
    STNTBL(JDF)
    STNTBL(SDF)
    STNTBL(FSP)
```

4.8.7 Dual Stream Coannular Jet Noise Module (CNLJET)

PURPOSE - CNLJET PREDICTS THE NOISE CHARACTERISTICS OF A COANNULAR JET EXHAUST NOZZLE WITH AN INVERTED VELOCITY PROFILE

AUTHOR - WKB(L03/00/00)
MMF(L03/00/09)

INPUT

USER PARAMETERS

		TYPE	DEFAULT (SI)
AE	ENGINE REFERENCE AREA, M**2 (FT**2)	RS	PI/4
NENG	NUMBER OF ENGINES	I	1
RS	DISTANCE FROM NOZZLE EXIT TO OBSERVER, M (FT)	RS	SQRT(AE)
A1	PRIMARY JET AREA DIMENSIONLESS RE AE	RS	1.0
T1	PRIMARY JET TOTAL TEMPERATURE DIMENSIONLESS RE AMBIENT TEMPERATURE	RS	1.0
V1	PRIMARY JET VELOCITY DIMENSIONLESS RE CA	RS	1.0
RHO1	PRIMARY JET DENSITY DIMENSIONLESS RE RHOA	RS	1.0
GAMMA1	RATIO OF SPECIFIC HEATS FOR PRIMARY JET	RS	1.4
A2	SECONDARY JET AREA DIMENSIONLESS RE AE	RS	.0
DH2	SECONDARY JET HYDRAULIC DIAMETER DIMENSIONLESS RE SQRT(AE)	RS	1.0
T2	SECONDARY JET TOTAL TEMPERATURE DIMENSIONLESS RE AMBIENT TEMPERATURE	RS	1.0
V2	SECONDARY JET VELOCITY DIMENSIONLESS RE CA	RS	.0
RHO2	SECONDARY JET DENSITY DIMENSIONLESS RE RHOA	RS	1.0
GAMMA2	RATIO OF SPECIFIC HEATS FOR SECONDARY JET	RS	1.4
CA	AMBIENT SPEED OF SOUND, M/S (FT/S)	RS	340.294

MA	AIRCRAFT MACH NUMBER	RS	.0
STIME	SOURCE TIME AT WHICH NOISE IS CALCULATED	RS	.0
RHOA	AMBIENT DENSITY, KG/M**3 (SLUG/FT**3)	RS	1.225
SCRXXX	THREE LETTER CODE XXX USED TO FORM TABLE UNIT MEMBER NAME CNLJET(XXXNNN). SEE USER PARAMETER SCRNNN	A	3HXXX
SCRNNN	INTEGER VALUE, NNN .GT. 0, USED TO FORM TABLE UNIT MEMBER NAME CNLJET(XXXNNN). SEE USER PARAMETER SCRXXX	I	001
IOUT	OUTPUT CODE FOR TABLE AND/OR PRINTED OUTPUT 0 NO PRINT BUT GENERATE TABLE CNLJET(XXXNNN) -1 PRINT OUTPUT IN DB UNITS BUT DO NOT GENERATE TABLE CNLJET(XXXNNN) -2 PRINT OUTPUT IN DIMENSIONLESS UNITS BUT DO NOT GENERATE TABLE CNLJET(XXXNNN) -3 BOTH OPTIONS -1 AND -2 1 PRINT OUTPUT IN DB UNITS AND GENERATE TABLE CNLJET(XXXNNN) 2 PRINT OUTPUT IN DIMENSIONLESS UNITS AND GENERATE TABLE CNLJET(XXXNNN) 3 BOTH OPTIONS 1 AND 2	I	3
IPRINT	PRINT OPTION CODE 0 NO PRINT DESIRED 1 PRINT INPUTS ONLY 2 PRINT OUTPUTS ONLY 3 BOTH OPTIONS 1 AND 2	I	3
IUNITS	=2HSI INPUTS ARE IN SI UNITS =7HENGLISH, INPUTS ARE IN ENGLISH UNITS	A	2HSI

DATA BASE UNITS AND MEMBERS

SFIELD(FREQ)	SFIELD(THETA)	SFIELD(PHI)
JWRCOAN(OM)	JWRCOAN(PDF)	JWRCOAN(S1)
JWRCOAN(S2)	JWRCOAN(NSF)	JWRCOAN(CBF)
JWRCOAN(ALPHA)	JWRCOAN(DIR)	JWRCOAN(MTH)

(FORMATS ARE DESCRIBED UNDER DATA BASE STRUCTURES)

OUTPUT

USER PARAMETERS

```
  RS          DISTANCE FROM SOURCE TO OBSERVER,
              M (FT)
  CNLERR      =0, NO ERRORS OCCURRED
              =1, MEMBER MANAGER ERROR OCCURRED ON OPEN
              =2, INSUFFICIENT LDS AVAILABLE
              =3, MEMBER MANAGER ERROR OCCURRED ON READ
              =4, AT LEAST ONE USER PARAMETER HAS AN INVALID
                  VALUE
              =5, UNABLE TO BUILD TABLE CNLJET(XXXNNN)
              =6, AN INVALID VALUE WAS CALCULATED DURING
                  EXECUTION
```

SYSTEM PARAMETERS

```
  NERR        SYSTEM ERROR PARAMETER
              =.TRUE. , ERROR OCCURRED DURING EXECUTION
                        (CNLERR = 1, 2, 3, 5)
              =.FALSE., NO ERROR OCCURRED
```

DATA BASE UNITS AND MEMBERS

```
  CNLJET(XXXNNN)  SEE FORMAT DESCRIPTIONS UNDER DATA BASE
                  STRUCTURES.  NOTE MEMBER NAME XXXNNN
                  IS FORMED FROM USER PARAMETERS SCRNNN AND
                  SCRXXX.  OUTPUT OF THIS TABLE IS CONTROLLED
                  BY USER PARAMETER IOUT
```

DATA BASE STRUCTURES

```
  JWRCOAN(OM)     TYPE 1 TABLE OF JET VARIABLE DENSITY
                  EXPONENT AS A FUNCTION OF LOG10 OF THE
                  OF RATIO OF JET VELOCITY TO THE SPEED
                  OF SOUND
  JWRCOAN(PDF)    TYPE 1 TABLE OF LOG10 OF POWER
                  DEVIATION FACTOR OF SAE METHOD FROM
                  V**8 LAW AS A FUNCTION OF LOG10 OF
                  RATIO OF JET VELOCITY TO THE SPEED
                  OF SOUND
  JWRCOAN(S1)     TYPE 1 TABLE OF LOG10 OF PEAK
                  STROUHAL NUMBER FOR FIRST COMPONENT
                  OF SPECTRUM AS A FUNCTION OF (1)
                  DIRECTIVITY ANGLE IN DEGREES AND (2)
                  LOG10 OF THE RATIO OF THE JET VELOCITY
                  TO THE SPEED OF SOUND
  JWRCOAN(S2)     TYPE 1 TABLE OF PEAK STROUHAL NUMBER
                  FOR THE SECOND COMPONENT OF SPECTRUM
                  AS A FUNCTION OF (1) DIRECTIVITY
```

	ANGLE IN DEGREES, (2) LOG10 OF THE RATIO OF THE JET VELOCITY TO THE SPEED OF SOUND
JWRCOAN(NSF)	TYPE 1 TABLE OF LOG10 OF NORMALIZED SPECTRUM FUNCTION AS A FUNCTION OF (1) LOG10 OF THE STROUHAL NUMBER AND (2) DIRECTIVITY ANGLE IN DEG
JWRCOAN(ALPHA)	TYPE 1 TABLE OF LOG10 OF RATIO OF PEAK STROUHAL NUMBER FOR SECOND COMPONENT TO THAT OF FIRST COMPONENT AS A FUNCTION OF (1) VELOCITY RATIO V2/V1, (2) LOG10 OF THE RATIO OF THE JET VELOCITY TO THE SPEED OF SOUND, AND (3) DIRECTIVITY ANGLE IN DEG
JWRCOAN(CBF)	TYPE 1 TABLE OF OVERALL POWER LEVEL BENEFIT FOR COANNULAR NOZZLE IN DB AS A FUNCTION OF (1) THE VELOCITY RATIO V2/V1 AND (2) LOG10 OF THE RATIO OF JET VELOCITY TO THE SPEED OF SOUND
JWRCOAN(DIR)	TYPE 1 TABLE OF 10*LOG10 OF NORMALIZED DIRECTIVITY FUNCTION FOR COANNULAR JETS AS A FUNCTION OF (1) DIRECTIVITY ANGLE IN DEGREES AND (2) LOG10 OF THE RATIO OF THE JET VELOCITY TO THE SPEED OF SOUND
JWRCOAN(MTH)	TYPE 1 TABLE OF FORWARD FLIGHT EFFECT CORRECTION (JET VELOCITY EXPONENT) AS A FUNCTION OF DIRECTIVITY ANGLE IN DEG
SFIELD(FREQ)	1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF 1/3 OCTAVE BAND CENTER FREQUENCIES IN HZ
SFIELD(THETA)	1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF POLAR DIRECTIVITY ANGLE IN DEG
SFIELD(PHI)	1 RECORD MEMBER IN *RS FORMAT CONTAINING VALUES OF AZIMUTHAL DIRECTIVITY ANGLE IN DEG
CNLJET(XXXNNN)	TYPE 1 TABLE OF MEAN SQUARE ACOUSTIC PRESSURE DATA AS A FUNCTION OF (1) FREQUENCY, (2) DIRECTIVITY ANGLE, AND (3) AZIMUTH ANGLE

ERRORS

NON-FATAL

1. INSUFFICIENT SPACE IN LDS
2. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT MEMBER
3. USER PARAMETER VALUE OUT OF RANGE - DEFAULT TO BE USED
4. UNABLE TO INTERPOLATE SPECIFIED TABLE
5. SECONDARY JET VELOCITY IS LESS THAN THE PRIMARY JET VELOCITY

4.8.8 Airframe Noise Module (FNKAFM)

PURPOSE - TO PREDICT THE BROADBAND NOISE FOR THE DOMINANT COMPONENTS OF THE AIRFRAME BY THE FINK METHOD
REFERENCE - FEDERAL AVIATION ADMINISTRATION REPORT NO. FAA-RD-77-28, MARCH 1977, AIRFRAME NOISE PREDICTION METHOD, M. R. FINK

AUTHOR - CBF(L03/00/00)
MMF(L03/00/09)

INPUT

USER PARAMETERS		DEFAULT SI UNITS
AF	- FLAP AREA (RS), M**2 (FT ** 2)	10.
AH	- HORIZONTAL TAIL AREA (RS), M**2 (FT ** 2)	20.
AV	- VERTICAL TAIL AREA (RS), M**2 (FT ** 2)	20.
AW	- WING AREA (RS), M**2 (FT ** 2)	100.
BF	- FLAP SPAN (RS), M (FT)	5.
BH	- HORIZONTAL TAIL SPAN (RS), M (FT)	10.
BV	- VERTICAL TAIL SPAN (RS), M (FT)	10.
BW	- WING SPAN (RS), M (FT)	20.
DELTAF	- FLAP SETTING (RS), DEG	0.
DMG	- TIRE DIAMETER OF MAIN LANDING GEAR (RS), M (FT)	1.
DNG	- TIRE DIAMETER OF NOSE LANDING GEAR (RS), M (FT)	1.
LMG	- MAIN LANDING GEAR STRUT LENGTH (RS), M (FT)	3.
LNG	- NOSE LANDING GEAR STRUT LENGTH (RS), M (FT)	3.
CA	- AMBIENT SPEED OF SOUND (RS), M/S (FT/S)	340.294
RHOA	- AMBIENT DENSITY (RS), KG/M**3 (SLUG/FT ** 3)	1.225
MUA	- AMBIENT DYNAMIC VISCOSITY (RS), KG/(M S) (SLUG/(FT S))	1.7894 *10**-5
RS	- DISTANCE FROM SOURCE TO OBSERVER (RS), M (FT)	BW
MA	- AIRCRAFT MACH NUMBER (RS)	.3
STIME	- SOURCE TIME (RS)	0.
NWMG	- NUMBER OF WHEELS PER MAIN LANDING GEAR (I)	4
NWNG	- NUMBER OF WHEELS PER NOSE LANDING GEAR (I)	2
NMG	- NUMBER OF MAIN LANDING GEAR (I)	2
NNG	- NUMBER OF NOSE LANDING GEAR (I)	1
NS	- NUMBER OF SLOTS FOR TRAILING EDGE FLAPS (I)	3
IPRINT	- PRINTED OUTPUT CODE (I) =0, NO PRINT =1, INPUT PRINT ONLY =2, OUTPUT PRINT ONLY =3, BOTH INPUT AND OUTPUT PRINT	3

Airframe Noise Module (FNKAFM)

```
IOUT    - TABLE AND PRINT OUTPUT OPTION (I)                  3
          = 0, NO PRINT BUT GENERATE TABLE
               FNKAFM(XXXNNN)
          =-1, PRINT OUTPUT IN DB UNITS, BUT DO NOT
               GENERATE TABLE FNKAFM(XXXNNN)
          =-2, PRINT OUTPUT IN DIMENSIONLESS FORM, BUT
               DO NOT GENERATE TABLE FNKAFM(XXXNNN)
          =-3, BOTH OPTIONS -1 AND -2
          = 1, PRINT OUTPUT IN DB UNITS AND GENERATE
               TABLE FNKAFM(XXXNNN)
          = 2, PRINT OUTPUT IN DIMENSIONLESS FORM AND
               GENERATE TABLE FNKAFM(XXXNNN)
          = 3, BOTH OPTIONS 1 AND 2
SCRNNN  - INTEGER VALUE, NNN .GT. 0 USED TO FORM             001
          TABLE UNIT MEMBER FNKAFM(XXXNNN)
DYNCLN  - DESCRIPTION OF AIRCRAFT                            F
          .TRUE.,  AERODYNAMICALLY CLEAN AIRCRAFT
          .FALSE., NON-AERODYNAMICALLY CLEAN AIRCRAFT
DELTAW  - TYPE OF WING PLAN FORM                             F
          .TRUE.,  DELTA-WINGED AIRCRAFT
          .FALSE., RECTANGULAR-WINGED AIRCRAFT
LANDG   - LANDING GEAR POSITION                              4HDOWN
          4HDOWN, INDICATES LANDING GEAR IS DOWN
          4HUP  , INDICATES LANDING GEAR IS UP
SCRXXX  - THREE LETTER CODE XXX USED TO FORM                 3HXXX
          TABLE UNIT MEMBER NAME FNKAFM(XXXNNN)
IUNITS  - INPUT UNITS FLAG                                   2HSI
          =2HSI,      SI UNITS
          =7HENGLISH, ENGLISH UNITS

( THE NEXT SEVEN CODES HAVE THE FOLLOWING VALUES )
( .FALSE., DO NOT INCLUDE IN NOISE PREDICTION    )
( .TRUE.,  INCLUDE IN TOTAL PREDICTION           )
TEWN    - TRAILING EDGE WING NOISE                           T
TEHTN   - TRAILING EDGE HORIZONTAL TAIL NOISE                T
TEVTN   - TRAILING EDGE VERTICAL TAIL NOISE                  T
TEFN    - TRAILING EDGE FLAP NOISE                           F
LESN    - LEADING EDGE SLAT NOISE                            F
MLGN    - MAIN LANDING GEAR NOISE                            T
NLGN    - NOSE LANDING GEAR NOISE                            T
DATA BASE UNIT MEMBERS
(DESCRIBED UNDER DATA BASE STRUCTURES)
  SFIELD(FREQ)
  SFIELD(THETA)
  SFIELD(PHI)
```

```
OUTPUT
  USER PARAMETERS
    RS     - DISTANCE FROM SOURCE TO OBSERVER
  SYSTEM PARAMETERS
    NERR   - ERROR FLAG
             .TRUE.,  IMPLIES AN ERROR OCCURRED DURING
                      MODULE EXECUTION
             .FALSE., NO ERROR ENCOUNTERED
  DATA BASE UNIT MEMBERS
    FNKAFM(XXXNNN) - SEE FORMAT UNDER DATA BASE STRUCTURES.
                     NOTE MEMBER NAME XXXNNN IS FORMED FROM
                     USER PARAMETERS SCRXXX AND SCRNNN. OUTPUT
                     OF THIS TABLE IS CONTROLLED BY USER
                     PARAMETER, IOUT.

DATA BASE STRUCTURES
  SFIELD(FREQ)     - 1 RECORD MEMBER IN *RS FORMAT CONTAINING
                     VALUES OF 1/3 OCTAVE BAND CENTER
                     FREQUENCIES IN HZ
  SFIELD(THETA)    - 1 RECORD MEMBER IN *RS FORMAT CONTAINING
                     VALUES OF THE POLAR DIRECTIVITY ANGLE IN
                     DEG
  SFIELD(PHI)      - 1 RECORD MEMBER IN *RS FORMAT CONTAINING
                     VALUES OF THE AZIMUTHAL DIRECTIVITY ANGLE
                     IN DEG
  FNKAFM(XXXNNN)   - TYPE 1 TABLE CONTAINING MEAN SQUARE
                     ACOUSTIC PRESSURE AS A FUNCTION OF
                     (1) FREQUENCY, (2) DIRECTIVITY ANGLE, AND
                     (3) AZIMUTHAL ANGLE

ERRORS
  NON-FATAL
    1. INSUFFICIENT LOCAL DYNAMIC STORAGE.
    2. MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT
       MEMBER.
  FATAL - NONE

REMARKS
  REFERENCES
    FINK, M. R., AIRFRAME NOISE PREDICTION METHOD, FEDERAL
      AVIATION ADMINISTRATION REPORT NO. FAA-RD-77-28, MARCH
      1977.

LDS REQUIREMENTS
  LENGTH = ( NFREQ * NTHETA * NPHI) + ( NTHETA * NPHI ) +
             NFREQ + NTHETA + NPHI
      WHERE :
        NFREQ  = NUMBER OF FREQUENCY VALUES
        NTHETA = NUMBER OF POLAR DIRECTIVITY ANGLES
        NPHI   = NUMBER OF AZIMUTHAL DIRECTIVITY ANGLES
```

Airframe Noise Module (FNKAFM)

GDS REQUIREMENTS
SUFFICIENT ALLOCATION FOR TABLE FNKAFM(XXXNNN)

4.8.9 Smith and Bushell Turbine Noise Module (SMBTUR)

PURPOSE - TO PREDICT THE BROADBAND NOISE FOR THE 'VORTEX' COMPONENT OF AN AXIAL FLOW TURBINE BY THE SMITH AND BUSHELL METHOD.
REFERENCE - ASME PAPER 69-WA/GT-12, NOVEMBER 1969, TURBINE NOISE - ITS SIGNIFICANCE IN THE CIVIL AIRCRAFT NOISE PROBLEM, J. J. T. SMITH AND K. W. BUSHELL

AUTHOR - CBF(L03/00/00)
MMF(L03/00/09)

INPUT

USER PARAMETERS

		DEFAULT SI UNITS
AE	- ENGINE REFERENCE AREA (RS), M**2 (FT**2)	PI/4
AREA	- TURBINE INLET CROSS-SECTIONAL AREA (RS), RE AE	1.
C	- ROTOR BLADE MEAN AXIAL CHORD OF THE LAST STAGE (RS), RE SQRT(AE)	1.
CA	- AMBIENT SPEED OF SOUND (RS), M/S (FT/S)	340.294
FA	- FUEL-TO-AIR RATIO (RS)	0.
HA	- ABSOLUTE HUMIDITY (RS), PERCENT MOLE FRACTION	0.
IOUT	- TABLE AND PRINT OUTPUT OPTION (I) 0, NO PRINT, BUT GENERATE TABLE SMBTUR(XXXNNN) -1, PRINT OUTPUT IN DB UNITS, BUT DO NOT GENERATE TABLE SMBTUR(XXXNNN) -2, PRINT OUTPUT IN DIMENSIONLESS FORM, BUT DO NOT GENERATE TABLE SMBTUR(XXXNNN) -3, BOTH OPTIONS 1, PRINT OUTPUT IN DB UNITS AND GENERATE TABLE SMBTUR(XXXNNN) 2, PRINT OUTPUT IN DIMENSIONLESS FORM AND GENERATE TABLE SMBTUR(XXXNNN) 3, BOTH OPTION 1 AND 2	3
IPRINT	- PRINT FLAG (I) 0, NO PRINT DESIRED 1, INPUT PRINT ONLY 2, OUTPUT PRINT ONLY 3, BOTH INPUT AND OUTPUT PRINT	3
IUNITS	- INPUT UNITS FLAG 2HSI, SI UNITS 7HENGLISH, ENGLISH UNITS	2HSI
MA	- AIRCRAFT MACH NUMBER (RS)	0.
MDOT	- CORE MASS FLOW RATE (RS), RE RHOA*CA*AE	.2
N	- ROTATIONAL SPEED (RS), RE CA/D	.3

```
      NENG    - NUMBER OF ENGINES                                    1
      NS      - NUMBER OF TURBINE STAGES                             1
      RHOA    - AMBIENT DENSITY (RS), KG/M**3
                (SLUG/FT**3)
      RS      - DISTANCE FROM SOURCE TO PSEUDO OBSERVER            1.225
                (RS), M (FT)
      SCRNNN  - INTEGER VALUE, NNN, .GT. 0 USED TO FORM             001
                TABLE UNIT MEMBER NAME SMBTUR(XXXNNN)
      SCRXXX  - THREE LETTER CODE, XXX, USED TO FORM                3HXXX
                TABLE UNIT MEMBER NAME SMBTUR(XXXNNN)
      STIME   - SOURCE TIME                                         0.
      TSJ     - EXIT STATIC TEMPERATURE (RS), RE TA                 2.

    DATA BASE UNIT MEMBERS
      (DESCRIBED UNDER DATA BASE STRUCTURES)
      SFIELD(FREQ)
      SFIELD(THETA)
      SFIELD(PHI)

  OUTPUT
    USER PARAMETERS
      RS      - DISTANCE FROM SOURCE TO OBSERVER

    SYSTEM PARAMETERS
      NERR    - ERROR FLAG
                .TRUE.,  IMPLIES AN ERROR WAS ENCOUNTERED
                         DURING MODULE EXECUTION
                .FALSE., NO ERROR ENCOUNTERED
    DATA BASE UNIT MEMBERS
      SMBTUR(XXXNNN)  SEE FORMAT UNDER DATA BASE STRUCTURES.
                      NOTE MEMBER NAME XXXNNN IS FORMED FROM
                      USER PARAMETERS SCRXXX AND SCRNNN.  OUTPUT
                      OF THIS TABLE IS CONTROLLED BY USER
                      PARAMETER, IOUT.

  DATA BASE STRUCTURES
    SFIELD(FREQ)      1 RECORD MEMBER IN *RS FORMAT CONTAINING
                      VALUES OF 1/3 OCTAVE BAND CENTER
                      FREQUENCIES IN HZ
    SFIELD(THETA)     1 RECORD MEMBER IN *RS FORMAT CONTAINING
                      VALUES OF THE POLAR DIRECTIVITY ANGLE IN
                      DEG
    SFIELD(PHI)       1 RECORD MEMBER IN *RS FORMAT CONTAINING
                      VALUES OF THE AZIMUTHAL DIRECTIVITY ANGLE
                      IN DEG
    SMBTUR(XXXNNN)    TYPE 1 TABLE CONTAINING MEAN SQUARE
                      ACOUSTIC PRESSURE AS A FUNCTION OF
                      (1) FREQUENCY, (2) DIRECTIVITY ANGLE,
                      AND (3) AZIMUTHAL ANGLE
```

```
ERRORS
  NON-FATAL
    1.  INSUFFICIENT LOCAL DYNAMIC STORAGE.
    2.  MEMBER MANAGER ERROR OCCURRED ON SPECIFIED UNIT
        MEMBER.
  FATAL - NONE

REMARKS
  REFERENCES
    SMITH, J. J. T., AND BUSHELL, K. W., TURBINE NOISE - ITS
        SIGNIFICANCE IN THE CIVIL AIRCRAFT NOISE PROBLEM,
        ASME PAPER 69-WA/GT-12, NOVEMBER 1969.

LDS REQUIREMENTS
  LENGTH = ( NFREQ * NTHETA * NPHI ) + ( NTHETA * NPHI )
         + NFREQ + NTHETA   NPHI
      WHERE
        NFREQ  = NUMBER OF FREQUENCY VALUES
        NTHETA = NUMBER OF DIRECTIVITY ANGLES
        NPHI   = NUMBER OF AZIMUTHAL ANGLES

GDS REQUIREMENTS
  SUFFICIENT ALLOCATION FOR TABLE SMBTUR(XXXNNN)
```

5 ANOPP CONTROL STATEMENT PROCEDURE LIBRARY

5.1 OVERVIEW

The ANOPP control statement procedure library is a set of ANOPP data members containing commonly used sequences of control statements. It is stored on the ANOPP permanent data base as data unit PROCLIB and is available for use in any ANOPP run. Use of the procedure library enables the user to save time by reducing the number of user supplied control statements.

There are four categories of procedures contained in the library. They can be identified by the format of the member name. The four categories are as follows:

1. Noise prediction with dimensional inputs. These procedures take the input noise prediction parameters, convert the input parameters to the proper dimensionless form for the noise prediction module, and execute the module. The member names for these procedures are of the form DXXXXXX, where D signifies dimensional input and XXXXXX presents the functional module name. For example, to execute the Fan Noise Module (HDNFAN) with dimensional inputs, procedure DHDNFAN should be used.

2. Time-dependent noise prediction. These procedures predict noise for time-dependent problems with the input data on flight-dynamics and engine-state tables. First, the appropriate source noise parameters module is executed to determine the prediction parameters and the number of source times. Then, the module is executed once for each source time, producing a set of tables of the source noise as a function of time. The member names for these procedures are of the form TXXXXXX, where T signifies time dependent and XXXXXX presents the noise prediction functional module. For example, to execute the Single Stream Circular Jet Noise Module (SGLJET) for time-dependent predictions, procedure TSGLJET should be used.

3. Module-sequencing procedures. These procedures execute a series of functional modules that are normally executed in a fixed order. The required transfer of data base members and parameters from one module to the next is handled automatically for the user. The member names for these procedures are of the form XXXYYY, where XXX is a three-letter code for the leading functional module and YYY is a three-letter code describing the function of the procedure. For example, the procedure PROSGL propagates a single noise source to the observer using the Propagation Module (PRO), the Noise Levels Module (LEV), and the Effective Noise Module (EFF).

4. Internal library procedures. These procedures are used internally within the library to perform various tasks, particularly name overrides. They are used from within other procedures and are not called directly by the user. The member names for these procedures are of the form XYYYYYY, where X denotes an internal library procedure and YYYYYY is an associated module name. For example, the procedure DGETUR calls procedure XGETUR to execute the functional module GETUR.

ANOPP procedures from the library are executed using the CALL CS. As discussed in section 3.9.2, the CALL CS transfers run processing control to the library procedure. It allows for name substitutions for the unit members listed as inputs and outputs in section 5.2. The capability for parameter-name substitution by the user does not exist for the procedure library.

As an example of ANOPP procedure usage, a request to produce the atmo sphere and atmospheric absorption tables for a constant, standard-sea-level atmosphere using procedure ATMDAT results in the following:

```
  :
CREATE ATMCON $
UPDATE NEWU=ATMCON SOURCE=* $
  -ADDR OLDM=IN FORMAT=4H3RS$ $
     0. 288.15  0.  $
END* $
PARAM IPRINT=3 $
 SETSYS JECHO=.FALSE. JLOG=.FALSE. $
CALL PROCLIB(ATMDAT) ATM=ATMCON $
SETSYS JECHO=.TRUE. JLOG=.TRUE. $
```

Note the following three aspects of this example:

1. Since a function of ATMDAT is to provide the SFIELD array, the user is not required to provide it.

2. The SETSYS before the CALL CS eliminates the print caused by the secondary edit phase and the execution of the procedure member.

3. The input and output members for the procedure are on unit ATMCON since this name is provided on the CALL CS.

5.2 PROCEDURE DESCRIPTIONS

This section contains the prologues which describe the ANOPP procedures in the library. Each prologue is headed by the name of the procedure it describes. Figure 1 shows the format for the prologues.

```
****PROCEDURE - name
    PURPOSE - short description of the procedure (1-2 sentences)
    AUTHOR -  initials and level number, such as L01/00/00
    INPUT                                                        MODULES
      USER PARAMETERS
          Name1 - description - default value -                  referencing modules
          .
          .
          .
          Namen - description - default value -                  referencing modules
      MEMBERS AND TABLES                                         MODULES
          DATA UNIT(DATA MEMBER) - description                   referencing modules
    OUTPUT
      USER PARAMETERS                                            MODULES
          Same as for INPUT
      MEMBERS AND TABLES                                         MODULES
          Same as for INPUT
    LOCAL
      USER PARAMETERS                                            MODULES
          Same as for INPUT
      MEMBERS AND TABLES
          Same as for INPUT
    FUNCTIONS - description of the basic functions performed by the procedure in
                the order they are performed.
    PROCEDURES CALLED - list of the procedures called by this procedure.
    CALLING PROCEDURES - list of other procedures that call this procedure.
```

Figure 1.- ANOPP procedure prologue format.

****PROCEDURE ATMDAT

PURPOSE - TO GENERATE THE ATMOSPHERIC DATA AND ATMOSPHERIC ABSORPTION DATA FOR A USER SPECIFIED TEMPERATURE PROFILE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT
USER PARAMETERS

USER PARAMETERS		MODULES
GRNALT	GROUND LEVEL ALTITUDE (M),(FT) (RS) (DEFAULT=0.)	ATM
GRNPRES	ATMOSPHERIC PRESSURE AT GROUND LEVEL (N/M**2),(LB/FT**2) (RS) (DEFAULT= 101325. N/M**2)	ATM
NALT	NUMBER OF ALTITUDES FOR OUTPUT (I) (DEFAULT=10)	IF,ATM, PARAM
DELALT	ALTITUDE INCREMENT FOR OUTPUT (M),(FT) (RS) (DEFAULT=100, M)	ATM
ATMPRT	ATM OUTPUT PRINT OPTION (I) =0 NO PRINT (DEFAULT) =1 INPUT PRINT ONLY =2 OUTPUT PRINT ONLY =3 BOTH INPUT AND OUTPUT PRINT	ATM
ABSPRT	ABS OUTPUT PRINT OPTION (I) =0 NO PRINT (DEFAULT) =1 INPUT PRINT ONLY =2 OUTPUT PRINT ONLY =3 BOTH INPUT AND OUTPUT PRINT	ABS
ATMOS	ATMOSPHERE OPTION (L) =.TRUE. - GENERATE ATMOSPHERIC TABLE (DEFAULT) =.FALSE.- NO TABLE NOTE: IF ATMOS=.FALSE. AND ABSORP=.TRUE., THEN USER MUST PROVIDE ATM(TMOD).	IF, PARAM
ABSORP	ATMOSPHERIC ABSORPTION OPTION (L) =.TRUE. - GENERATE ATMOSPHERIC ABSORPTION TABLE (DEFAULT) =.FALSE.- NO TABLE	IF, PARAM
IUNITS	INPUT UNITS OPTION (A) =2HSI - INPUT IN SI UNITS (DEFAULT) =7HENGLISH- INPUT IN ENGLISH UNITS	IF, PARAM

PROCEDURE PROCLIB (ATMDAT)

```
  MEMBERS                                                        MODULES

    ATM    (IN  ) MULTI-RECORD FORMAT IN 3RS FORMAT OF   ATM
                  TEMPERATURE AND HUMIDITY PROFILE:
                  (NOT REQUIRED IF ATMOS=.FALSE.)

                  RECORD WORD DESCRIPTION

                     1        FORMAT=3RS
                           1  ALTITUDE ABOVE SEA LEVEL
                           2  TEMPERATURE
                           3  PERCENT RELATIVE HUMIDITY
                     2        (SAME AS 1)
                     .
                     .
                     .
                     N        LAST ENTRY IN PROFILE

    SFIELD(FREQ) ONE-RECORD MEMBER IN *RS FORMAT        ABS
                 OF THE VALUES OF FREQUENCY IN HERTZ

OUTPUT
  USER PARAMETERS - NONE

  MEMBERS                                                        MODULES

    ATM (TMOD ) TYPE 1 DATA TABLE OF ATMOSPHERIC        ATM,ABS
                DATA AS A FUNCTION OF ALTITUDE.
    ATM (AAC  ) TYPE 1 DATA TABLE OF ATMOSPHERIC        ABS
                ABSORPTION DATA AS A FUNCTION OF
                ALTITUDE AND FREQUENCY.

LOCAL
  USER PARAMETERS                                                MODULES

    ABSINT  NUMBER OF INTEGRATION STEPS =5               ABS
    SAE     METHOD FOR ABSORPTION TABLE BASED ON PREDICTION
            PROCEDURE USED (L)
            =.TRUE. - FOR PREDICTION PROCEDURES REQUIRING
                      SAE ARP 866A ABSORPTION
            =.FALSE.- OTHERWISE (DEFAULT)

  MEMBERS                                                        MODULES

    SCRATCH (TAB1 ) TYPE 1 TABLE OF THE CONTENTS        ATM
                    OF ATM (IN ).
```

PROCEDURE PROCLIB (ATMDAT)

FUNCTIONS

1. LOAD SFIELD UNIT IF NOT ASSIGNED.
2. SET INPUT PARAMETERS FOR ATM MODULE.
3. EXECUTE ATM MODULE.
4. SET INPUT PARAMETERS FOR ABS MODULE.
5. EXECUTE ABS MODULE.

MODULES CALLED - ABS ATM

PROCEDURES CALLED - XABS XATM

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (ATMSTD)

****PROCEDURE ATMSTD

PURPOSE - TO LOAD THE STANDARD ATMOSPHERE AND STANDARD ATMOSPHERIC ABSORPTION TABLES FROM THE PERMANENT DATA BASE.

AUTHOR - DSW(L03/00/00)

INPUT - NONE

OUTPUT
USER PARAMETERS - NONE

MEMBERS		MODULES
ATM (TMOD)	TYPE 1 DATA TABLE OF STANDARD ATMOSPHERIC DATA AS A FUNCTION OF ALTITUDE.	LOAD
ATM (AAC)	TYPE 1 DATA TABLE OF STANDARD ATMOSPHERIC ABSORPTION DATA AS A FUNCTION OF ALTITUDE AND FREQUENCY	LOAD

LOCAL
USER PARAMETERS

SAE — METHOD FOR ABSORPTION TABLE BASED ON PREDICTION PROCEDURE USED (L)
=.TRUE. - FOR PREDICTION PROCEDURES REQUIRING SAE ARP 866A ABSORPTION
=.FALSE.- OTHERWISE (DEFAULT)

MEMBERS - NONE

FUNCTIONS

1. LOAD STANDARD ATMOSPHERE TABLE
2. LOAD PROPER ATMOSPHERIC ABSORPTION TABLE

MODULES CALLED - NONE

PROCEDURES CALLED - NONE

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (DCNLJET)

****PROCEDURE DCNLJET

PURPOSE - TO ALLOW EXECUTION OF THE DUAL STREAM COANNULAR JET NOISE MODULE WITH DIMENSIONAL INPUTS. USEFUL WHEN EXECUTION OF THE JET NOISE PARAMETERS MODULE IS NOT DESIRED.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)
MMF(L03/00/09)

INPUT

USER PARAMETERS

		MODULES
NENG	NUMBER OF ENGINES (I) (DEFAULT=1)	CNLJET
RS	DISTANCE FROM NOZZLE EXIT TO OBSERVER, (M),(FT) (RS) (DEFAULT=1. M)	CNLJET IF,PARAM
DELTA	ANGLE BETWEEN FLIGHT VECTOR AND ENGINE INLET AXIS (DEGREES) (RS) (DEFAULT=0.)	CNLJET
A1	PRIMARY JET AREA (M**2),(FT**2) (RS) (DEFAULT=PI/4. M**2)	CNLJET,PARAM IF,EVALUATE
T1	PRIMARY JET TOTAL TEMPERATURE (DEG K), (DEG R) (RS) (DEFAULT=288.15 DEG K)	CNLJET,PARAM IF,EVALUATE
V1	PRIMARY JET VELOCITY (M/S),(FT/S) (RS) (DEFAULT=340.294 M/S)	CNLJET,PARAM IF,EVALUATE
RHO1	PRIMARY JET DENSITY (KG/M**3), (SLUG/FT**3) (RS) (DEFAULT=1.225 KG/M**3)	CNLJET,PARAM IF,EVALUATE
GAMMA1	RATIO OF SPECIFIC HEATS FOR PRIMARY JET (RS) (DEFAULT=1.4)	CNLJET
A2	SECONDARY JET AREA (M**2),(FT**2) (RS) (DEFAULT=0.)	CNLJET,PARAM IF,EVALUATE
DH2	SECONDARY JET HYDRAULIC DIAMETER (M), (FT) (RS) (DEFAULT=SQRT(PI/4.) M)	CNLJET,PARAM IF,EVALUATE
T2	SECONDARY JET TOTAL TEMPERATURE (DEG K),(DEG R) (RS) (DEFAULT=288.15 DEG K)	CNLJET,PARAM IF,EVALUATE
V2	SECONDARY JET VELOCITY (M/S),(FT/S) (RS) (DEFAULT=0.)	CNLJET,PARAM IF,EVALUATE
RHO2	SECONDARY JET DENSITY (KG/M**3), (SLUGS/FT**3) (RS) (DEFAULT=1.225 KG/M**3)	CNLJET,PARAM IF,EVALUATE
GAMMA2	RATIO OF SPECIFIC HEATS FOR SECONDARY JET (RS) (DEFAULT=1.4)	CNLJET
CA	AMBIENT SPEED OF SOUND (M/S),(FT/S) (RS) (DEFAULT=340.294 M/S)	CNLJET,PARAM IF,EVALUATE
RHOA	AMBIENT DENSITY (KG/M**3),(SLUG/FT**3) (RS) (DEFAULT=1.225 KG/M**3)	CNLJET,PARAM IF,EVALUATE

PROCEDURE PROCLIB (DCNLJET)

```
TA      AMBIENT TEMPERATURE (DEG K),(DEG R)        IF,PARAM
        (RS) (DEFAULT=288.15 DEG K)                EVALUATE
VO      AIRCRAFT VELOCITY (M/S),(FT/S) (RS)        IF,PARAM
        (DEFAULT=0.)                               EVALUATE
STIME   SOURCE TIME AT WHICH NOISE IS CALCULATED    CNLJET
        (S) (RS) (DEFAULT=0.)
SCRXXX  THREE LETTER ID USED TO FORM TABLE NAME     CNLJET
        CNLJET(XXXNNN) (A) (DEFAULT=3HXXX)
SCRNNN  VALUE USED TO FORM TABLE NAME               CNLJET
        CNLJET(XXXNNN) (I) (DEFAULT=1)
IOUT    OUTPUT CODE FOR TABLE AND PRINTED OUTPUT    CNLJET
        (I) (DEFAULT=3)
        0 - NO PRINT BUT GENERATE CNLJET(XXXNNN).
       -1 - OUTPUT PRINT IN DB BUT NO TABLE.
       -2 - OUTPUT PRINT IN DIMENSIONLESS FORM
            BUT NO TABLE.
       -3 - OUTPUT PRINT IN DIMENSIONLESS FORM
            AND DB BUT NO TABLE.
        1 - OUTPUT PRINT IN DB AND GENERATE
            CNLJET(XXXNNN).
        2 - OUTPUT PRINT IN DIMENSIONLESS FORM
            AND GENERATE CNLJET(XXXNNN).
        3 - OUTPUT PRINT IN DIMENSIONLESS FORM
            AND DB AND GENERATE CNLJET(XXXNNN).
IPRINT  OUTPUT PRINT CODE (I) (DEFAULT=3)           CNLJET
        0 - NO PRINT
        1 - INPUT PRINT ONLY
        2 - OUTPUT PRINT ONLY
        3 - BOTH INPUT AND OUTPUT PRINT
IUNITS  UNIT OPTION CODE (A) (DEFAULT=2HSI)        IF,PARAM,
        2HSI      - SI UNITS                        CNLJET
        7HENGLISH - ENGLISH UNITS

MEMBERS                                            MODULES

SFIELD (FREQ ) - ONE RECORD MEMBER OF THE VALUES    CNLJET
                 OF FREQUENCY IN HERTZ.
SFIELD (THETA) - ONE RECORD MEMBER OF THE VALUES    CNLJET
                 OF POLAR DIRECTIVITY ANGLE IN
                 DEGREES.
SFIELD (PHI  ) - ONE RECORD MEMBER OF THE VALUES    CNLJET
                 OF AZIMUTHAL DIRECTIVITY ANGLE
                 IN DEGREES.
JWRCOAN(OM   ) - TYPE 1 DATA TABLE OF THE JET       CNLJET
                 VARIABLE DENSITY EXPONENT.
JWRCOAN(PDF  ) - TYPE 1 DATA TABLE OF THE POWER     CNLJET
                 DEVIATION FACTOR.
JWRCOAN(S1   ) - TYPE 1 DATA TABLE OF THE PEAK      CNLJET
                 STROUHAL NUMBER FOR THE FIRST
                 SPECTRUM COMPONENT.
```

PROCEDURE PROCLIB (DCNLJET)

JWRCOAN(S2)	- TYPE 1 DATA TABLE OF THE PEAK STROUHAL NUMBER FOR THE SECOND SPECTRUM COMPONENT.	CNLJET
JWRCOAN(NSF)	- TYPE 1 DATA TABLE OF THE NORMALIZED SPECTRUM FUNCTION.	CNLJET
JWRCOAN(ALPHA)	- TYPE 1 DATA TABLE OF THE RATIO OF THE SECOND COMPONENT SPECTRAL PEAK TO THE FIRST COMPONENT SPECTRAL PEAK.	CNLJET
JWRCOAN(CBF)	- TYPE 1 DATA TABLE OF THE COANNULAR BENEFIT FACTOR.	CNLJET
JWRCOAN(DIR)	- TYPE 1 DATA TABLE OF THE NORMALIZED DIRECTIVITY FUNCTION.	CNLJET
JWRCOAN(MTH)	- TYPE 1 DATA TABLE OF THE FORWARD VELOCITY INDEX.	CNLJET

OUTPUT

USER PARAMETERS		MODULES
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS)	CNLJET
CNLERR	ERROR PARAMETER (I) 0 - NO ERRORS ENCOUNTERED 1 - MEMBER MANAGER ERROR OCCURRED ON OPEN. 2 - INSUFFICIENT LDS AVAILABLE. 3 - MEMBER MANAGER ERROR OCCURRED ON READ. 4 - AT LEAST ONE USER PARAMETER HAS AN INVALID VALUE. 5 - UNABLE TO BUILD TABLE CNLJET(XXXNNN). 6 - AN INVALID VALUE WAS CALCULATED DURING EXECUTION.	CNLJET

MEMBERS		MODULES
CNLJET(XXXNNN)	- TYPE 1 DATA TABLE OF THE MEAN-SQUARE ACOUSTIC PRESSURE.	CNLJET

LOCAL

USER PARAMETERS		MODULES
MA	AIRCRAFT MACH NUMBER (RS)	CNLJET, EVALUATE
AE	ENGINE REFERENCE AREA (M**2),(FT**2) (RS)	PARAM,CNLJET EVALUATE
TEMP	9 ELEMENT PARAMETER FOR SAVING INPUT VALUES (RS)	PARAM, EVALUATE

MEMBERS - NONE

PROCEDURE PROCLIB (DCNLJET)

FUNCTIONS

1. LOAD SFIELD AND JWRCOAN DATA UNITS IF NOT ASSIGNED.
2. INITIALIZE DEFAULT VALUES OF USER PARAMETERS IF NOT DEFINED.
3. COMPUTE DIMENSIONLESS INPUTS TO MODULE CNLJET.
4. EXECUTE MODULE CNLJET.

MODULES CALLED - CNLJET

PROCEDURES CALLED - XCNLJET

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (DFNKAFM)

****PROCEDURE DFNKAFM

PURPOSE - TO ALLOW EXECUTION OF THE AIRFRAME NOISE MODULE WITH DIMENSIONAL INPUTS. USEFUL WHEN EXECUTION OF THE AIRFRAME NOISE PARAMETERS MODULE IS NOT DESIRED.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)
MMF(L03/00/09)

INPUT
USER PARAMETERS

		MODULES
AF	FLAP AREA (M**2),(FT**2) (RS) (DEFAULT=10. M**2)	FNKAFM
AH	HORIZONTAL TAIL AREA (M**2),(FT**2) (RS) (DEFAULT=20. M**2)	FNKAFM
AV	VERTICAL TAIL AREA (M**2),(FT**2) (RS) (DEFAULT=20. M**2)	FNKAFM
AW	WING AREA (M**2),(FT**2) (RS) (DEFAULT=100. M**2)	FNKAFM
BF	FLAP SPAN (M),(FT) (RS) (DEFAULT=5. M)	FNKAFM
BH	HORIZONTAL TAIL SPAN (M),(FT) (RS) (DEFAULT=10. M)	FNKAFM
BV	VERTICAL TAIL SPAN (M),(FT) (RS) (DEFAULT=10. M)	FNKAFM
BW	WING SPAN (M),(FT) (RS) (DEFAULT=20. M)	FNKAFM
DELTAF	FLAP SETTING (DEG) (RS) (DEFAULT=0.)	FNKAFM
DMG	TIRE DIAMETER OF MAIN LANDING GEAR (M), (FT) (RS) (DEFAULT=1. M)	FNKAFM
DNG	TIRE DIAMETER OF NOSE LANDING GEAR (M), (FT) (RS) (DEFAULT=1. M)	FNKAFM
LMG	MAIN LANDING GEAR STRUT LENGTH (M),(FT) (RS) (DEFAULT=3. M)	FNKAFM
LNG	NOSE LANDING GEAR STRUT LENGTH (M),(FT) (RS) (DEFAULT=3. M)	FNKAFM
CA	AMBIENT SPEED OF SOUND (M/S),(FT/S) (RS) (DEFAULT=340.294 M/S)	PARAM,FNKAFM IF,EVALUATE
RHOA	AMBIENT DENSITY (KG/M**3),(SLUG/FT**3) (RS) (DEFAULT=1.225 KG/M**3)	FNKAFM
MUA	AMBIENT DYNAMIC VISCOSITY (KG/(M S)), (SLUG/(FT S)) (RS) (DEFAULT=1.7894E-5 KG/(M S))	FNKAFM
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (DEFAULT=1. M)	IF,PARAM, FNKAFM
VO	AIRCRAFT VELOCITY (M/S),(FT/S) (RS) (DEFAULT=100. M/S)	PARAM,FNKAFM IF,EVALUATE
STIME	SOURCE TIME AT WHICH NOISE IS EVALUATED (S) (RS) (DEFAULT=0. S)	FNKAFM

PROCEDURE PROCLIB (DFNKAFM)

```
NWMG    NUMBER OF WHEELS PER MAIN LANDING GEAR      FNKAFM
        (I) (DEFAULT=4)
NWNG    NUMBER OF WHEELS PER NOSE LANDING GEAR      FNKAFM
        (I) (DEFAULT=2)
NMG     NUMBER OF MAIN LANDING GEAR (I)             FNKAFM
        (DEFAULT=2)
NNG     NUMBER OF NOSE LANDING GEAR (I)             FNKAFM
        (DEFAULT=1)
NS      NUMBER OF SLOTS FOR TRAILING EDGE FLAPS     FNKAFM
        (I) (DEFAULT=3)
IPRINT  OUTPUT PRINT CODE (I) (DEFAULT=3)           FNKAFM
        0 - NO PRINT
        1 - INPUT PRINT ONLY
        2 - OUTPUT PRINT ONLY
        3 - BOTH INPUT AND OUTPUT PRINT
IOUT    OUTPUT CODE FOR TABLE AND PRINTED OUTPUT    FNKAFM
        (I) (DEFAULT=3)
        0 - NO PRINT BUT GENERATE FNKAFM(XXXNNN).
       -1 - OUTPUT PRINT IN DB BUT NO TABLE
       -2 - OUTPUT PRINT IN DIMENSIONLESS FORM
            BUT NO TABLE.
       -3 - OUTPUT PRINT IN DIMENSIONLESS FORM
            AND DB BUT NO TABLE.
        1 - OUTPUT PRINT IN DB AND GENERATE
            FNKAFM(XXXNNN).
        2 - OUTPUT PRINT IN DIMENSIONLESS FORM
            AND GENERATE FNKAFM(XXXNNN).
        3 - OUTPUT PRINT IN DIMENSIONLESS FORM
            AND DB AND GENERATE FNKAFM(XXXNNN).
SCRNNN  VALUE USED TO FORM TABLE NAME               FNKAFM
        FNKAFM(XXXNNN) (I) (DEFAULT=1)
SCRXXX  THREE LETTER ID USED TO FORM TABLE NAME     FNKAFM
        FNKAFM(XXXNNN) (A) (DEFAULT=3HXXX)
DYNCLN  DESCRIPTION OF AIRCRAFT (L)                 FNKAFM
        (DEFAULT=.FALSE.)
        .TRUE.  - AERODYNAMICALLY CLEAN AIRCRAFT
        .FALSE. - NON-AERODYNAMICALLY CLEAN
                  AIRCRAFT
DELTAW  TYPE OF WING PLANFORM (L)                   FNKAFM
        (DEFAULT=.FALSE.)
        .TRUE.  - DELTA-WINGED AIRCRAFT
        .FALSE. - RECTANGULAR-WINGED AIRCRAFT
LANDG   LANDING GEAR POSITION (A) (DEFAULT=4HDOWN) FNKAFM
        4HDOWN - GEAR DOWN
        4HUP   - GEAR UP
IUNITS  UNIT OPTION CODE (A) (DEFAULT=2HSI)        FNKAFM
        2HSI      - SI UNITS
        7HENGLISH - ENGLISH UNITS
```

PROCEDURE PROCLIB (DFNKAFM)

```
(THE NEXT SEVEN PARAMETERS HAVE THE FOLLOWING VALUES)
(        .TRUE.  - INCLUDE IN TOTAL PREDICTION       )
(        .FALSE. - DO NOT INCLUDE                    )

TEWN     TRAILING EDGE WING NOISE (L)                 FNKAFM
         (DEFAULT=.TRUE.)
TEHTN    TRAILING EDGE HORIZONTAL TAIL NOISE (L)      FNKAFM
         (DEFAULT=.TRUE.)
TEVTN    TRAILING EDGE VERTICAL TAIL NOISE (L)        FNKAFM
         (DEFAULT=.TRUE.)
TEFN     TRAILING EDGE FLAP NOISE (L)                 FNKAFM
         (DEFAULT=.FALSE.)
LESN     LEADING EDGE SLAT NOISE (L)                  FNKAFM
         (DEFAULT=.FALSE.)
MLGN     MAIN LANDING GEAR NOISE (L)                  FNKAFM
         (DEFAULT=.TRUE.)
NLGN     NOSE LANDING GEAR NOISE (L)                  FNKAFM
         (DEFAULT=.TRUE.)

MEMBERS                                                MODULES

SFIELD(FREQ ) - ONE RECORD MEMBER OF THE VALUES        FNKAFM
                OF FREQUENCY IN HERTZ.
SFIELD(THETA) - ONE RECORD MEMBER OF THE VALUES        FNKAFM
                OF POLAR DIRECTIVITY ANGLE IN
                DEGREES.
SFIELD(PHI  ) - ONE RECORD MEMBER OF THE VALUES        FNKAFM
                OF AZIMUTHAL DIRECTIVITY ANGLE
                IN DEGREES.

OUTPUT
USER PARAMETERS                                        MODULES

RS       DISTANCE FROM SOURCE TO OBSERVER (M),(FT)     FNKAFM
         (RS)

MEMBERS                                                MODULES

FNKAFM(XXXNNN) - TYPE 1 DATA TABLE OF THE              FNKAFM
                 MEAN-SQUARE ACOUSTIC PRESSURE.

LOCAL
USER PARAMETERS                                        MODULES

MA       AIRCRAFT MACH NUMBER (RS)                     FNKAFM,
                                                      EVALUATE

MEMBERS - NONE
```

PROCEDURE PROCLIB (DFNKAFM)

FUNCTIONS

1. LOAD SFIELD DATA UNIT IF NOT ASSIGNED.
2. INITIALIZE DEFAULT VALUES OF USER PARAMETERS IF NOT DEFINED.
3. COMPUTE DIMENSIONLESS INPUT TO MODULE FNKAFM.
4. EXECUTE MODULE FNKAFM.

MODULES CALLED - FNKAFM

PROCEDURES CALLED - XFNKAFM

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (DGECOR)

****PROCEDURE DGECOR

PURPOSE - TO ALLOW EXECUTION OF THE COMBUSTION NOISE MODULE WITH DIMENSIONAL INPUTS. USEFUL WHEN EXECUTION OF THE CORE NOISE PARAMETERS MODULE IS NOT DESIRED.

AUTHOR - DSW(L03/00/00)
MMF(L03/00/09)

INPUT

USER PARAMETERS

	USER PARAMETERS	MODULES
NENG	NUMBER OF ENGINES (I) (DEFAULT=1)	GECOR
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS) (DEFAULT=1. M)	GECOR IF,PARAM
A	COMBUSTOR ENTRANCE AREA (M**2),(FT**2) (RS) (DEFAULT=PI/4. M**2)	IF,EVALUATE PARAM,GECOR
MDOT	COMBUSTOR ENTRANCE MASS FLOW RATE (KG/S),(SLUG/S) (RS) (DEFAULT=65 KG/S)	IF,EVALUATE PARAM,GECOR
VO	AIRCRAFT VELOCITY (M/S),(FT/S) (RS) (DEFAULT=0.)	IF,EVALUATE PARAM,GECOR
PI	COMBUSTOR ENTRANCE TOTAL PRESSURE (N/M**2),(LB/FT**2) (RS) (DEFAULT=101325. N/M**2)	IF,EVALUATE PARAM,GECOR
TI	COMBUSTOR ENTRANCE TOTAL TEMPERATURE (DEG K),(DEG R) (RS) (DEFAULT=288.15 DEG K)	IF,EVALUATE PARAM,GECOR
TJ	COMBUSTOR EXIT TOTAL TEMPERATURE (DEG K),(DEG R) (RS) (DEFAULT=576.30 DEG K)	IF,EVALUATE PARAM,GECOR
TDDELT	DESIGN TURBINE TEMPERATURE RISE (DEG K),(DEG R) (RS) (DEFAULT=144.07 DEG K)	IF,EVALUATE PARAM,GECOR
CA	AMBIENT SPEED OF SOUND (M/S),(FT/S) (RS) (DEFAULT=340.294 M/S)	IF,EVALUATE PARAM,GECOR
PA	AMBIENT PRESSURE (N/M**2),(LB/FT**2) (RS) (DEFAULT=101325. N/M**2)	IF,PARAM, EVALUATE
TA	AMBIENT TEMPERATURE (DEG K),(DEG R) (RS) (DEFAULT=288.15 DEG K)	IF,PARAM, EVALUATE
RHOA	AMBIENT DENSITY (KG/M**3),(SLUG/FT**3) (RS) (DEFAULT=1.225 KG/M**3)	IF,EVALUATE PARAM,GECOR
STIME	SOURCE TIME AT WHICH NOISE IS COMPUTED (S) (RS) (DEFAULT=0.)	GECOR
SCRXXX	THREE LETTER ID USED TO FORM TABLE NAME GECOR(XXXNNN). (A) (DEFAULT=3HXXX)	GECOR
SCRNNN	VALUE USED TO FORM TABLE NAME GECOR(XXXNNN) (I) (DEFAULT=1)	GECOR

```
    IOUT     OUTPUT CODE FOR TABLE AND PRINTED OUTPUT
             (I) (DEFAULT=3)
             0 - NO PRINT BUT GENERATE GECOR(XXXNNN).
            -1 - OUTPUT PRINT IN DB BUT NO TABLE.
            -2 - OUTPUT PRINT IN DIMENSIONLESS FORM
                 BUT NO TABLE.
            -3 - OUTPUT PRINT IN DIMENSIONLESS FORM
                 AND DB BUT NO TABLE.
             1 - OUTPUT PRINT IN DB AND GENERATE
                 GECOR(XXXNNN).
             2 - OUTPUT PRINT IN DIMENSIONLESS FORM
                 AND GENERATE GECOR(XXXNNN).
             3 - OUTPUT PRINT IN DIMENSIONLESS FORM
                 AND DB AND GENERATE GECOR(XXXNNN).
    IPRINT   OUTPUT PRINT CODE (I) (DEFAULT=3)          GECOR
             0 - NO PRINT
             1 - INPUT PRINT ONLY
             2 - OUTPUT PRINT ONLY
             3 - BOTH INPUT AND OUTPUT PRINT
    IUNITS   UNIT OPTION CODE (A) (DEFAULT=2HSI)        GECOR
             2HSI     - SI UNITS
             7HENGLISH - ENGLISH UNITS

  MEMBERS                                              MODULES

    SFIELD(FREQ  ) - ONE RECORD MEMBER OF THE VALUES    GECOR
                     OF FREQUENCY IN HERTZ.
    SFIELD(THETA ) - ONE RECORD MEMBER OF THE VALUES    GECOR
                     OF POLAR DIRECTIVITY ANGLE IN
                     DEGREES.
    SFIELD(PHI   ) - ONE RECORD MEMBER OF THE VALUES    GECOR
                     OF AZIMUTHAL DIRECTIVITY ANGLE
                     IN DEGREES.

OUTPUT
  USER PARAMETERS                                      MODULES

    RS       RADIAL DISTANCE FROM SOURCE TO OBSERVER    GECOR
             (M),(FT) (RS)
    GECERR   ERROR PARAMETER (I)                        GECOR
             0 - NO ERRORS ENCOUNTERED.
             1 - MEMBER MANAGER ERROR OCCURRED ON OPEN.
             2 - INSUFFICIENT LDS AVAILABLE.
             3 - MEMBER MANAGER ERROR OCCURRED ON READ.
             4 - AT LEAST ONE USER PARAMETER HAS AN
                 INVALID VALUE.
             5 - UNABLE TO BUILD TABLE GECOR(XXXNNN).
             6 - AN INVALID VALUE WAS CALCULATED
                 DURING EXECUTION.
```

PROCEDURE PROCLIB (DGECOR)

MEMBERS		MODULES
GECOR (XXXNNN)	- TYPE 1 DATA TABLE OF THE MEAN-SQUARE ACOUSTIC PRESSURE.	GECOR

LOCAL

USER PARAMETERS		MODULES
MA	AIRCRAFT MACH NUMBER (RS)	EVALUATE, GECOR
TEMP	6 ELEMENT PARAMETER FOR SAVING INPUT VALUES (RS)	PARAM, EVALUATE
AE	ENGINE REFERENCE AREA (M**2),(FT**2) (RS)	PARAM,GECOR, EVALUATE

MEMBERS - NONE

FUNCTIONS

1. LOAD SFIELD DATA UNIT IF NOT ASSIGNED.
2. INITIALIZE DEFAULT VALUES OF USER PARAMETERS IF NOT DEFINED.
3. COMPUTE DIMENSIONLESS INPUTS TO MODULE GECOR.
4. EXECUTE MODULE GECOR.

MODULES CALLED - GECOR

PROCEDURES CALLED - XGECOR

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (DGETUR)

****PROCEDURE DGETUR

PURPOSE - TO ALLOW EXECUTION OF THE TURBINE NOISE MODULE WITH DIMENSIONAL INPUTS. USEFUL WHEN EXECUTION OF THE JET NOISE PARAMETERS MODULE IS NOT DESIRED.

AUTHOR - DSW(L03/00/00)
WKB(L03/00/01)
MMF(L03/00/09)

INPUT

USER PARAMETERS

		MODULES
NENG	NUMBER OF ENGINES (I) (DEFAULT=1)	GETUR
RS	DISTANCE FROM SOURCE TO OBSERVER, (M),(FT) (RS) (DEFAULT=1. M)	GETUR IF,PARAM
AREA	TURBINE INLET CROSS-SECTIONAL AREA (M**2),(FT**2) (RS) (DEFAULT=PI/4. M**2)	GETUR,PARAM IF,EVALUATE
F	FUEL-TO-AIR RATIO (RS) (DEFAULT=0.)	GETUR
HA	ABSOLUTE HUMIDITY, PERCENT MOLE FRACTION (RS) (DEFAULT=0.)	GETUR
NBLADE	NUMBER OF ROTOR BLADES (I) (DEFAULT=20)	GETUR
D	TURBINE ROTOR DIAMETER (M),(FT) (RS) (DEFAULT=SQRT(PI/4.) M)	GETUR,PARAM IF,EVALUATE
VO	AIRCRAFT VELOCITY (M/S),(FT/S) (RS) (DEFAULT=0.)	IF,PARAM EVALUATE
ROTSPD	ROTATIONAL SPEED (RAD/S) (RS) (DEFAULT=115.2 RAD/S)	GETUR,PARAM IF,EVALUATE
TTI	ENTRANCE TOTAL TEMPERATURE (DEG K), (DEG R) (RS) (DEFAULT=1152.6 DEG K)	GETUR,PARAM IF,EVALUATE
TSJ	EXIT STATIC TEMPERATURE (DEG K), (DEG R) (RS) (DEFAULT=288.15 DEG K)	GETUR,PARAM IF,EVALUATE
CA	AMBIENT SPEED OF SOUND (M/S),(FT/S) (RS) (DEFAULT=340.294 M/S)	GETUR,PARAM IF,EVALUATE
RHOA	AMBIENT DENSITY (KG/M**3),(SLUG/FT**3) (RS) (DEFAULT=1.225 KG/M**3)	GETUR,PARAM IF,EVALUATE
TA	AMBIENT TEMPERATURE (DEG K),(DEG R) (RS) (DEFAULT=288.15)	IF,PARAM EVALUATE
SCRXXX	THREE LETTER ID USED TO FORM TABLE NAME GETUR(XXXNNN) (A) (DEFAULT=3HXXX)	GETUR
SCRNNN	VALUE USED TO FORM TABLE NAME GETUR(XXXNNN) (I) (DEFAULT=1)	GETUR
BROAD	BROADBAND NOISE OPTION (L) .TRUE. COMPUTE BROADBAND NOISE (DEFAULT) .FALSE. DO NOT COMPUTE	GETUR
PURE	PURE TONE OPTION (L) .TRUE. COMPUTE PURE TONES (DEFAULT) .FALSE. DO NOT COMPUTE	GETUR

PROCEDURE PROCLIB (DGETUR)

IOUT	OUTPUT CODE FOR TABLE AND PRINTED OUTPUT (I) (DEFAULT=3) 0 - NO PRINT BUT GENERATE GETUR(XXXNNN). -1 - OUTPUT PRINT IN DB BUT NO TABLE. -2 - OUTPUT PRINT IN DIMENSIONLESS FORM BUT NO TABLE. -3 - OUTPUT PRINT IN DIMENSIONLESS FORM AND DB BUT NO TABLE. 1 - OUTPUT PRINT IN DB AND GENERATE GETUR(XXXNNN). 2 - OUTPUT PRINT IN DIMENSIONLESS FORM AND GENERATE GETUR(XXXNNN). 3 - OUTPUT PRINT IN DIMENSIONLESS FORM AND DB AND GENERATE GETUR(XXXNNN).	GETUR
IPRINT	OUTPUT PRINT CODE (I) (DEFAULT=3) 0 - NO PRINT 1 - INPUT PRINT ONLY 2 - OUTPUT PRINT ONLY 3 - BOTH INPUT AND OUTPUT PRINT	GETUR
IUNITS	UNIT OPTION CODE (A) (DEFAULT=2HSI) 2HSI - SI UNITS 7HENGLISH - ENGLISH UNITS	IF,GETUR, PARAM

MEMBERS

		MODULES
SFIELD(FREQ)	- ONE RECORD MEMBER OF THE VALUES OF FREQUENCY IN HERTZ.	GETUR
SFIELD(THETA)	- ONE RECORD MEMBER OF THE VALUES OF POLAR DIRECTIVITY ANGLE IN DEGREES.	GETUR
SFIELD(PHI)	- ONE RECORD MEMBER OF THE VALUES OF AZIMUTHAL DIRECTIVITY ANGLE IN DEGREES.	GETUR

OUTPUT

USER PARAMETERS

		MODULES
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS)	GETUR
GETERR	ERROR PARAMETER (I) 0 - NO ERRORS ENCOUNTERED. 1 - MEMBER MANAGER ERROR OCCURRED ON OPEN. 2 - INSUFFICIENT LDS AVAILABLE. 3 - MEMBER MANAGER ERROR OCCURRED ON READ. 4 - AT LEAST ONE USER PARAMETER HAS AN INVALID VALUE. 5 - UNABLE TO BUILD TABLE GETUR(XXXNNN). 6 - AN INVALID VALUE WAS CALCULATED DURING EXECUTION.	GETUR

PROCEDURE PROCLIB (DGETUR)

MEMBERS		MODULES
GETUR(XXXNNN) - TYPE 1 DATA TABLE OF THE MEAN-SQUARE ACOUSTIC PRESSURE.		GETUR

LOCAL

USER PARAMETERS		MODULES
MA	AIRCRAFT MACH NUMBER (RS)	GETUR, EVALUATE
AE	ENGINE REFERENCE AREA (M**2),(FT**2) (RS)	PARAM,GETUR, EVALUATE
TEMP	5 ELEMENT PARAMETER FOR SAVING INPUT VALUES (RS)	PARAM, EVALUATE

MEMBERS - NONE

FUNCTIONS

1. LOAD SFIELD DATA UNIT IF NOT ASSIGNED.
2. INITIALIZE DEFAULT VALUES OF USER PARAMETERS IF NOT DEFINED.
3. COMPUTE DIMENSIONLESS INPUTS TO MODULE GETUR.
4. EXECUTE MODULE GETUR.

MODULES CALLED - GETUR

PROCEDURES CALLED - XGETUR

CALLING PROCEDURES - NONE

****PROCEDURE DHDNFAN

PURPOSE - TO ALLOW EXECUTION OF THE FAN NOISE MODULE WITH DIMENSIONAL INPUTS. USEFUL WHEN EXECUTION OF THE FAN NOISE PARAMETERS MODULE IS NOT DESIRED.

AUTHOR - DSW(L03/00/00)
WKB(L03/00/01)
MMF(L03/00/09)

INPUT
USER PARAMETERS

		MODULES
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS) (DEFAULT=1. M)	HDNFAN, IF,PARAM
AFAN	FAN INLET CROSS-SECTIONAL AREA (M),(FT) (RS) (DEFAULT=1. M**2)	HDNFAN,PARAM, IF,EVALUATE
DIAM	FAN ROTOR DIAMETER (M),(FT) (RS) (DEFAULT=1. M)	HDNFAN,PARAM, IF,EVALUATE
MD	FAN ROTOR RELATIVE TIP MACH NUMBER AT DESIGN POINT (RS) (DEFAULT=1.)	HDNFAN
RSS	ROTOR-STATOR SPACING (M),(FT) (RS) (DEFAULT=0.01 M)	HDNFAN,PARAM, IF,EVALUATE
CH	MEAN ROTOR BLADE CHORD (M),(FT) (RS) (DEFAULT=0.01 M)	EVALUATE, IF,PARAM
MDOT	MASS FLOW RATE (KG/S),(SLUG/S) (RS) (DEFAULT=80 KG/S)	HDNFAN,PARAM, IF,EVALUATE
VO	AIRCRAFT VELOCITY (M/S),(FT/S) (RS) (DEFAULT=0.)	EVALUATE, IF,PARAM
N	ROTATIONAL SPEED (HZ) (RS) (DEFAULT=100.)	HDNFAN,PARAM, IF,EVALUATE
DELTAT	TOTAL TEMPERATURE RISE ACROSS FAN (DEG K),(DEG R) (RS) (DEFAULT=60 DEG K)	HDNFAN,PARAM, IF,EVALUATE
CA	AMBIENT SPEED OF SOUND (M/S),(FT/S) (RS) (DEFAULT=340.294 M/S)	EVALUATE, IF,PARAM
RHOA	AMBIENT DENSITY (KG/M**3),(SLUG/FT**3) (DEFAULT=1.225 KG/M**3)	EVALUATE, IF,PARAM
TA	AMBIENT TEMPERATURE (DEG K),(DEG R) (RS) (DEFAULT=288.15 DEG K)	EVALUATE, IF,PARAM
NBANDS	NUMBER OF 1/3 OCTAVE BANDS FOR TONE FREQUENCY SHIFT (I) (DEFAULT=0)	HDNFAN
NENG	NUMBER OF ENGINES (I) (DEFAULT=1)	HDNFAN
NB	NUMBER OF ROTOR BLADES (I) (DEFAULT=20)	HDNFAN
NV	NUMBER OF STATOR VANES (I) (DEFAULT=50)	HDNFAN
IGV	INLET GUIDE VANE INDEX (I) (DEFAULT=1) 1 - FOR A FAN WITHOUT IGV'S 2 - FOR A FAN WITH IGV'S	HDNFAN

PROCEDURE PROCLIB (DHDNFAN)

DIS	INLET FLOW DISTORTION INDEX (I) 1 - NO INLET FLOW DISTORTION (DEFAULT) 2 - INLET FLOW DISTORTED	HDNFAN
IOUT	OUTPUT CODE FOR TABLE AND PRINTED OUTPUT (I) (DEFAULT=3) 0 - NO PRINT BUT GENERATE HDNFAN(XXXNNN). -1 - OUTPUT PRINT IN DB BUT NO TABLE. -2 - OUTPUT PRINT IN DIMENSIONLESS FORM BUT NO TABLE. -3 - OUTPUT PRINT IN DIMENSIONLESS FORM AND DB BUT NO TABLE. 1 - OUTPUT PRINT IN DB AND GENERATE HDNFAN(XXXNNN). 2 - OUTPUT PRINT IN DIMENSIONLESS FORM AND GENERATE HDNFAN(XXXNNN). 3 - OUTPUT PRINT IN DIMENSIONLESS FORM AND DB AND GENERATE HDNFAN(XXXNNN).	HDNFAN
IPRINT	OUTPUT PRINT CODE (I) (DEFAULT=3) 0 - NO PRINT 1 - INPUT PRINT ONLY 2 - OUTPUT PRINT ONLY 3 - BOTH INPUT AND OUTPUT PRINT	HDNFAN
STIME	SOURCE TIME AT WHICH NOISE IS CALCULATED (S) (RS) (DEFAULT=0.)	HDNFAN
SCRXXX	THREE LETTER ID USED TO FORM TABLE NAME HDNFAN(XXXNNN) (A) (DEFAULT=3HXXX)	HDNFAN
SCRNNN	VALUE USED TO FORM TABLE NAME HDNFAN(XXXNNN) (I) (DEFAULT=1)	HDNFAN
IUNITS	UNIT OPTION CODE (A) (DEFAULT=2HSI) 2HSI - SI UNITS 7HENGLISH - ENGLISH UNITS	HDNFAN

(THE NEXT SIX PARAMETERS HAVE THE FOLLOWING VALUES (L))
(.TRUE. - INCLUDE IN TOTAL PREDICTION (DEFAULT))
(.FALSE. - DO NOT INCLUDE)

INRS	INLET ROTOR STATOR INTERACTION TONES	HDNFAN
INCT	COMBINATION TONE NOISE	HDNFAN
INDIS	INLET FLOW DISTORTION TONES	HDNFAN
INBB	INLET BROADBAND NOISE	HDNFAN
IDRS	DISCHARGE ROTOR-STATOR INTERACTION TONES	HDNFAN
IDBB	DISCHARGE BROADBAND NOISE	HDNFAN

MEMBERS	MODULES
SFIELD(FREQ) - ONE RECORD MEMBER OF THE VALUES OF FREQUENCY IN HERTZ.	HDNFAN
SFIELD(THETA) - ONE RECORD MEMBER OF THE VALUES OF POLAR DIRECTIVITY ANGLE IN DEGREES.	HDNFAN

PROCEDURE PROCLIB (DHDNFAN)

SFIELD(PHI)	- ONE RECORD MEMBER OF THE VALUES OF AZIMUTHAL DIRECTIVITY ANGLE IN DEGREES.	HDNFAN

OUTPUT

USER PARAMETERS

		MODULES
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS)	HDNFAN

MEMBERS

		MODULES
HDNFAN(XXXNNN)	- TYPE ONE DATA TABLE OF THE MEAN-SQUARE ACOUSTIC PRESSURE	HDNFAN

LOCAL

USER PARAMETERS

		MODULES
MA	AIRCRAFT MACH NUMBER (RS)	EVALUATE, HDNFAN
AE	ENGINE REFERENCE AREA (M**2),(FT**2) (RS)	PARAM,HDNFAN, EVALUATE
TEMP	6 ELEMENT PARAMETER FOR SAVING INPUT VALUES (RS)	PARAM, EVALUATE

MEMBERS - NONE

FUNCTIONS

1. LOAD SFIELD UNIT IF NOT ASSIGNED.
2. INITIALIZE DEFAULT VALUES OF USER PARAMETERS IF NOT DEFINED.
3. COMPUTE DIMENSIONLESS INPUTS TO MODULE HDNFAN.
4. EXECUTE MODULE HDNFAN.

MODULES CALLED - HDNFAN

PROCEDURES CALLED - XHDNFAN

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (DSAESHK)

****PROCEDURE DSAESHK

PURPOSE - TO ALLOW EXECUTION OF THE CIRCULAR JET SHOCK CELL NOISE MODULE WITH DIMENSIONAL INPUTS. USEFUL WHEN EXECUTION OF THE JET NOISE PARAMETERS MODULE IS NOT DESIRED.

AUTHOR - DSW(L03/00/00)
MMF(L03/00/09)

INPUT
USER PARAMETERS

		MODULES
AJ	AREA OF THE JET (M**2),(FT**2) (RS) (DEFAULT=PI/4. M**2)	SAESHK,PARAM, IF,EVALUATE
VJ	FULLY EXPANDED JET VELOCITY (M/S), (FT/S) (RS) (DEFAULT=340.294 M/S)	SAESHK,PARAM, IF,EVALUATE
RHOJ	FULLY EXPANDED JET DENSITY (KG/M**3), (SLUG/FT**3) (RS) (DEFAULT=0.6125 KG/M**3)	SAESHK,PARAM, IF,EVALUATE
TJ	JET TOTAL TEMPERATURE (DEG K),(DEG R) (RS) (DEFAULT=288.15 DEG K)	SAESHK,PARAM, IF,EVALUATE
CA	AMBIENT SPEED OF SOUND (M/S),(FT/S) (RS) (DEFAULT=340.294 M/S)	SAESHK,PARAM, IF,EVALUATE
RHOA	AMBIENT DENSITY (KG/M**3),(SLUG/FT**3) (RS) (DEFAULT=1.225 KG/M**3)	SAESHK,PARAM, IF,EVALUATE
TA	AMBIENT TEMPERATURE (DEG K),(DEG R) (RS) (DEFAULT=288.15 DEG K)	EVALUATE, IF,PARAM
RS	RADIAL DISTANCE FROM NOZZLE EXIT TO OBSERVER (M),(FT) (RS) (DEFAULT=1. M)	SAESHK, IF,PARAM
VO	AIRCRAFT VELOCITY (M/S),(FT/S) (RS) (DEFAULT=0.)	EVALUATE, IF,PARAM
NSHK	NUMBER OF SHOCKS (I) (DEFAULT=8)	SAESHK
DELTA	ANGLE BETWEEN FLIGHT VECTOR AND ENGINE INLET AXIS (DEG) (RS) (DEFAULT=0.)	SAESHK
NENG	NUMBER OF ENGINES (I) (DEFAULT=1)	SAESHK
SCRXXX	THREE LETTER ID USED TO FORM TABLE NAME SAESHK(XXXNNN) (A) (DEFAULT=3HXXX)	SAESHK
SCRNNN	VALUE USED TO FORM TABLE NAME SAESHK(XXXNNN) (I) (DEFAULT=1)	SAESHK
STIME	SOURCE TIME AT WHICH NOISE IS CALCULATED (S) (RS) (DEFAULT=0. S)	SAESHK
IOUT	OUTPUT CODE FOR TABLE AND PRINTED OUTPUT (I) (DEFAULT=3) 0 - NO PRINT BUT GENERATE SAESHK(XXXNNN). -1 - OUTPUT PRINT IN DB BUT NO TABLE. -2 - OUTPUT PRINT IN DIMENSIONLESS FORM BUT NO TABLE.	SAESHK

```
          -3 - OUTPUT PRINT IN DIMENSIONLESS FORM
               AND DB BUT NO TABLE.
           1 - OUTPUT PRINT IN DB AND GENERATE
               SAESHK(XXXNNN).
           2 - OUTPUT PRINT IN DIMENSIONLESS FORM
               AND GENERATE SAESHK(XXXNNN).
           3 - OUTPUT PRINT IN DIMENSIONLESS FORM
               AND DB AND GENERATE SAESHK(XXXNNN).
   IPRINT  OUTPUT PRINT CODE (I) (DEFAULT=3)          SAESHK
           0 - NO PRINT
           1 - INPUT PRINT ONLY
           2 - OUTPUT PRINT ONLY
           3 - BOTH INPUT AND OUTPUT PRINT
   IUNITS  UNIT OPTION CODE (A) (DEFAULT=2HSI)        SAESHK
           2HSI     - SI UNITS
           7HENGLISH - ENGLISH UNITS

  MEMBERS                                             MODULES

   SFIELD(FREQ  ) - ONE RECORD MEMBER OF THE VALUES   SAESHK
                    OF FREQUENCY IN HERTZ.
   SFIELD(THETA ) - ONE RECORD MEMBER OF THE VALUES   SAESHK
                    OF POLAR DIRECTIVITY ANGLE IN
                    DEGREES.
   SFIELD(PHI   ) - ONE RECORD MEMBER OF THE VALUES   SAESHK
                    OF AZIMUTHAL DIRECTIVITY ANGLE
                    IN DEGREES.

OUTPUT
 USER PARAMETERS                                      MODULES

   RS      RADIAL DISTANCE FROM NOZZLE EXIT TO        SAESHK
           OBSERVER (M),(FT) (RS)
   SAEERR  ERROR PARAMETER (I)                        SAESHK
           0 - NO ERRORS ENCOUNTERED.
           1 - MEMBER MANAGER ERROR OCCURRED ON OPEN.
           2 - INSUFFICIENT LDS AVAILABLE.
           3 - MEMBER MANAGER ERROR OCCURRED ON READ.
           4 - AT LEAST ONE USER PARAMETER HAS AN
               INVALID VALUE.
           5 - UNABLE TO BUILD TABLE SAESHK(XXXNNN).
           6 - AN INVALID VALUE WAS CALCULATED
               DURING EXECUTION.

  MEMBERS                                             MODULES

   SAESHK(XXXNNN) - TYPE 1 DATA TABLE OF THE          SAESHK
                    MEAN-SQUARE ACOUSTIC PRESSURE.
```

PROCEDURE PROCLIB (DSAESHK)

LOCAL

USER PARAMETERS

		MODULES
MA	AIRCRAFT MACH NUMBER (RS)	EVALUATE, SAESHK
AE	ENGINE REFERENCE AREA (M**2),(FT**2) (RS)	EVALUATE, PARAM,SAESHK
TEMP	4 ELEMENT PARAMETER FOR SAVING INPUT VALUES	PARAM, EVALUATE

MEMBERS - NONE

FUNCTIONS

1. LOAD SFIELD DATA UNIT IF NOT ASSIGNED.
2. INITIALIZE DEFAULT VALUES OF USER PARAMETERS IF NOT DEFINED.
3. COMPUTE DIMENSIONLESS INPUTS TO MODULE SAESHK.
4. EXECUTE MODULE SAESHK.

MODULES CALLED - SAESHK

PROCEDURES CALLED - XSAESHK

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (DSGLJET)

****PROCEDURE DSGLJET

PURPOSE - TO ALLOW EXECUTION OF THE SINGLE STREAM CIRCULAR JET NOISE MODULE WITH DIMENSIONAL INPUTS. USEFUL WHEN EXECUTION OF THE JET NOISE PARAMETERS MODULE IS NOT DESIRED.

AUTHOR - DSW(L03/00/00)
MMF(L03/00/09)

INPUT

USER PARAMETERS

		MODULES
AJ	AREA OF FULLY EXPANDED JET (M**2), (FT**2) (RS) (DEFAULT=PI/4. M**2)	SGLJET,PARAM, IF,EVALUATE
RHOJ	DENSITY OF JET (KG/M**3),(SLUG/FT**3) (RS) (DEFAULT=1.225 KG/M**3)	SGLJET,PARAM, IF,EVALUATE
TJ	JET TOTAL TEMPERATURE (DEG K),(DEG R) (RS) (DEFAULT=288.15 DEG K)	SGLJET,PARAM, IF,EVALUATE
VJ	JET VELOCITY (M/S),(FT/S) (RS) (DEFAULT=340.294 M/S)	SGLJET,PARAM, IF,EVALUATE
RS	RADIAL DISTANCE FROM NOZZLE EXIT TO OBSERVER (M),(FT) (RS) (DEFAULT=1. M)	SGLJET, IF,PARAM
RHOA	AMBIENT DENSITY (KG/M**3),(SLUG/FT**3) (RS) (DEFAULT=1.225 KG/M**3)	SGLJET,PARAM, IF,EVALUATE
TA	AMBIENT TEMPERATURE (DEG K),(DEG R) (RS) (DEFAULT=288.15 DEG K)	IF,PARAM, EVALUATE
CA	AMBIENT SPEED OF SOUND (M/S),(FT/S) (RS) (DEFAULT=340.294 M/S)	SGLJET,PARAM, IF,EVALUATE
DELTA	ANGLE BETWEEN FLIGHT DIRECTION AND ENGINE INLET AXIS (DEGREES) (RS) (DEFAULT=0.)	SGLJET
VO	AIRCRAFT VELOCITY (M/S),(FT/S) (RS) (DEFAULT=0.)	IF,PARAM, EVALUATE
STIME	SOURCE TIME AT WHICH NOISE IS CALCULATED (S) (RS) (DEFAULT=0. S)	SGLJET
SCRXXX	THREE LETTER ID USED TO FORM TABLE NAME SGLJET(XXXNNN). (A) (DEFAULT=3HXXX)	SGLJET
SCRNNN	VALUE USED TO FORM TABLE NAME SGLJET(XXXNNN). (I) (DEFAULT=1)	SGLJET
IOUT	OUTPUT CODE FOR TABLE AND PRINTED OUTPUT (I) (DEFAULT=3) 0 - NO PRINT BUT GENERATE SGLJET(XXXNNN). -1 - OUTPUT PRINT IN DB BUT NO TABLE. -2 - OUTPUT PRINT IN DIMENSIONLESS FORM BUT NO TABLE. -3 - OUTPUT PRINT IN DIMENSIONLESS FORM AND DB BUT NO TABLE. 1 - OUTPUT PRINT IN DB AND GENERATE SGLJET(XXXNNN).	SGLJET

```
             2 - OUTPUT PRINT IN DIMENSIONLESS FORM
                 AND GENERATE SGLJET(XXXNNN).
             3 - OUTPUT PRINT IN DIMENSIONLESS FORM
                 AND DB AND GENERATE SGLJET(XXXNNN).
    IPRINT   OUTPUT PRINT CODE (I) (DEFAULT=3)               SGLJET
             0 - NO PRINT
             1 - INPUT PRINT ONLY
             2 - OUTPUT PRINT ONLY
             3 - BOTH INPUT AND OUTPUT PRINT
    IUNITS   UNIT OPTION CODE (A) (DEFAULT=2HSI)             SGLJET,
             2HSI      - SI UNITS                           IF,PARAM
             7HENGLISH - ENGLISH UNITS
    NENG     NUMBER OF ENGINES (I) (DEFAULT=1)               SGLJET

  MEMBERS                                                    MODULES

    SFIELD(FREQ  ) - ONE RECORD MEMBER OF THE VALUES        SGLJET
                     OF FREQUENCY IN HERTZ.
    SFIELD(THETA ) - ONE RECORD MEMBER OF THE VALUES        SGLJET
                     OF POLAR DIRECTIVITY ANGLE IN
                     DEGREES.
    SFIELD(PHI   ) - ONE RECORD MEMBER OF THE VALUES        SGLJET
                     OF AZIMUTHAL DIRECTIVITY ANGLE
                     IN DEGREES.
    SAE    (OM   ) - TYPE 1 DATA TABLE OF THE JET           SGLJET
                     VARIABLE DENSITY EXPONENT.
    SAE    (PDF  ) - TYPE 1 DATA TABLE OF THE POWER         SGLJET
                     DEVIATION FACTOR.
    SAE    (NDF  ) - TYPE 1 DATA TABLE OF THE               SGLJET
                     NORMALIZED DIRECTIVITY FUNCTION.
    SAE    (MTH  ) - TYPE 1 DATA TABLE OF THE FORWARD       SGLJET
                     VELOCITY INDEX.
    SAE    (SCJ  ) - TYPE 1 DATA TABLE OF THE               SGLJET
                     SPECTRAL DISTRIBUTION FACTOR.
    SAE    (SCF  ) - TYPE 1 DATA TABLE OF THE               SGLJET
                     STROUHAL NUMBER CORRECTION
                     FACTOR.
OUTPUT
  USER PARAMETERS                                            MODULES

    RS       RADIAL DISTANCE FROM NOZZLE EXIT TO             SGLJET
             OBSERVER (M),(FT) (RS)

  MEMBERS                                                    MODULES

    SGLJET(XXXNNN) - TYPE 1 DATA TABLE OF THE               SGLJET
                     MEAN-SQUARE ACOUSTIC PRESSURE.
```

PROCEDURE PROCLIB (DSGLJET)

LOCAL

USER PARAMETERS

		MODULES
MA	AIRCRAFT MACH NUMBER (RS)	SGLJET, EVALUATE
AE	ENGINE REFERENCE AREA (M**2),(FT**2) (RS)	EVALUATE, PARAM,SGLJET
TEMP	4 ELEMENT PARAMETER FOR SAVING INPUT VALUES (RS)	PARAM, EVALUATE

MEMBERS - NONE

FUNCTIONS

1. LOAD SAE AND SFIELD DATA UNITS IF NOT ASSIGNED.
2. INITIALIZE DEFAULT VALUES OF USER PARAMETERS IF NOT DEFINED.
3. COMPUTE DIMENSIONLESS INPUTS TO MODULE SGLJET.
4. EXECUTE MODULE SGLJET.

MODULES CALLED - SGLJET

PROCEDURES CALLED - XSGLJET

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (DSMBTUR)

****PROCEDURE DSMBTUR

PURPOSE - TO ALLOW EXECUTION OF THE SMITH AND BUSHELL TURBINE NOISE MODULE WITH DIMENSIONAL INPUTS. USEFUL WHEN EXECUTION OF THE JET NOISE PARAMETERS MODULE IS NOT DESIRED.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)
MMF(L03/00/09)

INPUT

USER PARAMETERS

		MODULES
NENG	NUMBER OF ENGINES (I) (DEFAULT=1)	SMBTUR
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS) (DEFAULT=1. M)	SMBTUR, IF,PARAM
AREA	TURBINE INLET CROSS-SECTIONAL AREA (M**2),(FT**2) (RS) (DEFAULT=PI/4. M**2)	SMBTUR,PARAM, IF,EVALUATE
FA	FUEL-TO-AIR RATIO (RS) (DEFAULT=0.)	SMBTUR
HA	ABSOLUTE HUMIDITY, PERCENT MOLE FRACTION (RS) (DEFAULT=0.)	SMBTUR
C	ROTOR BLADE MEAN AXIAL CHORD OF THE LAST STAGE (M),(FT) (RS) (DEFAULT=0.01 M)	SMBTUR,PARAM, IF,EVALUATE
D	TURBINE ROTOR DIAMETER (M),(FT) (RS) (DEFAULT=1. M)	IF,PARAM, EVALUATE
MDOT	CORE MASS FLOW RATE (KG/S),(SLUG/S) (RS) (DEFAULT=65. KG/S)	SMBTUR,PARAM, IF,EVALUATE
VO	AIRCRAFT VELOCITY (M/S),(FT/S) (RS) (DEFAULT=0.)	IF,PARAM, EVALUATE
N	ROTATIONAL SPEED (RAD/S) (RS) (DEFAULT=115.2 RAD/S)	SMBTUR,PARAM, IF,EVALUATE
TSJ	EXIT STATIC TEMPERATURE (DEG K), (DEG R) (RS) (DEFAULT=288.15 DEG K)	SMBTUR,PARAM, IF,EVALUATE
CA	AMBIENT SPEED OF SOUND (M/S),(FT/S) (RS) (DEFAULT=340.294 M/S)	SMBTUR,PARAM, IF,EVALUATE
RHOA	AMBIENT DENSITY (KG/M**3),(SLUG/FT**3) (RS) (DEFAULT=1.225 KG/M**3)	SMBTUR,PARAM, IF,EVALUATE
TA	AMBIENT TEMPERATURE (DEG K),(DEG R) (RS) (DEFAULT=288.15)	IF,PARAM, EVALUATE
NS	NUMBER OF STAGES (I) (DEFAULT=1)	SMBTUR
SCRXXX	THREE LETTER ID USED TO FORM TABLE NAME SMBTUR(XXXNNN) (A) (DEFAULT=3HXXX)	SMBTUR
SCRNNN	VALUE USED TO FORM TABLE NAME SMBTUR(XXXNNN) (I) (DEFAULT=1)	SMBTUR

PROCEDURE PROCLIB (DSMBTUR)

```
IOUT     OUTPUT CODE FOR TABLE AND PRINTED OUTPUT    SMBTUR
         (I) (DEFAULT=3)
          0 - NO PRINT BUT GENERATE SMBTUR(XXXNNN).
         -1 - OUTPUT PRINT IN DB BUT NO TABLE.
         -2 - OUTPUT PRINT IN DIMENSIONLESS FORM
              BUT NO TABLE.
         -3 - OUTPUT PRINT IN DIMENSIONLESS FORM
              AND DB BUT NO TABLE.
          1 - OUTPUT PRINT IN DB AND GENERATE
              SMBTUR(XXXNNN).
          2 - OUTPUT PRINT IN DIMENSIONLESS FORM
              AND GENERATE SMBTUR(XXXNNN).
          3 - OUTPUT PRINT IN DIMENSIONLESS FORM
              AND DB AND GENERATE SMBTUR(XXXNNN).
IPRINT   OUTPUT PRINT CODE (I) (DEFAULT=3)           SMBTUR
         0 - NO PRINT
         1 - INPUT PRINT ONLY
         2 - OUTPUT PRINT ONLY
         3 - BOTH INPUT AND OUTPUT PRINT
IUNITS   UNIT OPTION CODE (A) (DEFAULT=2HSI)         IF,SMBTUR,
         2HSI      - SI UNITS                         PARAM
         7HENGLISH - ENGLISH UNITS

MEMBERS                                              MODULES

  SFIELD(FREQ ) - ONE RECORD MEMBER OF THE VALUES    SMBTUR
                  OF FREQUENCY IN HERTZ.
  SFIELD(THETA) - ONE RECORD MEMBER OF THE VALUES    SMBTUR
                  OF POLAR DIRECTIVITY ANGLE IN
                  DEGREES.
  SFIELD(PHI  ) - ONE RECORD MEMBER OF THE VALUES    SMBTUR
                  OF AZIMUTHAL DIRECTIVITY ANGLE
                  IN DEGREES.

OUTPUT
 USER PARAMETERS                                     MODULES

  RS       DISTANCE FROM SOURCE TO OBSERVER (M),(FT) SMBTUR
           (RS)

 MEMBERS                                             MODULES

  SMBTUR(XXXNNN) - TYPE 1 DATA TABLE OF THE          SMBTUR
                   MEAN-SQUARE ACOUSTIC PRESSURE.
```

PROCEDURE PROCLIB (DSMBTUR)

LOCAL

USER PARAMETERS

		MODULES
MA	AIRCRAFT MACH NUMBER (RS)	SMBTUR, EVALUATE
AE	ENGINE REFERENCE AREA (M**2),(FT**2) (RS)	PARAM,SMBTUR, EVALUATE
TEMP	5 ELEMENT PARAMETER FOR SAVING INPUT VALUES (RS)	PARAM, EVALUATE

MEMBERS - NONE

FUNCTIONS

1. LOAD SFIELD DATA UNIT IF NOT ASSIGNED.
2. INITIALIZE DEFAULT VALUES OF USER PARAMETERS IF NOT DEFINED.
3. COMPUTE DIMENSIONLESS INPUTS TO MODULE SMBTUR.
4. EXECUTE MODULE SMBTUR.

MODULES CALLED - SMBTUR

PROCEDURES CALLED - XSMBTUR

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (DSTNJET)

****PROCEDURE DSTNJET

PURPOSE - TO ALLOW EXECUTION OF THE STONE JET NOISE MODULE WITH DIMENSIONAL INPUTS. USEFUL WHEN EXECUTION OF THE JET NOISE PARAMETERS MODULE IS NOT DESIRED.

AUTHOR - DSW(L03/00/05)
MMF(L03/00/09)

INPUT
USER PARAMETERS

		MODULES
CA	AMBIENT SPEED OF SOUND (M/S),(FT/S) (RS) (DEFAULT=340.294 M/S)	IF,EVALUATE, STNJET,PARAM
RHOA	AMBIENT DENSITY (KG/M**3),(SLUG/FT**3) (RS) (DEFAULT=1.225 KG/M**3)	IF,EVALUATE, STNJET,PARAM
TA	AMBIENT TEMPERATURE (DEG K),(DEG R) (RS) (DEFAULT=288.15 DEG K)	IF,PARAM, EVALUATE
VO	AIRCRAFT VELOCITY (M/S),(FT/S) (RS) (DEFAULT=0.)	IF,PARAM, EVALUATE
DELTA	ANGLE BETWEEN FLIGHT VECTOR AND ENGINE INLET AXIS (DEGREES) (RS) (DEFAULT=0.)	STNJET
RS	DISTANCE FROM NOZZLE EXIT TO OBSERVER (M),(FT) (RS) (DEFAULT=1. M)	IF,PARAM, STNJET
NENG	NUMBER OF ENGINES (I) (DEFAULT=1)	STNJET
A1	PRIMARY JET AREA (M**2),(FT**2) (RS) (DEFAULT=PI/4. M**2)	IF,EVALUATE, STNJET,PARAM
DE1	PRIMARY STREAM EQUIVALENT DIAMETER (M),(FT) (RS) (DEFAULT=1. M)	IF,EVALUATE, STNJET,PARAM
DH1	PRIMARY STREAM HYDRAULIC DIAMETER (M),(FT) (RS) (DEFAULT=1. M)	IF,EVALUATE, STNJET,PARAM
V1	PRIMARY JET VELOCITY (M/S),(FT/S) (RS) (DEFAULT=340.294 M/S)	IF,EVALUATE, STNJET,PARAM
RHO1	PRIMARY JET DENSITY (KG/M**3), (SLUG/FT**3) (RS) (DEFAULT=1.225 KG/M**3)	IF,EVALUATE, STNJET,PARAM
T1	PRIMARY JET TOTAL TEMPERATURE (DEG K), (DEG R) (RS) (DEFAULT=288.15 DEG K)	IF,EVALUATE, STNJET,PARAM
A2	SECONDARY JET AREA (M**2),(FT**2) (RS) (DEFAULT=0.)	IF,EVALUATE, STNJET,PARAM
V2	SECONDARY JET VELOCITY (M/S) (FT/S) (RS) (DEFAULT=0.)	IF,EVALUATE, STNJET,PARAM
RHO2	SECONDARY JET DENSITY (KG/M**3), (SLUG/FT**3) (RS) (DEFAULT=1.225 KG/M**3)	IF,EVALUATE, STNJET,PARAM
T2	SECONDARY JET TOTAL TEMPERATURE (DEG K),(DEG R) (RS) (DEFAULT=288.15 DEG K)	IF,EVALUATE, STNJET,PARAM
STIME	SOURCE TIME AT WHICH NOISE IS CALCULATED (S) (DEFAULT=0.)	STNJET

PROCEDURE PROCLIB (DSTNJET)

```
 SCRXXX  THREE LETTER ID USED TO FORM TABLE NAME   STNJET
         STNJET(XXXNNN) (A) (DEFAULT=3HXXX)
 SCRNNN  VALUE USED TO FORM TABLE NAME             STNJET
         STNJET(XXXNNN) (I) (DEFAULT=1)
 IOUT    OUTPUT CODE FOR TABLE AND PRINTED OUTPUT  STNJET
         (I) (DEFAULT=3)
         0 - NO PRINT BUT GENERATE STNJET(XXXNNN).
        -1 - OUTPUT PRINT IN DB BUT NO TABLE.
        -2 - OUTPUT PRINT IN DIMENSIONLESS FORM
             BUT NO TABLE.
        -3 - OUTPUT PRINT IN DIMENSIONLESS FORM
             AND DB BUT NO TABLE.
         1 - OUTPUT PRINT IN DB AND GENERATE
             STNJET(XXXNNN).
         2 - OUTPUT PRINT IN DIMENSIONLESS FORM
             AND GENERATE STNJET(XXXNNN).
         3 - OUTPUT PRINT IN DIMENSIONLESS FORM
             AND DB AND GENERATE STNJET(XXXNNN).
 IPRINT  OUTPUT PRINT CODE (I) (DEFAULT=3)         STNJET
         0 - NO PRINT
         1 - INPUT PRINT ONLY
         2 - OUTPUT PRINT ONLY
         3 - BOTH INPUT AND OUTPUT PRINT
 IUNITS  UNIT OPTION CODE (A) (DEFAULT=2HSI)     IF,PARAM,
         2HSI      - SI UNITS                      STNJET
         7HENGLISH - ENGLISH UNITS
 PLUG    NOZZLE PLUG OPTION (L) (DEFAULT=.FALSE.)  STNJET
         .TRUE.  - CONICAL PLUG NOZZLE
         .FALSE. - NO NOZZLE PLUG
 CIRCLE  NOZZLE TYPE OPTION (L) (DEFAULT=.FALSE.)  STNJET
         .TRUE.  - CIRCULAR NOZZLE
         .FALSE. - COAXIAL NOZZLE
 SUPER   SHOCK NOISE OPTION (L) (DEFAULT=.FALSE.)  STNJET
         .TRUE.  - COMPUTE SHOCK NOISE FOR
                   PRIMARY STREAM
         .FALSE. - DO NOT COMPUTE SHOCK NOISE

MEMBERS                                           MODULES

 SFIELD (FREQ ) - ONE RECORD MEMBER OF THE VALUES  STNJET
                  OF FREQUENCY IN HERTZ.
 SFIELD (THETA) - ONE RECORD MEMBER OF THE VALUES  STNJET
                  OF POLAR DIRECTIVITY ANGLE IN
                  DEGREES.
 SFIELD (PHI  ) - ONE RECORD MEMBER OF THE VALUES  STNJET
                  OF AZIMUTHAL DIRECTIVITY ANGLE
                  IN DEGREES.
 STNTBL (JDF  ) - TYPE 1 DATA TABLE OF THE JET     STNJET
                  MIXING NOISE SPECTRAL
                  DISTRIBUTION FACTOR.
```

PROCEDURE PROCLIB (DSTNJET)

STNTBL (SDF)	- TYPE 1 DATA TABLE OF THE SHOCK FACTOR.	STNJET
STNTBL (FSP)	- TYPE 1 DATA TABLE OF THE FREQUENCY SHIFT PARAMETER.	STNJET

OUTPUT
USER PARAMETERS

		MODULES
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS)	STNJET

MEMBERS

		MODULES
STNJET(XXXNNN)	- TYPE 1 DATA TABLE OF THE MEAN-SQUARE ACOUSTIC PRESSURE.	STNJET

LOCAL
USER PARAMETERS

		MEMBERS
MA	AIRCRAFT MACH NUMBER (RS)	STNJET, EVALUATE
M1	PRIMARY JET MACH NUMBER (RS)	STNJET, EVALUATE
M2	SECONDARY JET MACH NUMBER (RS)	STNJET, EVALUATE
AE	ENGINE REFERENCE AREA (M**2),(FT**2) (RS)	EVALUATE, STNJET,PARAM
TEMP	10 ELEMENT PARAMETER FOR SAVING INPUT VALUES (RS)	EVALUATE, PARAM

MEMBERS - NONE

FUNCTIONS

1. LOAD SFIELD AND STNTBL DATA UNITS IF NOT ASSIGNED.
2. INITIALIZE DEFAULT VALUES OF USER PARAMETERS IF NOT DEFINED.
3. COMPUTE DIMENSIONLESS INPUTS TO MODULE STNJET.
4. EXECUTE MODULE STNJET.

MODULES CALLED - STNJET

PROCEDURES CALLED - XSTNJET

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (ICA078)

****PROCEDURE ICA078

PURPOSE - TO LOAD THE UNIT MEMBERS AND SET THE PARAMETERS REQUIRED TO IMPLEMENT THE ICAO REFERENCE PROCEDURE.

AUTHOR - DSW(L03/00/05)

INPUT - NONE

OUTPUT

USER PARAMETERS

		MODULES
ICA078	LOGICAL PARAMETER TO IMPLEMENT ICAO PROCEDURE METHODS IN NOISE MODULES (=.TRUE.)	PARAM
SAE	LOGICAL PARAMETER TO REQUEST SAE ATMOSPHERIC ABSORPTION METHOD(=.TRUE.)	PARAM

MEMBERS

SAE(OM) SAE(PDF) SAE(NDF) SAE(MTH) SAE(SJC) SAE(SCF)	- TYPE 1 DATA TABLES WITH ICAO REVISIONS REQUIRED FOR SINGLE STREAM CIRCULAR JET NOISE MODULE (SGLJET)	LOAD

LOCAL - NONE

MODULES CALLED - NONE

PROCEDURES CALLED - NONE

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (LISTALL)

****PROCEDURE LISTALL

PURPOSE - TO PROVIDE A LISTING OF ALL PROCEDURE MEMBERS IN THE PROCEDURE LIBRARY.

AUTHOR - DSW(L03/00/00)

INPUT - NONE

OUTPUT - NONE

LOCAL - NONE

FUNCTION

1. PROVIDE LISTING OF PROCEDURE LIBRARY USING MEMLIST CONTROL STATEMENTS.

MODULES CALLED - NONE

PROCEDURES CALLED - NONE

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (LISTONE)

****PROCEDURE LISTONE

PURPOSE - TO PROVIDE A LISTING OF A SINGLE PROCEDURE MEMBER
IN THE PROCEDURE LIBRARY.

AUTHOR - DSW(L03/00/00)

INPUT

USER PARAMETERS - NONE

MEMBERS		MODULES
PROCLIB(PROC)	PROCEDURE MEMBER TO BE LISTED.	MEMLIST

OUTPUT - NONE

LOCAL - NONE

FUNCTION

1. PROVIDE LISTING OF A USER SPECIFIED PROCEDURE LIBRARY MEMBER USING A MEMLIST CONTROL STATEMENT.

MODULES CALLED - NONE

PROCEDURES CALLED - NONE

CALLING PROCEDURES - NONE

```
****PROCEDURE PROMUL

      PURPOSE - TO PROPAGATE A SET OF NOISE SOURCES AND THEIR SUM
                IN THE SAME COORDINATE SYSTEM TO AN ARRAY
                OF OBSERVERS AND COMPUTE NOISE SPECTRA, NOISE
                LEVELS, AND EFFECTIVE PERCEIVED NOISE LEVELS AT THE
                OBSERVER.

      AUTHOR  - DSW(L03/00/00)
                WKB(L03/00/01)
                DSW(L03/00/05)
                MMF(L03/00/09)

      INPUT
        USER PARAMETERS                                  MODULES

          NCOMP   NUMBER OF NOISE COMPONENTS TO BE        IF,PARAM
                  PROPAGATED (I) (DEFAULT=2)
          PROPRT  PRINT OPTION CODE FOR PRO MODULE (I)      PRO
                  =0  NO PRINTED OUTPUT
                  =1  PRINT INPUT DATA ONLY
                  =2  PRINT OUTPUT DATA ONLY
                  =3  PRINT INPUT AND OUTPUT DATA (DEFAULT)
          IOUT    OUTPUT OPTION CODE (I)                    PRO
                  =1  PRINT OUTPUT AS SOUND PRESSURE
                      LEVEL, SPL, IN DECIBELS
                  =2  PRINT OUTPUT AS DIMENSIONLESS
                      MEAN-SQUARE PRESSURE (DEFAULT)
                  =3  PRINT OUTPUT IN BOTH FORMS
          SIGMA   SPECIFIC FLOW RESISTANCE OF THE GROUND    PRO
                  (KG/(S M**3)),(SLUG/(S FT**3)) (RS)
                  (DEFAULT=2.5E5  KG/(S M**3))
          IUNITS  UNITS OPTION CODE (A) (DEFAULT=2HSI)     PRO,LEV
                  =2HSI       INPUTS ARE IN SI UNITS
                  =7HENGLISH INPUTS ARE IN ENGLISH UNITS
          NBAND   NUMBER OF SUBBANDS PER ONE-THIRD OCTAVE   PRO
                  BAND AS SPECIFIED FOR GRA (I) (DEFAULT=5)
          RS      SOURCE RADIUS (M),(FT) (RS) (DEFAULT=1M)  PRO
          SURFACE TYPE OF SURFACE TO BE USED IN COMPUTING   PRO
                  CHIEN-SOROKA GROUND EFFECTS
                  =4HSOFT SOFT GROUND SURFACE (DEFAULT)
                  =4HHARD HARD GROUND SURFACE
          ABSORP  ATMOSPHERIC ABSORPTION OPTION (L)         PRO
                  =.TRUE.  INCLUDE ABSORPTION EFFECTS
                  =.FALSE. DO NOT INCLUDE (DEFAULT)
          GROUND  GROUND EFFECTS OPTION (L)                 PRO
                  =.TRUE.  INCLUDE CHIEN-SOROKA GROUND
                           EFFECTS
                  =.FALSE. DO NOT INCLUDE (DEFAULT)
```

PROCEDURE PROCLIB (PROMUL)

PROTIME	THREE LETTER ID (XXX) TO APPEND TO 3HFLI TO DEFINE THE UNIT MEMBER FLI(FLIXXX) AND THE NOISE DATA UNIT MEMBER YYYYYY(XXXNNN). (DEFAULT=3HXXX)	PRO
PROSUM	CONTAINS THE NAMES OF THE NOISE UNITS TO BE INDIVIDUALLY PROPAGATED AND SUMMED AND PROPAGATED TO THE OBSERVER. (DEFAULT=6HYYYYYY)	PRO
IAWT	A-WEIGHTED SOUND PRESSURE LEVEL OPTION (L) =.TRUE. COMPUTE A-WEIGHTED SPL =.FALSE. DO NOT COMPUTE (DEFAULT)	LEV, IF,PARAM
IDWT	D-WEIGHTED SOUND PRESSURE LEVEL OPTION (L) =.TRUE. COMPUTE D-WEIGHTED SPL =.FALSE. DO NOT COMPUTE (DEFAULT)	LEV, IF,PARAM
IOSPL	OVERALL SOUND PRESSURE LEVEL OPTION (L) =.TRUE. COMPUTE OASPL =.FALSE. DO NOT COMPUTE (DEFAULT)	LEV, IF,PARAM
IPNL	PERCEIVED NOISE LEVEL OPTION (L) =.TRUE. COMPUTE PNL =.FALSE. DO NOT COMPUTE (DEFAULT)	LEV, IF,PARAM
IPNLT	TONE CORRECTED PERCEIVED NOISE LEVEL OPTION (L) =.TRUE. COMPUTE PNLT (DEFAULT) =.FALSE. DO NOT COMPUTE	LEV, IF,PARAM
LEVPRT	PRINT OPTION CODE FOR LEV MODULE (I) =0 NO PRINTED OUTPUT =1 PRINT INPUT DATA ONLY =2 PRINT OUTPUT DATA ONLY =3 PRINT INPUT AND OUTPUT DATA (DEFAULT)	LEV
IEPNL	EFFECTIVE PERCEIVED NOISE LEVEL OPTION (L) =.TRUE. COMPUTE EPNL (DEFAULT) =.FALSE. DO NOT COMPUTE	IF, PARAM
DTIME	RECEPTION TIME INCREMENT (S) (RS) (DEFAULT=0.5)	EFF
EFFPRT	PRINT OPTION CODE FOR EFF MODULE (I) =0 NO PRINTED OUTPUT =1 PRINT INPUT DATA ONLY =2 PRINT OUTPUT DATA ONLY =3 PRINT INPUT AND OUTPUT DATA (DEFAULT)	EFF

MEMBERS			MODULES
ATM	(TMOD)	TYPE 1 TABLE OF ATMOSPHERIC DATA.	PRO
ATM	(AAC)	TYPE 1 TABLE OF ATMOSPHERIC ABSORPTION DATA.	PRO
GEO	(GEOM)	MULTI-RECORD MEMBER OF GEOMETRY DATA.	PRO
FLI	(FLIXXX)	MULTI-RECORD MEMBER OF FLIGHT DATA. XXX IS DEFINED BY THE PROTIME PARAMETER.	PRO

PROCEDURE PROCLIB (PROMUL)

```
    YYYYYY(XXXNNN) TYPE 1 TABLE OF MEAN-SQUARE          PRO
                   ACOUSTIC PRESSURE. YYYYYY IS NOISE
                   MODULE(S) DEFINED BY PROSUM AND
                   NNN IS THE SOURCE TIME NUMBER.
    OBSERV(COORD ) MULTI-RECORD MEMBER OF OBSERVER    LEV,EFF
                   COORDINATES.
    SFIELD(FREQ  ) ONE RECORD MEMBER OF THE VALVES      LEV
                   OF FREQUENCY IN HERTZ.

OUTPUT
  USER PARAMETERS - NONE

  MEMBERS                                            MODULES

    PRO   (PRES  ) MULTI-RECORD MEMBER OF THE MEAN-     PRO
                   SQUARE ACOUSTIC PRESSURE AT THE
                   OBSERVERS.
                   (NOTE: OVERRIDE OF THIS NAME ON
                   THE CALL CONTROL STATEMENT MUST
                   BE DONE AS:

                   PRO=NEWNAME PRES=NEWNAME
                   6HPRO   =6HNEWNAM 6HPRES  =
                   6HNEWNAM

                   TO CORRECTLY CHANGE UNIT MEMBER
                   NAME AND VALUE OF LOCAL
                   PARAMETER MEMSUM)
    LEV   (PNLT  ) MULTI-RECORD MEMBER OF PNLT AT       LEV
                   THE OBSERVERS.
    EFF   (EPNL  ) MULTI-RECORD MEMBER OF EPNL AT       EFF
                   THE OBSERVERS.
LOCAL
  USER PARAMETERS                                    MODULES

    MEMSUM  UNIT MEMBER NAME OF LEV INPUT MEMBER,       LEV
            HAS VALUE 6HPRO    6HPRES  .

  MEMBERS                                            MODULES

    SCRATCH(XXXNNN) MEMBER CONTAINING RESULT OF NOISE  PRO
                    TABLE SUMMING.
FUNCTIONS

  1. CALL PROCLIB(XPROSGL) TO PROPAGATE SUM OF NOISE SOURCES
     TO OBSERVERS.
  2. INITIALIZE LOOP TO PROPAGATE INDIVIDUAL NOISE SOURCES.
  3. CALL PROCLIB(XPROSGL) TO PROPAGATE INDIVIDUAL NOISE
     SOURCES TO THE OBSERVERS.
```

PROCEDURE PROCLIB (PROMUL)

MODULES CALLED - EFF LEV PRO

PROCEDURES CALLED - PROSGL XEFF XLEV XPRO XPROSGL

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (PROSGL)

```
****PROCEDURE PROSGL

       PURPOSE - TO PROPAGATE A SINGLE NOISE SOURCE OR ONE SUM OF
                 NOISE SOURCES IN THE SAME COORDINATE SYSTEM TO AN
                 ARRAY OF OBSERVERS AND COMPUTE NOISE SPECTRA, NOISE
                 LEVELS, AND EFFECTIVE PERCEIVED NOISE LEVELS AT THE
                 OBSERVER.

       AUTHOR  - DSW(L03/00/00)
                 WKB(L03/00/01)
                 DSW(L03/00/05)
                 MMF(L03/00/09)

       INPUT
         USER PARAMETERS                                    MODULES

           PROPRT  PRINT OPTION CODE FOR PRO MODULE (I)       PRO
                   =0  NO PRINTED OUTPUT
                   =1  PRINT INPUT DATA ONLY
                   =2  PRINT OUTPUT DATA ONLY
                   =3  PRINT INPUT AND OUTPUT DATA (DEFAULT)
           IOUT    OUTPUT OPTION CODE (I)                     PRO
                   =1  PRINT OUTPUT AS SOUND PRESSURE
                       LEVEL, SPL, IN DECIBELS
                   =2  PRINT OUTPUT AS DIMENSIONLESS
                       MEAN-SQUARE PRESSURE (DEFAULT)
                   =3  PRINT OUTPUT IN BOTH FORMS
           SIGMA   SPECIFIC FLOW RESISTANCE OF THE GROUND     PRO
                   (KG/(S M**3)),(SLUG/(S FT**3)) (RS)
                   (DEFAULT=2.5E5  KG/(S M**3))
           IUNITS  UNITS OPTION CODE (A) (DEFAULT=2HSI)     PRO,LEV
                   =2HSI      INPUTS ARE IN SI UNITS
                   =7HENGLISH INPUTS ARE IN ENGLISH UNITS
           NBAND   NUMBER OF SUBBANDS PER ONE-THIRD OCTAVE    PRO
                   BAND AS SPECIFIED FOR GRA (I) (DEFAULT=5)
           RS      SOURCE RADIUS (M),(FT) (RS) (DEFAULT=1M)   PRO
           SURFACE TYPE OF SURFACE TO BE USED IN COMPUTING    PRO
                   CHIEN-SOROKA GROUND EFFECTS
                   =4HSOFT SOFT GROUND SURFACE (DEFAULT)
                   =4HHARD HARD GROUND SURFACE
           ABSORP  ATMOSPHERIC ABSORPTION OPTION (L)          PRO
                   =.TRUE.  INCLUDE ABSORPTION EFFECTS
                   =.FALSE. DO NOT INCLUDE (DEFAULT)
           GROUND  GROUND EFFECTS OPTION (L)                  PRO
                   =.TRUE.  INCLUDE CHIEN-SOROKA GROUND
                            EFFECTS
                   =.FALSE. DO NOT INCLUDE (DEFAULT)
```

PROCEDURE PROCLIB (PROSGL)

```
  PROTIME THREE LETTER ID (XXX) TO APPEND TO 3HFLI   PRO
          TO DEFINE THE UNIT MEMBER FLI(FLIXXX) AND
          THE NOISE DATA UNIT MEMBER YYYYYY(XXXNNN).
          (DEFAULT=3HXXX)
  PROSUM  CONTAINS THE NAME(S) OF THE NOISE UNIT(S)  PRO
          TO BE SUMMED (IF MORE THAN ONE )
          AND PROPAGATED TO THE OBSERVER.
          (DEFAULT=6HYYYYYY)
  IAWT    A-WEIGHTED SOUND PRESSURE LEVEL OPTION (L) LEV,
          =.TRUE.  COMPUTE A-WEIGHTED SPL           IF,PARAM
          =.FALSE. DO NOT COMPUTE (DEFAULT)
  IDWT    D-WEIGHTED SOUND PRESSURE LEVEL OPTION (L) LEV,
          =.TRUE.  COMPUTE D-WEIGHTED SPL           IF,PARAM
          =.FALSE. DO NOT COMPUTE (DEFAULT)
  IOSPL   OVERALL SOUND PRESSURE LEVEL OPTION (L)    LEV,
          =.TRUE.  COMPUTE OASPL                    IF,PARAM
          =.FALSE. DO NOT COMPUTE (DEFAULT)
  IPNL    PERCEIVED NOISE LEVEL OPTION (L)           LEV,
          =.TRUE.  COMPUTE PNL                      IF,PARAM
          =.FALSE. DO NOT COMPUTE (DEFAULT)
  IPNLT   TONE CORRECTED PERCEIVED NOISE LEVEL       LEV,
          OPTION (L)                                IF,PARAM
          =.TRUE.  COMPUTE PNLT (DEFAULT)
          =.FALSE. DO NOT COMPUTE
  LEVPRT  PRINT OPTION CODE FOR LEV MODULE (I)       LEV
          =0  NO PRINTED OUTPUT
          =1  PRINT INPUT DATA ONLY
          =2  PRINT OUTPUT DATA ONLY
          =3  PRINT INPUT AND OUTPUT DATA (DEFAULT)
  IEPNL   EFFECTIVE PERCEIVED NOISE LEVEL OPTION (L) IF,
          =.TRUE.  COMPUTE EPNL (DEFAULT)            PARAM
          =.FALSE. DO NOT COMPUTE
  DTIME   RECEPTION TIME INCREMENT (S) (RS)          EFF
          (DEFAULT=0.5)
  EFFPRT  PRINT OPTION CODE FOR EFF MODULE (I)       EFF
          =0  NO PRINTED OUTPUT
          =1  PRINT INPUT DATA ONLY
          =2  PRINT OUTPUT DATA ONLY
          =3  PRINT INPUT AND OUTPUT DATA (DEFAULT)
MEMBERS                                             MODULES

  ATM   (TMOD  ) TYPE 1 TABLE OF ATMOSPHERIC DATA.   PRO
  ATM   (AAC   ) TYPE 1 TABLE OF ATMOSPHERIC         PRO
                 ABSORPTION DATA.
  GEO   (GEOM  ) MULTI-RECORD MEMBER OF GEOMETRY     PRO
                 DATA.
```

PROCEDURE PROCLIB (PROSGL)

```
FLI   (FLIXXX) MULTI-RECORD MEMBER OF FLIGHT DATA. PRO
               XXX IS DEFINED BY THE PROTIME
               PARAMETER.
YYYYYY(XXXNNN) TYPE 1 TABLE OF MEAN-SQUARE         PRO
               ACOUSTIC PRESSURE. YYYYYY IS NOISE
               MODULE(S) DEFINED BY PROSUM AND
               NNN IS THE SOURCE TIME NUMBER.
OBSERV(COORD ) MULTI-RECORD MEMBER OF OBSERVER   LEV,EFF
               COORDINATES.
SFIELD(FREQ  ) ONE RECORD MEMBER OF THE VALVES     LEV
               OF FREQUENCY IN HERTZ.
```

OUTPUT
USER PARAMETERS - NONE

```
MEMBERS                                           MODULES

  PRO   (PRES  ) MULTI-RECORD MEMBER OF THE MEAN-    PRO
                 SQUARE ACOUSTIC PRESSURE AT THE
                 OBSERVERS.
                 (NOTE: OVERRIDE OF THIS NAME ON
                 THE CALL CONTROL STATEMENT MUST
                 BE DONE AS:

                 PRO=NEWNAME PRES=NEWNAME
                 6HPRO   =6HNEWNAM 6HPRES  =
                 6HNEWNAM

                 TO CORRECTLY CHANGE UNIT MEMBER
                 NAME AND VALUE OF LOCAL
                 PARAMETER MEMSUM)
  LEV   (PNLT  ) MULTI-RECORD MEMBER OF PNLT AT      LEV
                 THE OBSERVERS.
  EFF   (EPNL  ) MULTI-RECORD MEMBER OF EPNL AT      EFF
                 THE OBSERVERS.
```

LOCAL

```
USER PARAMETERS                                   MODULES

  MEMSUM  UNIT MEMBER NAME OF LEV INPUT MEMBER,      LEV
          HAS VALUE 6HPRO   6HPRES  .

MEMBERS                                           MODULES

  SCRATCH(XXXNNN) MEMBER CONTAINING RESULT OF NOISE  PRO
                  TABLE SUMMING.
```

PROCEDURE PROCLIB (PROSGL)

FUNCTIONS

1. LOAD SFIELD UNIT IF NOT ASSIGNED.
2. EXECUTE PRO MODULE.
3. EXECUTE LEV MODULE IF NOISE LEVELS ARE DESIRED.
4. EXECUTE EFF MODULE IF EPNL IS DESIRED.

MODULES CALLED - EFF LEV PRO

PROCEDURES CALLED - XEFF XLEV XPRO

CALLING PROCEDURES - XPROSGL

****PROCEDURE SDLDIR

```
PURPOSE - TO CONVERT A STANDARD TYPE 1 NOISE TABLE FROM POLAR
          DIRECTIVITY ANGLES TO SIDELINE DIRECTIVITY ANGLES.
          A TABLE IN THIS FORM IS USEFUL FOR COMPARISONS TO
          EXPERIMENTAL DATA WITH PROPAGATION EFFECTS REMOVED.

AUTHOR  - DSW(L03/00/05)
          MMF(L03/00/09)

INPUT
  USER PARAMETERS                                              MODULES

    SCRXXX  THREE LETTER ID USED TO FORM TABLE NAMES          GENSUP
            NOISE(XXXNNN) AND NOISES(XXXNNN) (A)
            (DEFAULT=3HXXX)
    SCRNNN  VALUE USED TO FORM TABLE NAMES                     GENSUP
            NOISE(XXXNNN) AND NOISES(XXXNNN) (I)
            (DEFAULT=1)
    IPRINT  OUTPUT PRINT CODE (I) (DEFAULT=3)                  GENSUP
            0 - NO PRINT
            1 - INPUT PRINT ONLY
            2 - OUTPUT PRINT ONLY
            3 - BOTH INPUT AND OUTPUT PRINT
    RHOA    AMBIENT DENSITY (KG/M**3),(SLUG/FT**3)             GENSUP
            (RS) (DEFAULT=1.225 KG/M**3)
    CA      AMBIENT SPEED OF SOUND (M/S),(FT/S)                GENSUP
            (RS) (DEFAULT=340.294 M/S)
    IOUT    OUTPUT CODE FOR TABLE AND PRINTED OUTPUT           GENSUP
            (I) (DEFAULT=3)
            0 - NO PRINT BUT GENERATE NOISES(XXXNNN).
           -1 - OUTPUT PRINT IN DB BUT NO TABLE.
           -2 - OUTPUT PRINT IN DIMENSIONLESS FORM
                BUT NO TABLE.
           -3 - OUTPUT PRINT IN DIMENSIONLESS FORM
                AND DB BUT NO TABLE.
            1 - OUTPUT PRINT IN DB AND GENERATE
                NOISES(XXXNNN).
            2 - OUTPUT PRINT IN DIMENSIONLESS FORM
                AND GENERATE NOISES(XXXNNN).
            3 - OUTPUT PRINT IN DIMENSIONLESS FORM
                AND DB AND GENERATE NOISES(XXXNNN).
    IUNITS  UNIT OPTION CODE (A) (DEFAULT=2HSI)                GENSUP
            2HSI      - SI UNITS
            7HENGLISH - ENGLISH UNITS
```

PROCEDURE PROCLIB (SDLDIR)

MEMBERS

		MODULES
NOISE (XXXNNN)	- TYPE 1 DATA TABLE OF THE MEAN-SQUARE ACOUSTIC PRESSURE AS PREDICTED BY ANY NOISE MODULE.	GENSUP

OUTPUT
USER PARAMETERS - NONE

MEMBERS

		MODULES
NOISES(XXXNN)	- TYPE 1 DATA TABLE OF THE MEAN-SQUARE ACOUSTIC PRESSURE IN TERMS OF SIDELINE DIRECTIVITY.	

LOCAL
USER PARAMETERS - NONE

MEMBERS

		MODULES
SDL (DIR)	- TYPE 1 DATA TABLE OF THE CONVERSION FACTOR FROM POLAR TO SIDELINE DIRECTIVITY ANGLE.	LOAD, GENSUP

FUNCTIONS

1. LOAD CONVERSION TABLE SDL(DIR) FROM DATA BASE.
2. EXECUTE GENSUP MODULE.

MODULES CALLED - GENSUP

PROCEDURES CALLED - XGENSUP

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (TCNLJET)

```
****PROCEDURE TCNLJET

       PURPOSE - TO COMPUTE JET EXHAUST NOISE USING MODULE CNLJET FOR
                 A TIME DEPENDENT PROBLEM WITH INPUT DATA ON ENGINE
                 STATE TABLES.

       AUTHOR  - DSW(L03/00/00)
                 DSW(L03/00/05)
                 MMF(L03/00/09)

       INPUT
         USER PARAMETERS
                                                                  MODULES
           AE      ENGINE REFERENCE AREA (M**2),(FT**2)           CNLJET,
                   (RS) (DEFAULT=PI/4. M**2)                      PREJET
           AP      PRIMARY NOZZLE PLUG AREA, RE AE (RS)           PREJET
                   (DEFAULT=0.)
           IRATIO  SPECIFIC HEATS OPTION (A)                      PREJET
                   8HCONSTANT - CONSTANT RATIO OF SPECIFIC
                                HEATS (DEFAULT)
                   8HVARIABLE - VARIABLE RATIO OF SPECIFIC
                                HEATS
           ITYPE   TYPE OF INTERPOLATION TO BE USED ON THE        PREJET
                   ENGINE STATE TABLES (I) (DEFAULT=1)
                   1 - LINEAR INTERPOLATION
                   2 - CUBIC SPLINE INTERPOLATION
           IUNITS  INPUT UNIT OPTION (A) (DEFAULT=2HSI)           PREJET,
                   2HSI      - SI UNITS                           CNLJET
                   7HENGLISH - ENGLISH UNITS
           PREPRT  OUTPUT PRINT OPTION FOR PREJET MODULE          PREJET
                   (I) (DEFAULT=3)
                   0 - NO PRINT
                   1 - INPUT PRINT ONLY
                   2 - OUTPUT PRINT ONLY
                   3 - BOTH INPUT AND OUTPUT PRINT
           RS      DISTANCE FROM NOZZLE EXIT TO OBSERVER          CNLJET
                   (M),(FT) (RS) (DEFAULT=SQRT(AE))
           DELTA   ANGLE BETWEEN FLIGHT VECTOR AND ENGINE         CNLJET
                   INLET AXIS (DEGREES) (RS) (DEFAULT=0.)
           NENG    NUMBER OF ENGINES (I) (DEFAULT=1)              CNLJET
           SCRXXX  THREE LETTER ID USED TO FORM MEMBER            CNLJET
                   NAME FLI(FLIXXX) AND CNLJET(XXXNNN) (A)        PREJET
                   (DEFAULT=3HXXX)
           IOUT    CODE FOR NOISE DATA OUTPUT (I)                 CNLJET
                   (DEFAULT=3)
                   0 - NO PRINT BUT GENERATE CNLJET(XXXNNN)
                  -1 - OUTPUT PRINT IN DB BUT NO TABLE.
                  -2 - OUTPUT PRINT IN DIMENSIONLESS FORM
                       BUT NO TABLE.
```

	-3 - OUTPUT PRINT IN DIMENSIONLESS FORM AND DB BUT NO TABLE.	
	1 - OUTPUT PRINT IN DB AND GENERATE CNLJET(XXXNNN).	
	2 - OUTPUT PRINT IN DIMENSIONLESS FORM AND GENERATE CNLJET(XXXNNN).	
	3 - OUTPUT PRINT IN DIMENSIONLESS FORM AND DB AND GENERATE CNLJET(XXXNNN).	
IPRINT	OUTPUT PRINT OPTION FOR CNLJET MODULE (I) (DEFAULT=3) 0 - NO PRINT 1 - INPUT PRINT ONLY 2 - OUTPUT PRINT ONLY 3 - BOTH INPUT AND OUTPUT PRINT	CNLJET

MEMBERS		MODULES
FLI (FLIXXX)	- MULTI-RECORD MEMBER OF SOURCE TIME DATA.	PREJET
ENG (PRIM)	- TYPE 1 DATA TABLE OF THE PRIMARY NOZZLE FLOW STATE.	PREJET
ENG (SEC)	- TYPE 1 DATA TABLE OF THE SECONDARY NOZZLE FLOW STATE.	PREJET
SFIELD (FREQ)	- ONE RECORD MEMBER OF THE VALUES OF FREQUENCY IN HERTZ.	CNLJET
SFIELD (THETA)	- ONE RECORD MEMBER OF THE VALUES OF POLAR DIRECTIVITY ANGLE IN DEGREES.	CNLJET
SFIELD (PHI)	- ONE RECORD MEMBER OF THE VALUES OF AZIMUTHAL DIRECTIVITY ANGLE IN DEGREES.	CNLJET
JWRCOAN(OM)	- TYPE 1 DATA TABLE OF THE JET VARIABLE DENSITY EXPONENT.	CNLJET
JWRCOAN(PDF)	- TYPE 1 DATA TABLE OF THE POWER DEVIATION FACTOR.	CNLJET
JWRCOAN(S1)	- TYPE 1 DATA TABLE OF THE PEAK STROUHAL NUMBER FOR THE FIRST SPECTRUM COMPONENT.	CNLJET
JWRCOAN(S2)	- TYPE 1 DATA TABLE OF THE PEAK STROUHAL NUMBER FOR THE SECOND SPECTRUM COMPONENT.	CNLJET
JWRCOAN(NSF)	- TYPE 1 DATA TABLE OF THE NORMALIZED SPECTRUM FUNCTION.	CNLJET
JWRCOAN(ALPHA)	- TYPE 1 DATA TABLE OF THE RATIO OF THE SECOND COMPONENT SPECTRAL PEAK TO THE FIRST COMPONENT SPECTRAL PEAK.	CNLJET
JWRCOAN(CBF)	- TYPE 1 DATA TABLE OF THE COANNULAR BENEFIT FACTOR.	CNLJET

PROCEDURE PROCLIB (TCNLJET)

JWRCOAN(DIR)	- TYPE 1 DATA TABLE OF THE NORMALIZED DIRECTIVITY FUNCTION.	CNLJET
JWRCOAN(MTH)	- TYPE 1 DATA TABLE OF THE FORWARD VELOCITY INDEX.	CNLJET

OUTPUT

USER PARAMETERS

		MODULES
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS)	CNLJET
CNLERR	ERROR PARAMETER (I) 0 - NO ERRORS ENCOUNTERED. 1 - MEMBER MANAGER ERROR OCCURRED ON OPEN. 2 - INSUFFICIENT LDS AVAILABLE. 3 - MEMBER MANAGER ERROR OCCURRED ON READ. 4 - AT LEAST ONE USER PARAMETER HAS AN INVALID VALUE. 5 - UNABLE TO BUILD TABLE CNLJET(XXXNNN). 6 - AN INVALID VALUE WAS CALCULATED DURING EXECUTION.	CNLJET

MEMBERS

		MODULES
CNLJET(XXX001) ... CNLJET(XXXNNN)	- NNN TYPE ONE DATA TABLES OF THE MEAN-SQUARE ACOUSTIC PRESSURE FOR NNN SOURCE TIMES.	CNLJET

LOCAL

USER PARAMETERS

		MODULES
NTIMES	NUMBER OF SOURCE TIME VALUES (I)	IF,PREJET
I	LOOP INDEX (I)	IF,PARAM
TIMES	MULTI-ELEMENT PARAMETER CONTAINING THE SOURCE TIMES (RS)	PREJET, PARAM
AT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET AREA FOR EACH TIME (RS)	PARAM, PREJET
MTA	MULTI-ELEMENT PARAMETER CONTAINING THE AIRCRAFT MACH NUMBER FOR EACH TIME (RS)	PARAM, PREJET
MT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET MACH NUMBER FOR EACH TIME (RS)	PREJET
TT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET TOTAL TEMPERATURE FOR EACH TIME (RS)	PARAM, PREJET
VT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET VELOCITY FOR EACH TIME (RS)	PARAM, PREJET
RSHT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET RATIO OF SPECIFIC HEATS FOR EACH TIME (RS)	PARAM, PREJET
RHOT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET DENSITY FOR EACH TIME (RS)	PARAM, PREJET

PROCEDURE PROCLIB (TCNLJET)

DET1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET EQUIVALENT DIAMETER FOR EACH TIME (RS)	PREJET
DHT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET HYDRAULIC DIAMETER FOR EACH TIME (RS)	PREJET
CTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT SPEED OF SOUND FOR EACH TIME (M/S),(FT/S) (RS)	PARAM, PREJET
RHOTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DENSITY FOR EACH TIME (KG/M**3), (SLUG/FT**3) (RS)	PARAM, PREJET
AT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET AREA FOR EACH TIME (RS)	PARAM, PREJET
MT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET MACH NUMBER FOR EACH TIME (RS)	PREJET
TT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET TOTAL TEMPERATURE FOR EACH TIME (RS)	PARAM, PREJET
VT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET VELOCITY FOR EACH TIME (RS)	PARAM, PREJET
RHOT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET DENSITY FOR EACH TIME (RS)	PARAM, PREJET
RSHT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET RATIO OF SPECIFIC HEATS FOR EACH TIME (RS)	PARAM, PREJET
DET2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET EQUIVALENT DIAMETER FOR EACH TIME (RS)	PREJET
DHT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET HYDRAULIC DIAMETER FOR EACH TIME (RS)	PARAM, PREJET
A1	VALUE OF PRIMARY JET AREA FOR TIME I (RS)	PARAM, CNLJET
T1	VALUE OF PRIMARY JET TOTAL TEMPERATURE FOR TIME I (RS)	PARAM, CNLJET
V1	VALUE OF PRIMARY JET VELOCITY FOR TIME I (RS)	PARAM, CNLJET
RHO1	VALUE OF PRIMARY JET DENSITY FOR TIME I (RS)	PARAM, CNLJET
GAMMA1	VALUE OF PRIMARY JET RATIO OF SPECIFIC HEATS FOR TIME I (RS)	PARAM, CNLJET
A2	VALUE OF SECONDARY JET AREA FOR TIME I (RS)	PARAM, CNLJET
DH2	VALUE OF SECONDARY JET HYDRAULIC DIAMETER FOR TIME I (RS)	PARAM, CNLJET
T2	VALUE OF SECONDARY JET TOTAL TEMPERATURE FOR TIME I (RS)	PARAM, CNLJET

PROCEDURE PROCLIB (TCNLJET)

V2	VALUE OF SECONDARY JET VELOCITY FOR TIME I (RS)	PARAM, CNLJET
RHO2	VALUE OF SECONDARY JET DENSITY FOR TIME I (RS)	PARAM, CNLJET
GAMMA2	VALUE OF SECONDARY JET RATIO OF SPECIFIC HEATS FOR TIME I (RS)	PARAM, CNLJET
CA	VALUE OF AMBIENT SPEED OF SOUND FOR TIME I (M/S),(FT/S) (RS)	PARAM, CNLJET
RHOA	VALUE OF AMBIENT DENSITY FOR TIME I (KG/M**3),(SLUG/FT**3) (RS)	PARAM, CNLJET
MA	VALUE OF AIRCRAFT MACH NUMBER FOR TIME I (RS)	PARAM, CNLJET
SCRNNN	INDEX FOR SOURCE MEMBER NAME = I (I)	PARAM, CNLJET
STIME	VALUE OF SOURCE TIME FOR TIME I (RS)	PARAM, CNLJET

MEMBERS - NONE

FUNCTIONS

1. LOAD JWRCOAN AND SFIELD DATA UNITS IF NOT ASSIGNED.
2. EXECUTE PREJET MODULE TO GET JET NOISE PARAMETERS.
3. INITIALIZE SOURCE TIME LOOP.
4. EXECUTE CNLJET MODULE FOR EACH SOURCE TIME ON FLI(FLIXXX) UNIT MEMBER.

MODULES CALLED - PREJET CNLJET

PROCEDURES CALLED - XPREJET XCNLJET

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (TCNLJET)

DET1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET EQUIVALENT DIAMETER FOR EACH TIME (RS)	PREJET
DHT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET HYDRAULIC DIAMETER FOR EACH TIME (RS)	PREJET
CTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT SPEED OF SOUND FOR EACH TIME (M/S),(FT/S) (RS)	PARAM, PREJET
RHOTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DENSITY FOR EACH TIME (KG/M**3), (SLUG/FT**3) (RS)	PARAM, PREJET
AT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET AREA FOR EACH TIME (RS)	PARAM, PREJET
MT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET MACH NUMBER FOR EACH TIME (RS)	PREJET
TT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET TOTAL TEMPERATURE FOR EACH TIME (RS)	PARAM, PREJET
VT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET VELOCITY FOR EACH TIME (RS)	PARAM, PREJET
RHOT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET DENSITY FOR EACH TIME (RS)	PARAM, PREJET
RSHT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET RATIO OF SPECIFIC HEATS FOR EACH TIME (RS)	PARAM, PREJET
DET2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET EQUIVALENT DIAMETER FOR EACH TIME (RS)	PREJET
DHT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET HYDRAULIC DIAMETER FOR EACH TIME (RS)	PARAM, PREJET
A1	VALUE OF PRIMARY JET AREA FOR TIME I (RS)	PARAM, CNLJET
T1	VALUE OF PRIMARY JET TOTAL TEMPERATURE FOR TIME I (RS)	PARAM, CNLJET
V1	VALUE OF PRIMARY JET VELOCITY FOR TIME I (RS)	PARAM, CNLJET
RHO1	VALUE OF PRIMARY JET DENSITY FOR TIME I (RS)	PARAM, CNLJET
GAMMA1	VALUE OF PRIMARY JET RATIO OF SPECIFIC HEATS FOR TIME I (RS)	PARAM, CNLJET
A2	VALUE OF SECONDARY JET AREA FOR TIME I (RS)	PARAM, CNLJET
DH2	VALUE OF SECONDARY JET HYDRAULIC DIAMETER FOR TIME I (RS)	PARAM, CNLJET
T2	VALUE OF SECONDARY JET TOTAL TEMPERATURE FOR TIME I (RS)	PARAM, CNLJET

PROCEDURE PROCLIB (TFNKAFM)

**** PROCEDURE TFNKAFM

PURPOSE - TO COMPUTE AIRFRAME NOISE USING MODULE FNKAFM FOR A TIME DEPENDENT PROBLEM.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)
MMF(L03/00/09)

INPUT
USER PARAMETERS

		MODULES
AF	FLAP AREA (M**2),(FT**2) (RS) (DEFAULT=10. M**2)	FNKAFM
AH	HORIZONTAL TAIL AREA (M**2),(FT**2) (RS) (DEFAULT=20. M**2)	FNKAFM
AV	VERTICAL TAIL AREA (M**2),(FT**2) (RS) (DEFAULT=20. M**2)	FNKAFM
AW	WING AREA (M**2),(FT**2) (RS) (DEFAULT=100. M**2)	FNKAFM
BF	FLAP SPAN (M),(FT) (RS) (DEFAULT=5. M)	FNKAFM
BH	HORIZONTAL TAIL SPAN (M),(FT) (RS) (DEFAULT=10. M)	FNKAFM
BV	VERTICAL TAIL SPAN (M),(FT) (RS) (DEFAULT=10. M)	FNKAFM
BW	WING SPAN (M),(FT) (RS) (DEFAULT=20. M)	FNKAFM
DMG	TIRE DIAMETER OF MAIN LANDING GEAR (M), (FT) (RS) (DEFAULT=1. M)	FNKAFM
DNG	TIRE DIAMETER OF NOSE LANDING GEAR (M), (FT) (RS) (DEFAULT=1. M)	FNKAFM
LMG	MAIN LANDING GEAR STRUT LENGTH (M),(FT) (RS) (DEFAULT=3. M)	FNKAFM
LNG	NOSE LANDING GEAR STRUT LENGTH (M),(FT) (RS) (DEFAULT=3. M)	FNKAFM
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS) (DEFAULT=BW)	FNKAFM
NWMG	NUMBER OF WHEELS PER MAIN LANDING GEAR (I) (DEFAULT=4)	FNKAFM
NWNG	NUMBER OF WHEELS PER NOSE LANDING GEAR (I) (DEFAULT=2)	FNKAFM
NMG	NUMBER OF MAIN LANDING GEAR (I) (DEFAULT=2)	FNKAFM
NNG	NUMBER OF NOSE LANDING GEAR (I) (DEFAULT=1)	FNKAFM
NS	NUMBER OF SLOTS FOR TRAILING EDGE FLAPS (I) (DEFAULT=3)	FNKAFM
PREPRT	OUTPUT PRINT CODE FOR PREAFM MODULE (I) (DEFAULT=3) 0 - NO PRINT 1 - INPUT PRINT ONLY	PREAFM

```
        2 - OUTPUT PRINT ONLY
        3 - BOTH INPUT AND OUTPUT PRINT
IOUT    OUTPUT CODE FOR TABLE AND PRINTED OUTPUT   FNKAFM
        (I) (DEFAULT=3)
        0 - NO PRINT BUT GENERATE FNKAFM(XXXNNN).
       -1 - OUTPUT PRINT IN DB BUT NO TABLE.
       -2 - OUTPUT PRINT IN DIMENSIONLESS FORM
            BUT NO TABLE.
       -3 - OUTPUT PRINT IN DIMENSIONLESS FORM
            AND DB BUT NO TABLE.
        1 - OUTPUT PRINT IN DB AND GENERATE
            FNKAFM(XXXNNN).
        2 - OUTPUT PRINT IN DIMENSIONLESS FORM
            AND GENERATE FNKAFM(XXXNNN).
        3 - OUTPUT PRINT IN DIMENSIONLESS FORM
            AND DB AND GENERATE FNKAFM(XXXNNN).
IPRINT  OUTPUT PRINT CODE FOR FNKAFM MODULE (I)    FNKAFM
        (DEFAULT=3)
        0 - NO PRINT
        1 - INPUT PRINT ONLY
        2 - OUTPUT PRINT ONLY
        3 - BOTH INPUT AND OUTPUT PRINT
SCRXXX  THREE LETTER ID USED TO FORM MEMBER NAMES  PREAFM,
        FLI(FLIXXX) AND FNKAFM(XXXNNN) (A)         FNKAFM
        (DEFAULT=3HXXX)
DYNCLN  DESCRIPTION OF AIRCRAFT (L)                FNKAFM
        (DEFAULT=.FALSE.)
        .TRUE.  - AERODYNAMICALLY CLEAN AIRCRAFT
        .FALSE. - NON-AERODYNAMICALLY CLEAN
                  AIRCRAFT
DELTAW  TYPE OF WING PLANFORM (L)                  FNKAFM
        (DEFAULT=.FALSE.)
        .TRUE.  - DELTA-WINGED AIRCRAFT
        .FALSE. - RECTANGULAR-WINGED AIRCRAFT
IUNITS  UNIT OPTION CODE (A) (DEFAULT=2HSI)        FNKAFM,
        2HSI      - SI UNITS                       PREAFM
        7HENGLISH - ENGLISH UNITS

(THE NEXT SEVEN PARAMETERS HAVE THE FOLLOWING VALUES)
(       .TRUE.  - INCLUDE IN TOTAL PREDICTION       )
(       .FALSE. - DO NOT INCLUDE                    )

TEWN    TRAILING EDGE WING NOISE (L)               FNKAFM
        (DEFAULT=.TRUE.)
TEHTN   TRAILING EDGE HORIZONTAL TAIL NOISE (L)    FNKAFM
        (DEFAULT=.TRUE.)
TEVTN   TRAILING EDGE VERTICAL TAIL NOISE (L)      FNKAFM
        (DEFAULT=.TRUE.)
TEFN    TRAILING EDGE FLAP NOISE (L)               FNKAFM
        (DEFAULT=.FALSE.)
```

```
   LESN    LEADING EDGE SLAT NOISE (L)                  FNKAFM
           (DEFAULT=.FALSE.)
   MLGN    MAIN LANDING GEAR NOISE (L)                  FNKAFM
           (DEFAULT=.TRUE.)
   NLGN    NOSE LANDING GEAR NOISE (L)                  FNKAFM
           (DEFAULT=.TRUE.)

  MEMBERS                                               MODULES

   FLI  (FLIXXX) - MULTI-RECORD MEMBER OF SOURCE        PREAFM
                   TIME DATA.
   SFIELD(FREQ ) - ONE RECORD MEMBER OF THE VALUES      FNKAFM
                   OF FREQUENCY IN HERTZ.
   SFIELD(THETA) - ONE RECORD MEMBER OF THE VALUES      FNKAFM
                   OF POLAR DIRECTIVITY ANGLE IN
                   DEGREES.
   SFIELD(PHI  ) - ONE RECORD MEMBER OF THE VALUES      FNKAFM
                   OF AZIMUTHAL DIRECTIVITY ANGLE
                   IN DEGREES.

OUTPUT
  USER PARAMETERS                                       MODULES

   RS      DISTANCE FROM SOURCE TO OBSERVER (M),(FT)    FNKAFM
           (RS)

  MEMBERS                                               MODULES

   FNKAFM(XXX001) - NNN TYPE ONE DATA TABLES OF THE     FNKAFM
       ...          MEAN-SQUARE ACOUSTIC PRESSURE
   FNKAFM(XXXNNN)   FOR NNN SOURCE TIMES

LOCAL
  USER PARAMETERS                                       MODULES

   NTIMES  NUMBER OF SOURCE TIME VALUES (I)             IF,PREAFM
   I       LOOP INDEX (I)                               IF,PARAM
   TIMES   MULTI-ELEMENT PARAMETER CONTAINING THE       PARAM,
           SOURCE TIMES (RS)                            PREAFM
   LANDTG  MULTI-ELEMENT PARAMETER CONTAINING THE       PARAM,
           LANDING GEAR POSITION FOR EACH TIME (A)      PREAFM
   MTA     MULTI-ELEMENT PARAMETER CONTAINING THE       PARAM,
           AIRCRAFT MACH NUMBER FOR EACH TIME (RS)      PREAFM
   DELTATF MULTI-ELEMENT PARAMETER CONTAINING THE       PARAM,
           FLAP SETTING ANGLE FOR EACH TIME (RS)        PREAFM
   CTA     MULTI-ELEMENT PARAMETER CONTAINING THE       PARAM,
           AMBIENT SPEED OF SOUND FOR EACH TIME         PREAFM
           (M/S),(FT/S) (RS)
   RHOTA   MULTI-ELEMENT PARAMETER CONTAINING THE       PARAM,
           AMBIENT DENSITY FOR EACH TIME (KG/M**3),     PREAFM
           (SLUG/FT**3) (RS)
```

PROCEDURE PROCLIB (TFNKAFM)

MUTA	MULTI-ELEMENT PARAMETER CONTAINING THE DYNAMIC VISCOSITY FOR EACH TIME (KG/(M S)),(SLUG/(FT S)) (RS)	PARAM, PREAFM
LANDG	LANDING GEAR POSITION FOR TIME I (A)	PARAM, FNKAFM
MA	AIRCRAFT MACH NUMBER FOR TIME I (RS)	PARAM, FNKAFM
DELTAF	FLAP SETTING ANGLE FOR TIME I (RS)	PARAM, FNKAFM
STIME	SOURCE TIME FOR TIME I (RS)	PARAM, FNKAFM
CA	AMBIENT SPEED OF SOUND FOR TIME I (M/S), (FT/S) (RS)	PARAM, FNKAFM
RHOA	AMBIENT DENSITY FOR TIME I (KG/M**3), (SLUG/FT**3) (RS)	PARAM, FNKAFM
MUA	DYNAMIC VISCOSITY AT TIME I (KG/(M S)), (SLUG/(FT S)) (RS)	PARAM, FNKAFM
SCRNNN	INDEX FOR SOURCE MEMBER NAME = I (I)	PARAM, FNKAFM

MEMBERS - NONE

FUNCTIONS

1. LOAD SFIELD DATA UNIT IF NOT ASSIGNED.
2. EXECUTE PREAFM MODULE TO GET AIRFRAME NOISE PARAMETERS.
3. INITIALIZE SOURCE TIME LOOP.
4. EXECUTE FNKAFM MODULE FOR EACH SOURCE TIME ON FLI(FLIXXX) UNIT MEMBER.

MODULES CALLED - PREAFM FNKAFM

PROCEDURES CALLED - XPREAFM XFNKAFM

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (TGECOR)

****PROCEDURE TGECOR

PURPOSE - TO COMPUTE THE COMBUSTION NOISE USING MODULE GECOR FOR A TIME DEPENDENT PROBLEM WITH INPUT DATA ON ENGINE STATE TABLES.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)
MMF(L03/00/09)

INPUT

USER PARAMETERS

PARAMETER	DESCRIPTION	MODULES
AE	ENGINE REFERENCE AREA (M**2),(FT**2) (RS) (DEFAULT=0.7854 M**2)	GECOR
A	COMBUSTOR ENTRANCE AREA, RE AE (RS) (DEFAULT=1.)	GECOR
PREPRT	OUTPUT PRINT OPTION FOR PRECOR MODULE (I) (DEFAULT=3) 0 - NO PRINT 1 - INPUT PRINT ONLY 2 - OUTPUT PRINT ONLY 3 - BOTH INPUT AND OUTPUT PRINT	PRECOR
IUNITS	INPUT UNIT OPTION (A) (DEFAULT=2HSI) 2HSI - SI UNITS 7HENGLISH - ENGLISH UNITS	PRECOR, GECOR
ITYPE	TYPE OF INTERPOLATION TO BE USED ON THE ENGINE STATE TABLES (I) (DEFAULT=1) 1 - LINEAR INTERPOLATION 2 - CUBIC SPLINE INTERPOLATION	PRECOR
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS) (DEFAULT=SQRT(AE))	GECOR
NENG	NUMBER OF ENGINES (I) (DEFAULT=1)	GECOR
SCRXXX	THREE LETTER ID USED TO FORM MEMBER NAME FLI(FLIXXX) AND GECOR(XXXNNN) (A) (DEFAULT=3HXXX)	GECOR, PRECOR
IOUT	CODE FOR NOISE DATA OUTPUT (I) (DEFAULT=3) 0 - NO PRINT BUT GENERATE GECOR(XXXNNN). -1 - OUTPUT PRINT IN DB BUT NO TABLE. -2 - OUTPUT PRINT IN DIMENSIONLESS FORM BUT NO TABLE. -3 - OUTPUT PRINT IN DIMENSIONLESS FORM AND DB BUT NO TABLE. 1 - OUTPUT PRINT IN DB AND GENERATE GECOR(XXXNNN). 2 - OUTPUT PRINT IN DIMENSIONLESS FORM AND GENERATE GECOR(XXXNNN). 3 - OUTPUT PRINT IN DIMENSIONLESS FORM AND DB AND GENERATE GECOR(XXXNNN).	GECOR

PROCEDURE PROCLIB (TGECOR)

```
IPRINT  OUTPUT PRINT OPTION FOR GECOR MODULE        GECOR
        (I) (DEFAULT=3)
        0 - NO PRINT
        1 - INPUT PRINT ONLY
        2 - OUTPUT PRINT ONLY
        3 - BOTH INPUT AND OUTPUT PRINT
TDDELT  DESIGN TURBINE TEMPERATURE RISE,            GECOR
        RE AMBIENT TEMPERATURE (RS)
        (DEFAULT=0.5 )

MEMBERS                                             MODULES

FLI   (FLIXXX) - MULTI-RECORD MEMBER OF SOURCE      PRECOR
                 TIME DATA.
ENG   (CORE1 ) - TYPE 1 DATA TABLE OF THE CORE      PRECOR
                 ENTRANCE FLOW STATE DATA.
ENG   (CORE2 ) - TYPE 1 DATA TABLE OF THE CORE      PRECOR
                 EXIT FLOW STATE DATA.
SFIELD(FREQ  ) - ONE RECORD MEMBER OF THE VALUES    PRECOR
                 OF FREQUENCY IN HERTZ.
SFIELD(THETA ) - ONE RECORD MEMBER OF THE VALUES    PRECOR
                 OF POLAR DIRECTIVITY ANGLE IN
                 DEGREES.
SFIELD(PHI   ) - ONE RECORD MEMBER OF THE VALUES    PRECOR
                 OF AZIMUTHAL DIRECTIVITY ANGLE
                 IN DEGREES.

OUTPUT
USER PARAMETERS                                     MODULES

RS      RADIAL DISTANCE FROM SOURCE TO OBSERVER     GECOR
        (M),(FT) (RS)
GECERR  ERROR PARAMETER (I)
        0 - NO ERRORS ENCOUNTERED.
        1 - MEMBER MANAGER ERROR OCCURRED ON OPEN.
        2 - INSUFFICIENT LDS AVAILABLE.
        3 - MEMBER MANAGER ERROR OCCURRED ON READ.
        4 - AT LEAST ONE USER PARAMETER HAS AN
            INVALID VALUE.
        5 - UNABLE TO BUILD TABLE GECOR(XXXNNN).
        6 - AN INVALID VALUE WAS CALCULATED DURING
            EXECUTION.

MEMBERS                                             MODULES

GECOR (XXX001) - NNN TYPE 1 DATA TABLES OF THE      GECOR
   ...           MEAN-SQUARE ACOUSTIC PRESSURE
GECOR (XXXNNN)   FOR NNN SOURCE TIMES.
```

PROCEDURE PROCLIB (TGECOR)

LOCAL

USER PARAMETERS

USER PARAMETERS		MODULES
NTIMES	NUMBER OF SOURCE TIME VALUES (I)	IF,PRECOR
I	LOOP INDEX (I)	IF,PARAM
TIMES	MULTI-ELEMENT PARAMETER CONTAINING THE SOURCE TIMES (RS)	PARAM, PRECOR
MT1	MULTI-ELEMENT PARAMETER CONTAINING THE ENTRANCE MASS FLOW RATE FOR EACH TIME (RS)	PARAM, PRECOR
MTA	MULTI-ELEMENT PARAMETER CONTAINING THE AIRCRAFT MACH NUMBER FOR EACH TIME (RS)	PARAM, PRECOR
PT1	MULTI-ELEMENT PARAMETER CONTAINING THE COMBUSTOR ENTRANCE TOTAL PRESSURE FOR EACH TIME (RS)	PARAM, PRECOR
TT1	MULTI-ELEMENT PARAMETER CONTAINING THE COMBUSTOR ENTRANCE TOTAL TEMPERATURE FOR EACH TIME (RS)	PARAM, PRECOR
TT2	MULTI-ELEMENT PARAMETER CONTAINING THE COMBUSTOR EXIT TOTAL TEMPERATURE FOR EACH TIME (RS)	PARAM, PRECOR
CTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT SPEED OF SOUND FOR EACH TIME (RS)	PARAM, PRECOR
RHOTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DENSITY FOR EACH TIME (RS)	PARAM, PRECOR
STIME	SOURCE TIME VALUE FOR TIME I (RS)	PARAM, GECOR
MDOT	COMBUSTOR ENTRANCE MASS FLOW RATE FOR TIME I (RS)	PARAM, GECOR
MA	AIRCRAFT MACH NUMBER FOR TIME I (RS)	PARAM, GECOR
PI	COMBUSTOR ENTRANCE TOTAL PRESSURE FOR TIME I (RS)	PARAM, GECOR
TI	COMBUSTOR ENTRANCE TOTAL TEMPERATURE FOR TIME I (RS)	PARAM, GECOR
TJ	COMBUSTOR EXIT TOTAL TEMPERATURE FOR TIME I (RS)	PARAM, GECOR
CA	AMBIENT SPEED OF SOUND FOR TIME I (M/S), (FT/S) (RS)	PARAM, GECOR
RHOA	AMBIENT DENSITY FOR TIME I (KG/M**3), (SLUG/FT**3) (RS)	PARAM, GECOR
SCRNNN	INDEX FOR SOURCE MEMBER NAME = I (I)	PARAM, GECOR

MEMBERS - NONE

FUNCTIONS

1. LOAD SFIELD DATA UNIT IF NOT ASSIGNED.
2. EXECUTE PRECOR MODULE TO GET CORE NOISE PARAMETERS.

PROCEDURE PROCLIB (TGECOR)

3. INITIALIZE SOURCE TIME LOOP.
4. EXECUTE GECOR MODULE FOR EACH SOURCE TIME ON FLI(FLIXXX) UNIT MEMBER.

MODULES CALLED - PRECOR GECOR

PROCEDURES CALLED - XPRECOR XGECOR

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (TGETUR)

****PROCEDURE TGETUR

```
PURPOSE - TO COMPUTE THE TURBINE NOISE USING MODULE GETUR
          FOR A TIME DEPENDENT PROBLEM WITH INPUT DATA ON
          ENGINE STATE TABLES.

AUTHOR  - DSW(L03/00/00)
          WKB(L03/00/01)
          DSW(L03/00/05)
          MMF(L03/00/09)

INPUT
  USER PARAMETERS                                          MODULES

    AE      ENGINE REFERENCE AREA (M**2),(FT**2) (RS)   GETUR
            (DEFAULT=PI/4. M**2)
    NENG    NUMBER OF ENGINES (I) (DEFAULT=1)            GETUR
    RS      DISTANCE FROM SOURCE TO OBSERVER (M),(FT)    GETUR
            (RS) (DEFAULT=SQRT(AE))
    AREA    TURBINE INLET CROSS-SECTIONAL AREA, RE AE    GETUR
            (RS) (DEFAULT=1.)
    NBLADE  NUMBER OF ROTOR BLADES (I) (DEFAULT=20)      GETUR
    D       TURBINE ROTOR DIAMETER, RE SQRT(AE) (RS)     GETUR
            (DEFAULT=1.)
    SCRXXX  THREE LETTER ID USED TO FORM MEMBER NAME     GETUR,
            FLI(FLIXXX) AND GETUR(XXXNNN) (A)            PRETUR
            (DEFAULT=3HXXX)
    BROAD   BROADBAND NOISE OPTION (L)                   GETUR
            .TRUE.  COMPUTE BROADBAND NOISE (DEFAULT)
            .FALSE. DO NOT COMPUTE
    PURE    PURE TONE OPTION (L)                         GETUR
            .TRUE.  COMPUTE PURE TONES (DEFAULT)
            .FALSE. DO NOT COMPUTE
    IOUT    OUTPUT CODE FOR TABLE AND PRINTED OUTPUT     GETUR
            (I) (DEFAULT=3)
            0 - NO PRINT BUT GENERATE GETUR(XXXNNN).
           -1 - OUTPUT PRINT IN DB BUT NO TABLE.
           -2 - OUTPUT PRINT IN DIMENSIONLESS FORM
                BUT NO TABLE.
           -3 - OUTPUT PRINT IN DIMENSIONLESS FORM
                AND DB BUT NO TABLE.
            1 - OUTPUT PRINT IN DB AND GENERATE
                GETUR(XXXNNN).
            2 - OUTPUT PRINT IN DIMENSIONLESS FORM
                AND GENERATE GETUR(XXXNNN).
            3 - OUTPUT PRINT IN DIMENSIONLESS FORM
                AND DB AND GENERATE GETUR(XXXNNN).
```

PROCEDURE PROCLIB (TGETUR)

IPRINT	OUTPUT PRINT CODE FOR GETUR MODULE (I) (DEFAULT=3) 0 - NO PRINT 1 - INPUT PRINT ONLY 2 - OUTPUT PRINT ONLY 3 - BOTH INPUT AND OUTPUT PRINT	GETUR
IUNITS	INPUT UNIT OPTION (A) (DEFAULT=2HSI) 2HSI - SI UNITS 7HENGLISH - ENGLISH UNITS	GETUR, PRETUR
ITYPE	TYPE OF INTERPOLATION TO BE USED ON THE ENGINE STATE TABLES (I) (DEFAULT=1) 1 - LINEAR INTERPOLATION 2 - CUBIC SPLINE INTERPOLATION	PRETUR
IRATIO	SPECIFIC HEATS OPTION (A) 8HCONSTANT - CONSTANT RATIO OF SPECIFIC HEATS (DEFAULT) 8HVARIABLE - VARIABLE RATIO OF SPECIFIC HEATS	PRETUR
PREPRT	OUTPUT PRINT CODE FOR PRETUR MODULE (I) (DEFAULT=3) 0 - NO PRINT 1 - INPUT PRINT ONLY 2 - OUTPUT PRINT ONLY 3 - BOTH INPUT AND OUTPUT PRINT	PRETUR

MEMBERS		MODULES
FLI (FLIXXX)	- MULTI-RECORD MEMBER OF SOURCE TIME DATA.	PRETUR
ENG (TURBINE1)	- TYPE 1 DATA TABLE OF THE TURBINE ENTRANCE FLOW STATE.	PRETUR
ENG (TURBINE2)	- TYPE 1 DATA TABLE OF THE TURBINE EXIT FLOW STATE.	PRETUR
SFIELD(FREQ)	- ONE RECORD MEMBER OF THE VALUES OF FREQUENCY IN HERTZ.	GETUR
SFIELD(THETA)	- ONE RECORD MEMBER OF THE VALUES OF POLAR DIRECTIVITY ANGLE IN DEGREES.	GETUR
SFIELD(PHI)	- ONE RECORD MEMBER OF THE VALUES OF AZIMUTHAL DIRECTIVITY ANGLE IN DEGREES.	GETUR

OUTPUT

USER PARAMETERS		MODULES
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS)	GETUR

PROCEDURE PROCLIB (TGETUR)

GETERR	ERROR PARAMETER (I) 0 - NO ERRORS ENCOUNTERED. 1 - MEMBER MANAGER ERROR OCCURRED ON OPEN. 2 - INSUFFICIENT LDS AVAILABLE. 3 - MEMBER MANAGER ERROR OCCURRED ON READ. 4 - AT LEAST ONE USER PARAMETER HAS AN INVALID VALUE. 5 - UNABLE TO BUILD TABLE GETUR(XXXNNN) 6 - AN INVALID VALUE WAS CALCULATED DURING EXECUTION.	GETUR

MEMBERS

		MODULES
GETUR(XXX001) ... GETUR(XXXNNN)	- NNN TYPE 1 DATA TABLES OF THE MEAN-SQUARE ACOUSTIC PRESSURE FOR NNN SOURCE TIMES.	GETUR

LOCAL

USER PARAMETERS

		MODULES
NTIMES	NUMBER OF SOURCE TIME VALUES (I)	IF,PRETUR
I	LOOP INDEX (I)	IF,PARAM
TIMES	MULTI-ELEMENT PARAMETER CONTAINING THE SOURCE TIMES (RS)	PRETUR PARAM
MTA	MULTI-ELEMENT PARAMETER CONTAINING THE AIRCRAFT MACH NUMBER FOR EACH TIME (RS)	PARAM, PRETUR
MT1	MULTI-ELEMENT PARAMETER CONTAINING THE CORE MASS FLOW RATE FOR EACH TIME (RS)	PRETUR
NT1	MULTI-ELEMENT PARAMETER CONTAINING THE ROTATIONAL SPEED FOR EACH TIME (RS)	PARAM, PRETUR
TT1	MULTI-ELEMENT PARAMETER CONTAINING THE ENTRANCE TOTAL TEMPERATURE FOR EACH TIME (RS)	PARAM, PRETUR
TT2	MULTI-ELEMENT PARAMETER CONTAINING THE EXIT STATIC TEMPERATURE FOR EACH TIME (RS)	PARAM, PRETUR
CTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT SPEED OF SOUND FOR EACH TIME (M/S),(FT/S) (RS)	PARAM, PRETUR
RHOTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DENSITY FOR EACH TIME (KG/M**3), (SLUG/FT**3) (RS)	PARAM, PRETUR
FTA	MULTI-ELEMENT PARAMETER CONTAINING THE FUEL-TO-AIR RATIO FOR EACH TIME (RS)	PARAM, PRETUR
HTA	MULTI-ELEMENT PARAMETER CONTAINING THE ABSOLUTE HUMIDITY FOR EACH TIME (RS)	PARAM, PRETUR
MA	AIRCRAFT MACH NUMBER AT TIME I (RS)	PARAM, GETUR
ROTSPD	ROTATIONAL SPEED AT TIME I (RS)	PARAM, GETUR

PROCEDURE PROCLIB (TGETUR)

TTI	ENTRANCE TOTAL TEMPERATURE AT TIME I (RS)	PARAM, GETUR
TSJ	EXIT STATIC TEMPERATURE AT TIME I (RS)	PARAM, GETUR
CA	AMBIENT SPEED OF SOUND AT TIME I (M/S), (FT/S) (RS)	PARAM, GETUR
RHOA	AMBIENT DENSITY AT TIME I (KG/M**3), (SLUG/FT**3) (RS)	PARAM, GETUR
SCRNNN	INDEX FOR SOURCE MEMBER NAME = I (I)	PARAM, GETUR
STIME	SOURCE TIME VALUE AT TIME I (RS)	PARAM, GETUR
F	FUEL-TO-AIR RATIO AT TIME I (RS)	PARAM, GETUR
HA	ABSOLUTE HUMIDITY AT TIME I (RS)	PARAM, GETUR

MEMBERS - NONE

FUNCTIONS

1. LOAD SFIELD DATA UNIT IF NOT ASSIGNED.
2. EXECUTE PRETUR MODULE TO GET TURBINE NOISE PARAMETERS.
3. INITIALIZE SOURCE TIME LOOP.
4. EXECUTE GETUR MODULE FOR EACH SOURCE TIME ON FLI(FLIXXX) UNIT MEMBER.

MODULES CALLED - PRETUR GETUR

PROCEDURES CALLED - XPRETUR XGETUR

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (THDNFAN)

****PROCEDURE THDNFAN

PURPOSE - TO COMPUTE FAN NOISE USING MODULE HDNFAN FOR A TIME DEPENDENT PROBLEM WITH INPUT DATA ON ENGINE STATE TABLES.

AUTHOR - DSW(L03/00/00)
WKB(L03/00/01)
DSW(L03/00/05)
MMF(L03/00/09)

INPUT

USER PARAMETERS

USER PARAMETERS		MODULES
PREPRT	OUTPUT PRINT OPTION FOR PREFAN MODULE (I) (DEFAULT=3) 0 - NO PRINT 1 - INPUT PRINT ONLY 2 - OUTPUT PRINT ONLY 3 - BOTH INPUT AND OUTPUT PRINT	PREFAN
IUNITS	INPUT UNIT OPTION (A) (DEFAULT=2HSI) 2HSI - SI UNITS 7HENGLISH - ENGLISH UNITS	PREFAN, HDNFAN
ITYPE	TYPE OF INTERPOLATION TO BE USED ON THE ENGINE STATE TABLES (I) (DEFAULT=1) 1 - LINEAR INTERPOLATION 2 - CUBIC SPLINE INTERPOLATION	PREFAN
AE	ENGINE REFERENCE AREA (M**2),(FT**2) (RS) (DEFAULT=PI/4. M**2)	HDNFAN
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS) (DEFAULT=SQRT(AE))	HDNFAN
AFAN	FAN INLET CROSS-SECTIONAL AREA, RE AE (RS) (DEFAULT=1.)	HDNFAN
DIAM	FAN ROTOR DIAMETER, RE SQRT(AE) (RS) (DEFAULT=1.128)	HDNFAN
MD	FAN ROTOR RELATIVE TIP MACH NUMBER AT DESIGN POINT (RS) (DEFAULT=1.)	HDNFAN
RSS	ROTOR-STATOR SPACING, RE MEAN ROTOR BLADE CHORD (RS) (DEFAULT=1.)	HDNFAN
NBANDS	NUMBER OF 1/3 OCTAVE BANDS FOR TONE FREQUENCY SHIFT (I) (DEFAULT=0)	HDNFAN
NENG	NUMBER OF ENGINES (I) (DEFAULT=1)	HDNFAN
NB	NUMBER OF ROTOR BLADES (I) (DEFAULT=20)	HDNFAN
NV	NUMBER OF STATOR VANES (I) (DEFAULT=50)	HDNFAN
IGV	INLET GUIDE VANE INDEX (I) (DEFAULT=1) 1 - FOR A FAN WITHOUT IGV'S 2 - FOR A FAN WITH IGV'S	HDNFAN
DIS	INLET FLOW DISTORTION INDEX (I) 1 - NO INLET FLOW DISTORTION (DEFAULT) 2 - INLET FLOW DISTORTED	HDNFAN

PROCEDURE PROCLIB (THDNFAN)

```
IOUT    OUTPUT CODE FOR TABLE AND PRINTED OUTPUT    HDNFAN
        (I) (DEFAULT=3)
         0 - NO PRINT BUT GENERATE HDNFAN(XXXNNN).
        -1 - OUTPUT PRINT IN DB BUT NO TABLE.
        -2 - OUTPUT PRINT IN DIMENSIONLESS FORM
             BUT NO TABLE.
        -3 - OUTPUT PRINT IN DIMENSIONLESS FORM
             AND DB BUT NO TABLE.
         1 - OUTPUT PRINT IN DB AND GENERATE
             HDNFAN(XXXNNN).
         2 - OUTPUT PRINT IN DIMENSIONLESS FORM
             AND GENERATE HDNFAN(XXXNNN).
         3 - OUTPUT PRINT IN DIMENSIONLESS FORM
             AND DB AND GENERATE HDNFAN(XXXNNN).
IPRINT  OUTPUT PRINT OPTION FOR HDNFAN MODULE       HDNFAN
        (I) (DEFAULT=3)
        0 - NO PRINT
        1 - INPUT PRINT ONLY
        2 - OUTPUT PRINT ONLY
        3 - BOTH INPUT AND OUTPUT PRINT
SCRXXX  THREE LETTER ID USED TO FORM MEMBER NAME    HDNFAN,
        FLI(FLIXXX) AND HDNFAN(XXXNNN) (A)          PREFAN
        (DEFAULT=3HXXX)

(THE NEXT SIX PARAMETERS HAVE THE FOLLOWING VALUES (L)   )
(        .TRUE.  - INCLUDE IN TOTAL PREDICTION (DEFAULT))
(        .FALSE. - DO NOT INCLUDE                        )

 INRS    INLET ROTOR-STATOR INTERACTION TONES
 INCT    COMBINATION TONE NOISE
 INDIS   INLET FLOW DISTORTION TONES
 INBB    INLET BROADBAND NOISE
 IDRS    DISCHARGE ROTOR-STATOR INTERACTION TONES
 IDBB    DISCHARGE BROADBAND NOISE

                                                     MODULES
MEMBERS

 FLI    (FLIXXX) - MULTI-RECORD MEMBER OF SOURCE       PREFAN
                   TIME DATA.
 ENG    (FAN1  ) - TYPE 1 DATA TABLE OF THE FAN        PREFAN
                   ENTRANCE FLOW STATE.
 ENG    (FAN2  ) - TYPE 1 DATA TABLE OF THE FAN        PREFAN
                   EXIT FLOW STATE.
 SFIELD(FREQ   ) - ONE RECORD MEMBER OF THE VALUES     HDNFAN
                   OF FREQUENCY IN HERTZ.
 SFIELD(THETA  ) - ONE RECORD MEMBER OF THE VALUES     HDNFAN
                   OF POLAR DIRECTIVITY ANGLE IN
                   DEGREES.
```

PROCEDURE PROCLIB (THDNFAN)

SFIELD(PHI)	- ONE RECORD MEMBER OF THE VALUES OF AZIMUTHAL DIRECTIVITY ANGLE IN DEGREES.	HDNFAN

OUTPUT

USER PARAMETERS

		MODULES
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS)	HDNFAN

MEMBERS

		MODULES
HDNFAN(XXX001) ... HDNFAN(XXXNNN)	- NNN TYPE ONE DATA TABLES OF THE MEAN-SQUARE ACOUSTIC PRESSURE FOR NNN SOURCE TIMES.	HDNFAN

LOCAL

USER PARAMETERS

		MODULES
NTIMES	NUMBER OF SOURCE TIME VALUES (I)	IF,PREFAN
I	LOOP INDEX (I)	IF,PARAM
TIMES	MULTI-ELEMENT PARAMETER CONTAINING THE SOURCE TIMES (RS)	PARAM, PREFAN
MT1	MULTI-ELEMENT PARAMETER CONTAINING THE FAN MASS FLOW RATE FOR EACH TIME (RS)	PARAM, PREFAN
MTA	MULTI-ELEMENT PARAMETER CONTAINING THE AIRCRAFT MACH NUMBER FOR EACH TIME (RS)	PARAM, PREFAN
NT1	MULTI-ELEMENT PARAMETER CONTAINING THE ROTATIONAL SPEED FOR EACH TIME (RS)	PARAM, PREFAN
DT1	MULTI-ELEMENT PARAMETER CONTAINING THE FAN TOTAL TEMPERATURE RISE FOR EACH TIME (RS)	PARAM, PREFAN
MT2	MULTI-ELEMENT PARAMETER CONTAINING THE EXIT MASS FLOW RATE FOR EACH TIME (RS)	PREFAN
CTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT SPEED OF SOUND FOR EACH TIME (M/S),(FT/S) (RS)	PARAM, PREFAN
RHOTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DENSITY FOR EACH TIME (KG/M**3), (SLUG/FT**3) (RS)	PARAM, PREFAN
MDOT	VALUE OF MASS FLOW RATE FOR TIME I (RS)	PARAM, HDNFAN
MA	VALUE OF AIRCRAFT MACH NUMBER FOR TIME I (RS)	PARAM, HDNFAN
N	VALUE OF ROTATIONAL SPEED AT TIME I (RS)	PARAM, HDNFAN
DELTAT	VALUE OF FAN TOTAL TEMPERATURE RISE FOR TIME I (RS)	PARAM, HDNFAN
CA	VALUE OF AMBIENT SPEED OF SOUND FOR TIME I (M/S),(FT/S) (RS)	PARAM, HDNFAN

RHOA	VALUE OF AMBIENT DENSITY FOR TIME I (KG/M**3),(SLUG/FT**3) (RS)	PARAM, HDNFAN
SCRNNN	INDEX FOR SOURCE MEMBER NAME = I (I)	PARAM, HDNFAN
STIME	VALUE OF SOURCE TIME FOR TIME I (RS)	PARAM, HDNFAN

MEMBERS - NONE

FUNCTIONS

1. LOAD SFIELD DATA UNIT IF NOT ASSIGNED.
2. EXECUTE PREFAN MODULE TO GET FAN NOISE PARAMETERS.
3. INITIALIZE SOURCE TIME LOOP.
4. EXECUTE HDNFAN MODULE FOR EACH SOURCE TIME ON FLI(FLIXXX) UNIT MEMBER.

MODULES CALLED - PREFAN HDNFAN

PROCEDURES CALLED - XPREFAN XHDNFAN

CALLING PROCEDURES - NONE

```
****PROCEDURE TSAESHK

        PURPOSE - TO COMPUTE THE SHOCK CELL NOISE USING MODULE SAESHK
                  FOR A TIME DEPENDENT PROBLEM WITH INPUT DATA ON AN
                  ENGINE STATE TABLE.

        AUTHOR  - DSW(L03/00/00)
                  DSW(L03/00/05)
                  MMF(L03/00/09)

        INPUT
         USER PARAMETERS
                                                                 MODULES
           AE      ENGINE REFERENCE AREA (M**2),(FT**2)          PREJET,
                   (RS) (DEFAULT=PI/4. M**2)                     SAESHK
           AP      PRIMARY NOZZLE PLUG AREA, RE AE (RS)          PREJET
                   (DEFAULT=0.)
           DELTA   ANGLE BETWEEN FLIGHT VECTOR AND ENGINE        SAESHK
                   INLET AXIS (DEG) (RS) (DEFAULT=0.)
           IRATIO  SPECIFIC HEATS OPTION (A)                     PREJET
                   8HCONSTANT - CONSTANT RATIO OF SPECIFIC
                                HEATS (DEFAULT)
                   8HVARIABLE - VARIABLE RATIO OF SPECIFIC
                                HEATS
           ITYPE   TYPE OF INTERPOLATION TO BE USED ON THE       PREJET
                   ENGINE STATE TABLES (I) (DEFAULT=1)
                   1 - LINEAR INTERPOLATION
                   2 - CUBIC SPLINE INTERPOLATION
           IUNITS  INPUT UNIT OPTION (A) (DEFAULT=2HSI)          PREJET,
                   2HSI      - SI UNITS                          SAESHK
                   7HENGLISH - ENGLISH UNITS
           PREPRT  OUTPUT PRINT OPTION FOR PREJET MODULE         PREJET
                   (I) (DEFAULT=3)
                   0 - NO PRINT
                   1 - INPUT PRINT ONLY
                   2 - OUTPUT PRINT ONLY
                   3 - BOTH INPUT AND OUTPUT PRINT
           RS      RADIAL DISTANCE FROM NOZZLE EXIT TO           SAESHK
                   OBSERVER (M),(FT) (RS) (DEFAULT=SQRT(AE))
           NENG    NUMBER OF ENGINES (I) (DEFAULT=1)             SAESHK
           SCRXXX  THREE LETTER ID USED TO FORM MEMBER           PREJET,
                   NAME FLI(FLIXXX) AND SAESHK(XXXNNN) (A)       SAESHK
                   (DEFAULT=3HXXX)
           IOUT    CODE FOR NOISE DATA OUTPUT (I)                SAESHK
                   (DEFAULT=3)
                    0 - NO PRINT BUT GENERATE SAESHK(XXXNNN).
                   -1 - OUTPUT PRINT IN DB BUT NO TABLE.
                   -2 - OUTPUT PRINT IN DIMENSIONLESS FORM
                        BUT NO TABLE.
```

```
        -3 - OUTPUT PRINT IN DIMENSIONLESS FORM
             AND DB BUT NO TABLE.
         1 - OUTPUT PRINT IN DB AND GENERATE
             SAESHK(XXXNNN).
         2 - OUTPUT PRINT IN DIMENSIONLESS FORM
             AND GENERATE SAESHK(XXXNNN).
         3 - OUTPUT PRINT IN DIMENSIONLESS FORM
             AND DB AND GENERATE SAESHK(XXXNNN).
  IPRINT   OUTPUT PRINT OPTION FOR SAESHK MODULE      SAESHK
           (I) (DEFAULT=3)
           0 - NO PRINT
           1 - INPUT PRINT ONLY
           2 - OUTPUT PRINT ONLY
           3 - BOTH INPUT AND OUTPUT PRINT

 MEMBERS                                              MODULES

  FLI    (FLIXXX) - MULTI-RECORD MEMBER OF SOURCE     PREJET
                    TIME DATA.
  ENG    (PRIM  ) - TYPE 1 DATA TABLE OF THE NOZZLE   PREJET
                    FLOW STATE DATA.
  SFIELD(FREQ   ) - ONE RECORD MEMBER OF THE VALUES   SAESHK
                    OF FREQUENCY IN HERTZ.
  SFIELD(THETA  ) - ONE RECORD MEMBER OF THE VALUES   SAESHK
                    OF POLAR DIRECTIVITY ANGLE  IN
                    DEGREES.
  SFIELD(PHI    ) - ONE RECORD MEMBER OF THE VALUES   SAESHK
                    OF AZIMUTHAL DIRECTIVITY ANGLE
                    IN DEGREES.

OUTPUT                                                MODULES
 USER PARAMETERS

  RS       RADIAL DISTANCE FROM NOZZLE EXIT TO        SAESHK
           OBSERVER (M),(FT) (RS)
  SAEERR   ERROR PARAMETER (I)                        SAESHK
           0 - NO ERRORS ENCOUNTERED.
           1 - MEMBER MANAGER ERROR OCCURRED ON OPEN.
           2 - INSUFFICIENT LDS AVAILABLE.
           3 - MEMBER MANAGER ERROR OCCURRED ON READ.
           4 - AT LEAST ONE USER PARAMETER HAS AN
               INVALID VALUE.
           5 - UNABLE TO BUILD TABLE SAESHK(XXXNNN).
           6 - AN INVALID VALUE WAS CALCULATED
               DURING EXECUTION.
```

PROCEDURE PROCLIB (TSAESHK)

MEMBERS

			MODULES
SAESHK(XXX001)	-	NNN TYPE 1 DATA TABLES OF THE	SAESHK
...		MEAN-SQUARE ACOUSTIC PRESSURE	
SAESHK(XXXNNN)		FOR NNN SOURCE TIMES.	

LOCAL

USER PARAMETERS

		MODULES
NTIMES	NUMBER OF SOURCE TIME VALUES (I)	IF,PREJET
I	LOOP INDEX (I)	IF,PARAM
TIMES	MULTI-ELEMENT PARAMETER CONTAINING THE SOURCE TIMES (RS)	PREJET, PARAM
AT1	MULTI-ELEMENT PARAMETER CONTAINING THE JET AREA FOR EACH TIME (RS)	PARAM, PREJET
MTA	MULTI-ELEMENT PARAMETER CONTAINING THE AIRCRAFT MACH NUMBER FOR EACH TIME (RS)	PARAM, PREJET
DET1	MULTI-ELEMENT PARAMETER CONTAINING THE JET EQUIVALENT DIAMETER FOR EACH TIME (RS)	PREJET
DHT1	MULTI-ELEMENT PARAMETER CONTAINING THE JET HYDRAULIC DIAMETER FOR EACH TIME (RS)	PREJET
MT1	MULTI-ELEMENT PARAMETER CONTAINING THE JET MACH NUMBER FOR EACH TIME (RS)	PARAM, PREJET
TT1	MULTI-ELEMENT PARAMETER CONTAINING THE JET TOTAL TEMPERATURE FOR EACH TIME (RS)	PARAM, PREJET
VT1	MULTI-ELEMENT PARAMETER CONTAINING THE JET VELOCITY FOR EACH TIME (RS)	PARAM, PREJET
RSHT1	MULTI-ELEMENT PARAMETER CONTAINING THE RATIO OF SPECIFIC HEATS FOR EACH TIME (RS)	PREJET
RHOT1	MULTI-ELEMENT PARAMETER CONTAINING THE JET DENSITY FOR EACH TIME (RS)	PREJET
CTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT SPEED OF SOUND FOR EACH TIME (M/S),(FT/S) (RS)	PARAM, PREJET
RHOTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DENSITY FOR EACH TIME (KG/M**3), (SLUG/FT**3) (RS)	PARAM, PREJET
AJ	VALUE OF JET AREA FOR TIME I (RS)	PARAM, SAESHK
TJ	VALUE OF JET TEMPERATURE FOR TIME I (RS)	PARAM, SAESHK
VJ	VALUE OF JET VELOCITY FOR TIME I (RS)	PARAM, SAESHK
MJ	VALUE OF JET MACH NUMBER FOR TIME I (RS)	PARAM, SAESHK
RHOA	VALUE OF AMBIENT DENSITY FOR TIME I (KG/M**3),(SLUG/FT**3) (RS)	PARAM, SAESHK
CA	VALUE OF AMBIENT SPEED OF SOUND FOR TIME I (M/S),(FT/S) (RS)	PARAM, SAESHK

PROCEDURE PROCLIB (TSAESHK)

MA	VALUE OF THE AIRCRAFT MACH NUMBER FOR TIME I (RS)	PARAM, SAESHK
SCRNNN	INDEX FOR SOURCE MEMBER NAME = I (I)	PARAM, SAESHK
STIME	VALUE OF SOURCE TIME FOR TIME I (RS)	PARAM, SAESHK

MEMBERS - NONE

FUNCTIONS

1. LOAD SFIELD DATA UNIT IF NOT ASSIGNED.
2. EXECUTE PREJET MODULE TO GET JET NOISE PARAMETERS.
3. INITIALIZE SOURCE TIME LOOP.
4. EXECUTE SAESHK MODULE FOR EACH SOURCE TIME ON FLI(FLIXXX) UNIT MEMBER.

MODULES CALLED - PREJET SAESHK

PROCEDURES CALLED - XPREJET XSAESHK

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (TSGLJET)

```
****PROCEDURE TSGLJET

        PURPOSE - TO COMPUTE THE JET EXHAUST NOISE USING MODULE SGLJET
                  FOR A TIME DEPENDENT PROBLEM WITH INPUT DATA ON AN
                  ENGINE STATE TABLE.

        AUTHOR  - DSW(L03/00/00)
                  DSW(L03/00/05)
                  MMF(L03/00/09)

        INPUT
         USER PARAMETERS
                                                                 MODULES
          AE      ENGINE REFERENCE AREA (M**2),(FT**2)           PREJET,
                  (RS) (DEFAULT=PI/4. M**2)                      SGLJET
          AP      PRIMARY NOZZLE PLUG AREA, RE AE (RS)           PREJET
                  (DEFAULT=0.)
          IRATIO  SPECIFIC HEATS OPTION (A)
                  8HCONSTANT - CONSTANT RATIO OF SPECIFIC
                               HEATS (DEFAULT)
                  8HVARIABLE - VARIABLE RATIO OF SPECIFIC
                               HEATS
          ITYPE   TYPE OF INTERPOLATION TO BE USED ON THE        PREJET
                  ENGINE STATE TABLES (I) (DEFAULT=1)
                  1 - LINEAR INTERPOLATION
                  2 - CUBIC SPLINE INTERPOLATION
          IUNITS  INPUT UNIT OPTION (A) (DEFAULT=2HSI)           PREJET,
                  2HSI      - SI UNITS                           SGLJET
                  7HENGLISH - ENGLISH UNITS
          PREPRT  OUTPUT PRINT OPTION FOR PREJET MODULE          PREJET
                  (I) (DEFAULT=3)
                  0 - NO PRINT
                  1 - INPUT PRINT ONLY
                  2 - OUTPUT PRINT ONLY
                  3 - BOTH INPUT AND OUTPUT PRINT
          RS      RADIAL DISTANCE FROM NOZZLE EXIT TO            SGLJET
                  OBSERVER (M),(FT) (RS) (DEFAULT=SQRT(AE))
          DELTA   ANGLE BETWEEN FLIGHT VECTOR AND ENGINE         SGLJET
                  INLET AXIS (DEGREES) (RS) (DEFAULT=0.)
          NENG    NUMBER OF ENGINES (I) (DEFAULT=1)              SGLJET
          SCRXXX  THREE LETTER ID USED TO FORM MEMBER            PREJET,
                  NAME FLI(FLIXXX) AND SGLJET(XXXNNN) (A)        SGLJET
                  (DEFAULT=3HXXX)
          IOUT    CODE FOR NOISE DATA OUTPUT (I)                 SGLJET
                  (DEFAULT=3)
                   0 - NO PRINT BUT GENERATE SGLJET(XXXNNN).
                  -1 - OUTPUT PRINT IN DB BUT NO TABLE.
                  -2 - OUTPUT PRINT IN DIMENSIONLESS FORM
                       BUT NO TABLE.
```

```
        -3 - OUTPUT PRINT IN DIMENSIONLESS FORM
             AND DB BUT NO TABLE.
         1 - OUTPUT PRINT IN DB AND GENERATE
             SGLJET(XXXNNN).
         2 - OUTPUT PRINT IN DIMENSIONLESS FORM
             AND GENERATE SGLJET(XXXNNN).
         3 - OUTPUT PRINT IN DIMENSIONLESS FORM
             AND DB AND GENERATE SGLJET(XXXNNN).
 IPRINT  OUTPUT PRINT OPTION FOR SGLJET MODULE       SGLJET
         (I) (DEFAULT=3)
         0 - NO PRINT
         1 - INPUT PRINT ONLY
         2 - OUTPUT PRINT ONLY
         3 - BOTH INPUT AND OUTPUT PRINT

MEMBERS                                              MODULES

  FLI    (FLIXXX) - MULTI-RECORD MEMBER OF SOURCE    PREJET
                    TIME DATA.
  ENG    (PRIM  ) - TYPE 1 TABLE OF THE NOZZLE       PREJET
                    FLOW STATE DATA.
  SFIELD(FREQ  ) - ONE RECORD MEMBER OF THE VALUES   SGLJET
                    OF FREQUENCY IN HERTZ.
  SFIELD(THETA ) - ONE RECORD MEMBER OF THE VALUES   SGLJET
                    OF POLAR DIRECTIVITY ANGLE IN
                    DEGREES.
  SFIELD(PHI   ) - ONE RECORD MEMBER OF THE VALUES   SGLJET
                    OF AZIMUTHAL DIRECTIVITY ANGLE
                    IN DEGREES.
  SAE    (OM    ) - TYPE 1 DATA TABLE OF THE JET     SGLJET
                    VARIABLE DENSITY EXPONENT.
  SAE    (PDF   ) - TYPE 1 DATA TABLE OF THE POWER   SGLJET
                    DEVIATION FACTOR.
  SAE    (NDF   ) - TYPE 1 DATA TABLE OF THE         SGLJET
                    NORMALIZED DIRECTIVITY FUNCTION.
  SAE    (MTH   ) - TYPE 1 DATA TABLE OF THE         SGLJET
                    FORWARD VELOCITY INDEX.
  SAE    (SCJ   ) - TYPE 1 DATA TABLE OF THE         SGLJET
                    SPECTRAL DISTRIBUTION FACTOR.
  SAE    (SCF   ) - TYPE 1 DATA TABLE OF THE         SGLJET
                    STROUHAL NUMBER CORRECTION
                    FACTOR.

OUTPUT
 USER PARAMETERS                                     MODULES

  RS     RADIAL DISTANCE FROM NOZZLE EXIT TO         SGLJET
         OBSERVER (M),(FT) (RS)
```

PROCEDURE PROCLIB (TSGLJET)

MEMBERS		MODULES
SGLJET(XXX001) - ... SGLJET(XXXNNN)	NNN TYPE 1 DATA TABLES OF THE MEAN-SQUARE ACOUSTIC PRESSURE FOR NNN SOURCE TIMES.	SGLJET

LOCAL
USER PARAMETERS

		MODULES
NTIMES	NUMBER OF SOURCE TIME VALUES (I)	IF,PREJET
I	LOOP INDEX (I)	IF,PARAM
TIMES	MULTI-ELEMENT PARAMETER CONTAINING THE SOURCE TIMES (RS)	PREJET, PARAM
AT1	MULTI-ELEMENT PARAMETER CONTAINING THE JET AREA FOR EACH TIME (RS)	PARAM, PREJET
MTA	MULTI-ELEMENT PARAMETER CONTAINING THE AIRCRAFT MACH NUMBER FOR EACH TIME (RS)	PARAM, PREJET
DET1	MULTI-ELEMENT PARAMETER CONTAINING THE JET EQUIVALENT DIAMETER FOR EACH TIME (RS)	PREJET
DHT1	MULTI-ELEMENT PARAMETER CONTAINING THE JET HYDRAULIC DIAMETER FOR EACH TIME (RS)	PREJET
MT1	MULTI-ELEMENT PARAMETER CONTAINING THE JET MACH NUMBER FOR EACH TIME (RS)	PREJET
TT1	MULTI-ELEMENT PARAMETER CONTAINING THE JET TOTAL TEMPERATURE FOR EACH TIME (RS)	PARAM, PREJET
VT1	MULTI-ELEMENT PARAMETER CONTAINING THE JET VELOCITY FOR EACH TIME (RS)	PARAM, PREJET
RSHT1	MULTI-ELEMENT PARAMETER CONTAINING THE RATIO OF SPECIFIC HEATS FOR EACH TIME (RS)	PREJET
RHOT1	MULTI-ELEMENT PARAMETER CONTAINING THE JET DENSITY FOR EACH TIME (RS)	PARAM, PREJET
CTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT SPEED OF SOUND FOR EACH TIME (M/S),(FT/S) (RS)	PARAM, PREJET
RHOTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DENSITY FOR EACH TIME (KG/M**3), (SLUG/FT**3) (RS)	PARAM, PREJET
AJ	VALUE OF JET AREA FOR TIME I (RS)	PARAM, SGLJET
RHOJ	VALUE OF JET DENSITY FOR TIME I (RS)	PARAM, SGLJET
TJ	VALUE OF JET TEMPERATURE FOR TIME I (RS)	PARAM, SGLJET
VJ	VALUE OF JET VELOCITY FOR TIME I (RS)	PARAM, SGLJET
RHOA	VALUE OF AMBIENT DENSITY FOR TIME I (KG/M**3),(SLUG/FT**3) (RS)	PARAM, SGLJET
CA	VALUE OF AMBIENT SPEED OF SOUND FOR TIME I (M/S),(FT/S) (RS)	PARAM, SGLJET

PROCEDURE PROCLIB (TSGLJET)

STIME	VALUE OF SOURCE TIME FOR TIME I (RS)	PARAM, SGLJET
MA	VALUE OF THE AIRCRAFT MACH NUMBER FOR TIME I (RS)	PARAM, SGLJET
SCRNNN	INDEX FOR SOURCE MEMBER NAME = I (I)	PARAM, SGLJET

MEMBERS - NONE

FUNCTIONS

1. LOAD SAE AND SFIELD DATA UNITS IF NOT ASSIGNED.
2. EXECUTE PREJET MODULE TO GET JET NOISE PARAMETERS
3. INITIALIZE SOURCE TIME LOOP.
4. EXECUTE SGLJET MODULE FOR EACH SOURCE TIME ON FLI(FLIXXX) UNIT MEMBER.

MODULES CALLED - PREJET SGLJET

PROCEDURES CALLED - XPREJET XSGLJET

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (TSMBTUR)

****PROCEDURE TSMBTUR

PURPOSE - TO COMPUTE THE TURBINE NOISE USING MODULE SMBTUR FOR A TIME DEPENDENT PROBLEM WITH INPUT DATA ON ENGINE STATE TABLES.

AUTHOR - DSW(L03/00/00)
WKB(L03/00/01)
DSW(L03/00/05)
MMF(L03/00/09)

INPUT

USER PARAMETERS

		MODULES
AE	ENGINE REFERENCE AREA (M**2),(FT**2) (RS) (DEFAULT=PI/4. M**2)	SMBTUR
NENG	NUMBER OF ENGINES (I) (DEFAULT=1)	SMBTUR
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS) (DEFAULT=SQRT(AE))	SMBTUR
AREA	TURBINE INLET CROSS-SECTIONAL AREA, RE AE (RS) (DEFAULT=1.)	SMBTUR
C	ROTOR BLADE MEAN AXIAL CHORD OF THE LAST STAGE, RE SQRT(AE) (RS) (DEFAULT=1.)	SMBTUR
NS	NUMBER OF STAGES (I) (DEFAULT=1)	SMBTUR
SCRXXX	THREE LETTER ID USED TO FORM MEMBER NAME FLI(FLIXXX) AND SMBTUR(XXXNNN) (A) (DEFAULT=3HXXX)	SMBTUR, PRETUR
IOUT	OUTPUT CODE FOR TABLE AND PRINTED OUTPUT (I) (DEFAULT=3) 0 - NO PRINT BUT GENERATE SMBTUR(XXXNNN). -1 - OUTPUT PRINT IN DB BUT NO TABLE. -2 - OUTPUT PRINT IN DIMENSIONLESS FORM BUT NO TABLE. -3 - OUTPUT PRINT IN DIMENSIONLESS FORM AND DB BUT NO TABLE. 1 - OUTPUT PRINT IN DB AND GENERATE SMBTUR(XXXNNN). 2 - OUTPUT PRINT IN DIMENSIONLESS FORM AND GENERATE SMBTUR(XXXNNN). 3 - OUTPUT PRINT IN DIMENSIONLESS FORM AND DB AND GENERATE SMBTUR(XXXNNN).	SMBTUR
IPRINT	OUTPUT PRINT CODE FOR SMBTUR MODULE (I) (DEFAULT=3) 0 - NO PRINT 1 - INPUT PRINT ONLY 2 - OUTPUT PRINT ONLY 3 - BOTH INPUT AND OUTPUT PRINT	SMBTUR

PROCEDURE PROCLIB (TSMBTUR)

```
  IUNITS  INPUT UNITS OPTION (A) (DEFAULT=2HSI)        SMBTUR,
          2HSI      - SI UNITS                         PRETUR
          7HENGLISH - ENGLISH UNITS
  ITYPE   TYPE OF INTERPOLATION TO BE USED ON THE      PRETUR
          ENGINE STATE TABLES (I) (DEFAULT=1)
          1 - LINEAR INTERPOLATION
          2 - CUBIC SPLINE INTERPOLATION
  IRATIO  SPECIFIC HEATS OPTION (A)                    PRETUR
          8HCONSTANT - CONSTANT RATIO OF SPECIFIC
                       HEATS (DEFAULT)
          8HVARIABLE - VARIABLE RATIO OF SPECIFIC
                       HEATS
  PREPRT  OUTPUT PRINT CODE FOR PRETUR MODULE          PRETUR
          (I) (DEFAULT=3)
          0 - NO PRINT
          1 - INPUT PRINT ONLY
          2 - OUTPUT PRINT ONLY
          3 - BOTH INPUT AND OUTPUT PRINT

 MEMBERS                                              MODULES

  FLI (FLIXXX  ) - MULTI-RECORD MEMBER OF SOURCE       PRETUR
                   TIME DATA.
  ENG (TURBINE1) - TYPE 1 DATA TABLE OF THE            PRETUR
                   TURBINE ENTRANCE FLOW STATE.
  ENG (TURBINE2) - TYPE 1 DATA TABLE OF THE            PRETUR
                   TURBINE EXIT FLOW STATE.
  SFIELD(FREQ  ) - ONE RECORD MEMBER OF THE VALUES     SMBTUR
                   OF FREQUENCY IN HERTZ.
  SFIELD(THETA ) - ONE RECORD MEMBER OF THE VALUES     SMBTUR
                   OF POLAR DIRECTIVITY ANGLE IN
                   DEGREES.
  SFIELD(PHI   ) - ONE RECORD MEMBER OF THE VALUES     SMBTUR
                   OF AZIMUTHAL DIRECTIVITY ANGLE
                   IN DEGREES.

OUTPUT
 USER PARAMETERS                                      MODULES

  RS      DISTANCE FROM SOURCE TO OBSERVER (M),(FT)    SMBTUR
          (RS)

 MEMBERS                                              MODULES

  SMBTUR(XXX001) - NNN TYPE 1 DATA TABLES OF THE       SMBTUR
      ...          MEAN-SQUARE ACOUSTIC PRESSURE
  SMBTUR(XXXNNN)   FOR NNN SOURCE TIMES.
```

PROCEDURE PROCLIB (TSMBTUR)

LOCAL

USER PARAMETERS

		MODULES
NTIMES	NUMBER OF SOURCE TIME VALUES (I)	IF,PRETUR
I	LOOP INDEX (I)	IF,PARAM
TIMES	MULTI-ELEMENT PARAMETER CONTAINING THE SOURCE TIMES (RS)	PARAM, PRETUR
MTA	MULTI-ELEMENT PARAMETER CONTAINING THE AIRCRAFT MACH NUMBER FOR EACH TIME (RS)	PARAM, PRETUR
MT1	MULTI-ELEMENT PARAMETER CONTAINING THE CORE MASS FLOW RATE FOR EACH TIME (RS)	PARAM, PRETUR
NT1	MULTI-ELEMENT PARAMETER CONTAINING THE ROTATIONAL SPEED FOR EACH TIME (RS)	PARAM, PRETUR
TT1	MULTI-ELEMENT PARAMETER CONTAINING THE ENTRANCE TOTAL TEMPERATURE FOR EACH TIME (RS)	PRETUR
TT2	MULTI-ELEMENT PARAMETER CONTAINING THE EXIT STATIC TEMPERATURE FOR EACH TIME (RS)	PARAM, PRETUR
CTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT SPEED OF SOUND FOR EACH TIME (M/S),(FT/S) (RS)	PARAM, PRETUR
RHOTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DENSITY FOR EACH TIME (KG/M**3), (SLUG/FT**3) (RS)	PARAM, PRETUR
FTA	MULTI-ELEMENT PARAMETER CONTAINING THE FUEL-TO-AIR RATIO FOR EACH TIME (RS)	PARAM, PRETUR
HTA	MULTI-ELEMENT PARAMETER CONTAINING THE ABSOLUTE HUMIDITY FOR EACH TIME (RS)	PARAM, PRETUR
MA	AIRCRAFT MACH NUMBER AT TIME I (RS)	PARAM, SMBTUR
N	ROTATIONAL SPEED AT TIME I (RS)	PARAM, SMBTUR
MDOT	CORE MASS FLOW RATE AT TIME T (RS)	PARAM, SMBTUR
TSJ	EXIT STATIC TEMPERATURE AT TIME I (RS)	PARAM, SMBTUR
CA	AMBIENT SPEED OF SOUND AT TIME I (M/S), (FT/S) (RS)	PARAM, SMBTUR
RHOA	AMBIENT DENSITY AT TIME I (KG/M**3), (SLUG/FT**3) (RS)	PARAM, SMBTUR
SCRNNN	INDEX FOR SOURCE MEMBER NAME = I (I)	PARAM, SMBTUR
STIME	SOURCE TIME VALUE AT TIME I (RS)	PARAM, SMBTUR
FA	FUEL-TO-AIR RATIO AT TIME I (RS)	PARAM, SMBTUR
HA	ABSOLUTE HUMIDITY AT TIME I (RS)	PARAM, SMBTUR

PROCEDURE PROCLIB (TSMBTUR)

MEMBERS - NONE

FUNCTIONS

1. LOAD SFIELD DATA UNIT IF NOT ASSIGNED.
2. EXECUTE PRETUR MODULE TO GET TURBINE NOISE PARAMETERS.
3. INITIALIZE SOURCE TIME LOOP.
4. EXECUTE SMBTUR MODULE FOR EACH SOURCE TIME ON FLI(FLIXXX) UNIT MEMBER.

MODULES CALLED - PRETUR SMBTUR

PROCEDURES CALLED - XPRETUR XSMBTUR

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (TSTNJET)

****PROCEDURE TSTNJET

PURPOSE - TO COMPUTE JET EXHAUST NOISE USING MODULE STNJET FOR A TIME DEPENDENT PROBLEM WITH INPUT DATA ON ENGINE STATE TABLES.

AUTHOR - DSW(L03/00/05)
MMF(L03/00/09)

INPUT
USER PARAMETERS

		MODULES
AE	ENGINE REFERENCE AREA (M**2),(FT**2) (RS) (DEFAULT=PI/4. M**2)	PREJET, STNJET
AP	PRIMARY NOZZLE PLUG AREA, RE AE (RS) (DEFAULT=0.)	PREJET
IRATIO	SPECIFIC HEATS OPTION (A) 8HCONSTANT - CONSTANT RATIO OF SPECIFIC HEATS (DEFAULT) 8HVARIABLE - VARIABLE RATIO OF SPECIFIC HEATS	PREJET
ITYPE	TYPE OF INTERPOLATION TO BE USED ON THE ENGINE STATE TABLES (I) (DEFAULT=1) 1 - LINEAR INTERPOLATION 2 - CUBIC SPLINE INTERPOLATION	PREJET
IUNITS	INPUT UNIT OPTION (A) (DEFAULT=2HSI) 2HSI - SI UNITS 7HENGLISH - ENGLISH UNITS	PREJET, STNJET
PREPRT	OUTPUT PRINT OPTION FOR PREJET MODULE (I) (DEFAULT=3) 0 - NO PRINT 1 - INPUT PRINT ONLY 2 - OUTPUT PRINT ONLY 3 - BOTH INPUT AND OUTPUT PRINT	PREJET
RS	DISTANCE FROM NOZZLE EXIT TO OBSERVER (M),(FT) (RS) (DEFAULT=SQRT(AE))	STNJET
DELTA	ANGLE BETWEEN FLIGHT VECTOR AND ENGINE INLET AXIS (DEGREES) (RS) (DEFAULT=0.)	STNJET
NENG	NUMBER OF ENGINES (I) (DEFAULT=1)	STNJET
SCRXXX	THREE LETTER ID USED TO FORM MEMBER NAME FLI(FLIXXX) AND STNJET(XXXNNN) (A) (DEFAULT=3HXXX)	
IOUT	CODE FOR NOISE DATA OUTPUT (I) (DEFAULT=3) 0 - NO PRINT BUT GENERATE STNJET(XXXNNN). -1 - OUTPUT PRINT IN DB BUT NO TABLE. -2 - OUTPUT PRINT IN DIMENSIONAL FORM BUT NO TABLE.	STNJET

	-3 - OUTPUT PRINT IN DIMENSIONAL FORM AND DB BUT NO TABLE.	
	1 - OUTPUT PRINT IN DB AND GENERATE STNJET(XXXNNN).	
	2 - OUTPUT PRINT IN DIMENSIONLESS FORM AND GENERATE STNJET(XXXNNN).	
	3 - OUTPUT PRINT IN DIMENSIONLESS FORM AND DB AND GENERATE STNJET(XXXNNN).	
IPRINT	OUTPUT PRINT OPTION FOR STNJET MODULE (I) (DEFAULT=3) 0 - NO PRINT 1 - INPUT PRINT ONLY 2 - OUTPUT PRINT ONLY 3 - BOTH INPUT AND OUTPUT PRINT	STNJET
PLUG	NOZZLE PLUG OPTION (L) (DEFAULT=.FALSE.) .TRUE. - CONICAL PLUG NOZZLE .FALSE. - NO PLUG	STNJET
CIRCLE	NOZZLE TYPE OPTION (L) (DEFAULT=.FALSE.) .TRUE. - CIRCULAR NOZZLE .FALSE. - COAXIAL NOZZLE	STNJET
SUPER	SHOCK NOISE OPTION (L) (DEFAULT=.FALSE.) .TRUE. - COMPUTE PRIMARY STREAM SHOCK NOISE .FALSE. - DO NOT COMPUTE	STNJET

MEMBERS

		MODULES
FL1 (FLIXXX)	- MULTI-RECORD MEMBER OF SOURCE TIME DATA.	PREJET
ENG (PRIM)	- TYPE 1 DATA TABLE OF THE PRIMARY NOZZLE FLOW STATE.	PREJET
ENG (SEC)	- TYPE 1 DATA TABLE OF THE SECONDARY NOZZLE FLOW STATE.	PREJET
SFIELD(FREQ)	- ONE RECORD MEMBER OF THE VALUES OF FREQUENCY IN HERTZ.	STNJET
SFIELD(THETA)	- ONE RECORD MEMBER OF THE VALUES OF POLAR DIRECTIVITY ANGLE IN DEGREES.	STNJET
SFIELD(PHI)	- ONE RECORD MEMBER OF THE VALUES OF AZIMUTHAL DIRECTIVITY ANGLE IN DEGREES.	STNJET
STNTBL(JDF)	- TYPE 1 DATA TABLE OF THE JET MIXING NOISE SPECTRAL DISTRIBUTION FACTOR.	STNJET
STNTBL(SDF)	- TYPE 1 DATA TABLE OF THE SHOCK NOISE SPECTRAL DISTRIBUTION FACTOR.	STNJET
STNTBL(FSP)	- TYPE 1 DATA TABLE OF THE FREQUENCY SHIFT PARAMETER.	STNJET

OUTPUT

USER PARAMETERS

		MODULES
RS	DISTANCE FROM SOURCE TO OBSERVER (M),(FT) (RS)	STNJET

MEMBERS

		MODULES
STNJET(XXX001) ... STNJET(XXXNNN)	- NNN TYPE 1 DATA TABLES OF THE MEAN-SQUARE ACOUSTIC PRESSURE FOR NNN SOURCE TIMES.	STNJET

LOCAL

USER PARAMETERS

		MODULES
NTIMES	NUMBER OF SOURCE TIME VALUES (I)	IF,PREJET
I	LOOP INDEX (I)	IF,PARAM
TIMES	MULTI-ELEMENT PARAMETER CONTAINING THE SOURCE TIMES (RS)	PARAM, PREJET
AT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET AREA FOR EACH TIME (RS)	PARAM, PREJET
MTA	MULTI-ELEMENT PARAMETER CONTAINING THE AIRCRAFT MACH NUMBER FOR EACH TIME (RS)	PARAM, PREJET
MT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET MACH NUMBER FOR EACH TIME (RS)	PARAM, PREJET
TT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET TOTAL TEMPERATURE FOR EACH TIME (RS)	PARAM, PREJET
VT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET VELOCITY FOR EACH TIME (RS)	PARAM, PREJET
RSHT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET RATIO OF SPECIFIC HEATS FOR EACH TIME (RS)	PREJET
RHOT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET DENSITY FOR EACH TIME (RS)	PARAM, PREJET
DET1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET EQUIVALENT DIAMETER FOR EACH TIME (RS)	PARAM, PREJET
DHT1	MULTI-ELEMENT PARAMETER CONTAINING THE PRIMARY JET HYDRAULIC DIAMETER FOR EACH TIME (RS)	PARAM, PREJET
CTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT SPEED OF SOUND FOR EACH TIME (M/S),(FT/S) (RS)	PARAM, PREJET
RHOTA	MULTI-ELEMENT PARAMETER CONTAINING THE AMBIENT DENSITY FOR EACH TIME (KG/M**3), (SLUG/FT**3) (RS)	PARAM, PREJET
AT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET AREA FOR EACH TIME (RS)	PARAM, PREJET

MT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET MACH NUMBER FOR EACH TIME (RS)	PARAM, PREJET
TT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET TOTAL TEMPERATURE FOR EACH TIME (RS)	PARAM, PREJET
VT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET VELOCITY FOR EACH TIME (RS)	PARAM, PREJET
RHOT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET DENSITY FOR EACH TIME (RS)	PARAM, PREJET
RSHT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET RATIO OF SPECIFIC HEATS FOR EACH TIME (RS)	PREJET
DET2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET EQUIVALENT DIAMETER FOR EACH TIME (RS)	PREJET
DHT2	MULTI-ELEMENT PARAMETER CONTAINING THE SECONDARY JET HYDRAULIC DIAMETER FOR EACH TIME (RS)	PREJET
CA	VALUE OF AMBIENT SPEED OF SOUND FOR TIME I (M/S),(FT/S) (RS)	PARAM, STNJET
RHOA	VALUE OF AMBIENT DENSITY FOR TIME I (KG/M**3),(SLUG/FT**3) (RS)	PARAM, STNJET
MA	VALUE OF AIRCRAFT MACH NUMBER FOR TIME I (RS)	PARAM, STNJET
A1	VALUE OF PRIMARY JET AREA FOR TIME I (RS)	PARAM, STNJET
DE1	VALUE OF PRIMARY JET EQUIVALENT DIAMETER FOR TIME I (RS)	PARAM, STNJET
DH1	VALUE OF PRIMARY JET HYDRAULIC DIAMETER FOR TIME I (RS)	PARAM, STNJET
V1	VALUE OF PRIMARY JET VELOCITY FOR TIME I (RS)	PARAM, STNJET
RHO1	VALUE OF PRIMARY JET DENSITY FOR TIME I (RS)	PARAM, STNJET
T1	VALUE OF PRIMARY JET TOTAL TEMPERATURE FOR TIME I (RS)	PARAM, STNJET
M1	VALUE OF PRIMARY JET MACH NUMBER FOR TIME I (RS)	PARAM, STNJET
A2	VALUE OF SECONDARY JET AREA FOR TIME I (RS)	PARAM, STNJET
V2	VALUE OF SECONDARY JET VELOCITY FOR TIME I (RS)	PARAM, STNJET
RHO2	VALUE OF SECONDARY JET DENSITY FOR TIME I (RS)	PARAM, STNJET
T2	VALUE OF SECONDARY JET TOTAL TEMPERATURE FOR TIME I (RS)	PARAM, STNJET
M2	VALUE OF SECONDARY JET MACH NUMBER FOR TIME I (RS)	PARAM, STNJET

PROCEDURE PROCLIB (TSTNJET)

SCRNNN	INDEX FOR SOURCE MEMBER NAME = I (I)	PARAM, STNJET
STIME	VALUE OF SOURCE TIME FOR TIME I (RS)	PARAM, STNJET

MEMBERS - NONE

FUNCTIONS

1. LOAD STNJET AND SFIELD DATA UNITS IF NOT ASSIGNED.
2. EXECUTE PREJET MODULE TO GET JET NOISE PARAMETERS.
3. INITIALIZE SOURCE TIME LOOP.
4. EXECUTE STNJET MODULE FOR EACH SOURCE TIME ON FLI(FLIXXX) UNIT MEMBER.

MODULES CALLED - PREJET STNJET

PROCEDURES CALLED - XPREJET XSTNJET

CALLING PROCEDURES - NONE

PROCEDURE PROCLIB (XABS)

****PROCEDURE XABS

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE ABS MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
ATM (TMOD)	ATMOS (INPUT)

OUTPUT
USER PARAMETERS - SEE ABS MODULE

MEMBERS

OLD NAME	NEW NAME
ATM (AAC)	ATMOS (OUTPUT)

LOCAL - NONE

MODULES CALLED - ABS

PROCEDURES CALLED - NONE

CALLING PROCEDURES - ATMDAT

****PROCEDURE XATM

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE ATM MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT
H1	XP1
P1	XP2
DELH	XP3
NHO	XP4

MEMBERS

OLD NAME	NEW NAME
ATM (IN)	ATMOS (INPUT)

OUTPUT
USER PARAMETERS - SEE ATM MODULE

MEMBERS

OLD NAME	NEW NAME
ATM (TMOD)	ATMOS (OUTPUT)

LOCAL
USER PARAMETERS - NONE

MEMBERS

OLD NAME	NEW NAME
SCRATCH(TAB1)	TEMP (TABLE)

MODULES CALLED - ATM

PROCEDURES CALLED - NONE

CALLING PROCEDURES - ATMDAT

PROCEDURE PROCLIB (XCNLJET)

****PROCEDURE XCNLJET

```
PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT
          MEMBERS AND PARAMETERS OF THE CNLJET MODULE.  THIS
          ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME
          DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR  - DSW(L03/00/00)
          DSW(L03/00/05)

INPUT
  USER PARAMETERS

    OLD NAME              NEW NAME

    IPRINT                PRT

  MEMBERS

    OLD NAME              NEW NAME

    SFIELD (FREQ  )       ARRAY (FREQS )
    SFIELD (THETA )       ARRAY (THETAS)
    SFIELD (PHI   )       ARRAY (PHIS  )
    JWRCOAN(OM    )       CNLDAT(OMS    )
    JWRCOAN(PDF   )       CNLDAT(PDFS   )
    JWRCOAN(S1    )       CNLDAT(S1S    )
    JWRCOAN(S2    )       CNLDAT(S2S    )
    JWRCOAN(NSF   )       CNLDAT(NSFS   )
    JWRCOAN(ALPHA )       CNLDAT(ALPHAS)
    JWRCOAN(CBF   )       CNLDAT(CBFS   )
    JWRCOAN(DIR   )       CNLDAT(DIRS   )
    JWRCOAN(MTH   )       CNLDAT(MTHS   )

OUTPUT
  USER PARAMETERS - SEE CNLJET MODULE

  MEMBERS - NO NAME OVERRIDE REQUIRED FOR CNLJET(XXXNNN)

LOCAL - NONE

MODULES CALLED - CNLJET

PROCEDURES CALLED - NONE

CALLING PROCEDURES - DCNLJET TCNLJET
```

PROCEDURE PROCLIB (XEFF)

****PROCEDURE XEFF

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE EFF MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
OBSERV(COORD)	OBS (LOC)
LEV (PNLT)	SOUND (LEVELS)

OUTPUT
USER PARAMETERS - SEE EFF MODULE

MEMBERS

OLD NAME	NEW NAME
EFF (EPNL)	NOISE (EFFECT)

LOCAL - NONE

MODULES CALLED - EFF

PROCEDURES CALLED - NONE

CALLING PROCEDURES - PROSGL

PROCEDURE PROCLIB (XFNKAFM)

***PROCEDURE XFNKAFM

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE FNKAFM MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN WITH THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
SFIELD(FREQ)	ARRAY (FREQS)
SFIELD(THETA)	ARRAY (THETAS)
SFIELD(PHI)	ARRAY (PHIS)

OUTPUT
USER PARAMETERS - SEE FNKAFM MODULE

MEMBERS - NO NAME OVERRIDE REQUIRED FOR FNKAFM(XXXNNN)

LOCAL - NONE

MODULES CALLED - FNKAFM

PROCEDURES CALLED - NONE

CALLING PROCEDURES - DFNKAFM TFNKAFM

PROCEDURE PROCLIB (XGECOR)

****PROCEDURE XGECOR

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE GECOR MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
SFIELD(FREQ)	ARRAY (FREQS)
SFIELD(THETA)	ARRAY (THETAS)
SFIELD(PHI)	ARRAY (PHIS)

OUTPUT
USER PARAMETERS - SEE GECOR MODULE

MEMBERS - NO NAME OVERRIDE REQUIRED FOR GECOR(XXXNNN)

LOCAL - NONE

MODULES CALLED - GECOR

PROCEDURES CALLED - NONE

CALLING PROCEDURES - TGECOR DGECOR

PROCEDURE PROCLIB (XGENSUP)

****PROCEDURE XGENSUP

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE GENSUP MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/05)

INPUT
USER PARAMETERS - SEE GENSUP MODULE

MEMBERS

OLD NAME	NEW NAME
NOISE (XXXNNN)	UNIT (XXXNNN)
SUPPRESS(FACTOR)	FACT (MULT)

OUTPUT
USER PARAMETERS - SEE GENSUP MODULE

MEMBERS

OLD NAME	NEW NAME
NOISES (XXXNNN)	UNITS (XXXNNN)

LOCAL - NONE

MODULES CALLED - GENSUP

PROCEDURES CALLED - NONE

CALLING PROCEDURES - SDLDIR

PROCEDURE PROCLIB (XGETUR)

****PROCEDURE XGETUR

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE GETUR MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
SFIELD(FREQ)	ARRAY (FREQS)
SFIELD(THETA)	ARRAY (THETAS)
SFIELD(PHI)	ARRAY (PHIS)

OUTPUT
USER PARAMETERS - SEE GETUR MODULE

MEMBERS - NO NAME OVERRIDE REQUIRED FOR GETUR(XXXNNN)

LOCAL - NONE

MODULES CALLED - GETUR

PROCEDURES CALLED - NONE

CALLING PROCEDURES - DGETUR TGETUR

PROCEDURE PROCLIB (XGRA)

```
****PROCEDURE XGRA

     PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE INPUT
               AND OUTPUT MEMBERS OF THE GRA MODULE.  THIS ALLOWS
               PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES
               AS THE ORIGINAL MODULE.

     AUTHOR  - DSW(L03/00/00)
               WKB(L03/00/01)

     INPUT
       USER PARAMETERS - SEE GRA MODULE

       MEMBERS - NONE

     OUTPUT
       USER PARAMETERS - SEE GRA MODULE

       MEMBERS

         OLD NAME            NEW NAME

         TAB   (GE     )     GRA   (TABLE )

     LOCAL - NONE

     MODULES CALLED - GRA

     PROCEDURES CALLED - NONE

     CALLING PROCEDURES - NONE
```

PROCEDURE PROCLIB (XHDNFAN)

****PROCEDURE XHDNFAN

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE HDNFAN MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
SFIELD(FREQ)	ARRAY (FREQS)
SFIELD(THETA)	ARRAY (THETAS)
SFIELD(PHI)	ARRAY (PHIS)

OUTPUT
USER PARAMETERS - SEE HDNFAN MODULE

MEMBERS

OLD NAME	NEW NAME
HDNFAN(XXXNNN)	HDNNAM(XXXNNN)

LOCAL - NONE

MODULES CALLED - HDNFAN

PROCEDURES CALLED - NONE

CALLING PROCEDURES - DHDNFAN THDNFAN

PROCEDURE PROCLIB (XLEV)

****PROCEDURE XLEV

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE LEV MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
OBSERV(COORD)	OBS (LOC)
SFIELD(FREQ)	FREQU (ARRAY)

OUTPUT
USER PARAMETERS - SEE LEV MODULE

MEMBERS

OLD NAME	NEW NAME
LEV (PNLT)	NOISE(LEVELS)

LOCAL - NONE

MODULES CALLED - LEV

PROCEDURES CALLED - NONE

CALLING PROCEDURES - PROSGL

c - 4

PROCEDURE PROCLIB (XPREAFM)

****PROCEDURE XPREAFM

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE PREAFM MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN WITH THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)
MMF(L03/00/09)

INPUT

USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
FLI (FLIXXX)	FLIGHT(FLIXXX)

OUTPUT

USER PARAMETERS - SEE PREAFM MODULE

MEMBERS - NONE

LOCAL - NONE

MODULES CALLED - PREAFM

PROCEDURES CALLED - NONE

CALLING PROCEDURES - TFNKAFM

PROCEDURE PROCLIB (XPRECOR ,

****PROCEDURE XPRECOR

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE PRECOR MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
FLI (FLIXXX)	FLIGHT(FLIXXX)
ENG (CORE1)	ENGINE(CORIN)
ENG (CORE2)	ENGINE(COROUT)

OUTPUT
USER PARAMETERS - SEE PRECOR MODULE

MEMBERS - NONE

LOCAL - NONE

MODULES CALLED - PRECOR

PROCEDURES CALLED - NONE

CALLING PROCEDURES - TGECOR

PROCEDURE PROCLIB (XPREFAN)

****PROCEDURE XPREFAN

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE PREFAN MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)
MMF(L03/00/09)

INPUT

USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
FLI (FLIXXX)	FLIGHT(FLIXXX)
ENG (FAN1)	ENGINE(FANIN)
ENG (FAN2)	ENGINE(FANOUT)

OUTPUT

USER PARAMETERS - SEE PREFAN MODULE

MEMBERS - NONE

LOCAL - NONE

MODULES CALLED - PREFAN

PROCEDURES CALLED - NONE

CALLING PROCEDURES - THDNFAN

PROCEDURE PROCLIB (XPREJET)

```
****PROCEDURE XPREJET

        PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT
                  MEMBERS AND PARAMETERS OF THE PREJET MODULE.  THIS
                  ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME
                  DEFAULT NAMES AS THE ORIGINAL MODULE.

        AUTHOR  - DSW(L03/00/00)
                  DSW(L03/00/05)

        INPUT
          USER PARAMETERS

            OLD NAME              NEW NAME

            IPRINT                PRT

          MEMBERS

            OLD NAME              NEW NAME

            FLI   (FLIXXX)        FLIGHT(FLIXXX)
            ENG   (PRIM  )        ENGINE(ONE    )
            ENG   (SEC   )        ENGINE(TWO    )

        OUTPUT
          USER PARAMETERS - SEE PREJET MODULE

          MEMBERS - NONE

        LOCAL - NONE

        MODULES CALLED - PREJET

        PROCEDURES CALLED - NONE

        CALLING PROCEDURES - TCNLJET TSAESHK TSGLJET TSTNJET
```

PROCEDURE PROCLIB (XPRETUR)

****PROC JRE XPRETUR

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE PRETUR MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)
MMF(L03/00/09)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME		NEW NAME
FLI	(FLIXXX)	FLIGHT(FLIXXX)
ENG	(TURBINE1)	ENGINE(TURIN)
ENG	(TURBINE2)	ENGINE(TUROUT)

OUTPUT
USER PARAMETERS - SEE PRETUR MODULE

MEMBERS - NONE

LOCAL - NONE

MODULES CALLED - PRETUR

PROCEDURES CALLED - NONE

CALLING PROCEDURES - TGETUR TSMBTUR

PROCEDURE PROCLIB (XPRO)

****PROCEDURE XPRO

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE PRO MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT

USER PARAMETERS

OLD NAME	NEW NAME
PROSUM	PRONAM
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
ATM (TMOD)	ATMOS (DATA)
ATM (AAC)	ATMOS (ABSORP)
GEO (GEOM)	OBS (VECTOR)
FLI (FLIXXX)	FLIGHT(FLIXXX)

OUTPUT

USER PARAMETERS - SEE PRO MODULE

MEMBERS

OLD NAME	NEW NAME
PRO (PRES)	NOISE (SPEC)

LOCAL - NONE

MODULES CALLED - PRO

PROCEDURES CALLED - NONE

CALLING PROCEDURES - PROSGL

PROCEDURE PROCLIB (XPROSGL)

****PROCEDURE XPROSGL

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF PROCEDURE PROSGL. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL PROCEDURE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)
MMF(L03/00/09)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
PROSUM	PRONAM
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
ATM (TMOD)	ATMOS (DATA)
ATM (AAC)	ATMOS (ABSORP)
GEO (GEOM)	OBS (VECTOR)
FLI (FLIXXX)	FLIGHT(FLIXXX)
FLI (PATH)	FLIGHT(TRAJ)
OBSERV(COORD)	OBSER (LOC)
SFIELD(FREQ)	FREQU (ARRAY)

OUTPUT
USER PARAMETERS - UNCHANGED

MEMBERS

OLD NAME	NEW NAME
PRO (PRES)	NOIS1 (SPEC)
LEV (PNLT)	NOIS2 (LEVELS)
EFF (EPNL)	NOIS3 (EFFECT)
SAELAT(EPNL)	NOIS4 (EFFECT)

LOCAL - NONE

MODULES CALLED - EFF LEV PRO SAELAT

PROCEDURES CALLED - PROSGL XEFF XLEV XPRO XSAELAT

CALLING PROCEDURES - PROMUL

PROCEDURE PROCLIB (XSAESHK)

****PROCEDURE XSAESHK

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE SAESHK MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
SFIELD(FREQ)	ARRAY (FREQS)
SFIELD(THETA)	ARRAY (THETAS)
SFIELD(PHI)	ARRAY (PHIS)

OUTPUT
USER PARAMETERS - SEE SAESHK MODULE

MEMBERS - NO NAME OVERRIDE REQUIRED FOR SAESHK(XXXNNN)

LOCAL - NONE

MODULES CALLED - SAESHK

PROCEDURES CALLED - NONE

CALLING PROCEDURES - TSAESHK DSAESHK

PROCEDURE PROCLIB (XSGLJET)

****PROCEDURE XSGLJET

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE SGLJET MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
SFIELD(FREQ)	ARRAY (FREQS)
SFIELD(THETA)	ARRAY (THETAS)
SFIELD(PHI)	ARRAY (PHIS)
SAE (MTH)	SAEDAT(MTHTMP)
SAE (PDF)	SAEDAT(PDFTMP)
SAE (NDF)	SAEDAT(NDFTMP)
SAE (SCF)	SAEDAT(SCFTMP)
SAE (OM)	SAEDAT(OMTMP)
SAE (SJC)	SAEDAT(SJCTMP)

OUTPUT
USER PARAMETERS - SEE SGLJET MODULE

MEMBERS - NO NAME OVERRIDE REQUIRED FOR SGLJET(XXXNNN)

LOCAL - NONE

MODULES CALLED - SGLJET

PROCEDURES CALLED - NONE

CALLING PROCEDURES - TSGLJET DSGLJET

PROCEDURE PROCLIB (XSMBTUR)

****PROCEDURE XSMBTUR

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE SMBTUR MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/00)
DSW(L03/00/05)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
SFIELD(FREQ)	ARRAY (FREQS)
SFIELD(THETA)	ARRAY (THETAS)
SFIELD(PHI)	ARRAY (PHIS)

OUTPUT
USER PARAMETERS - SEE SMBTUR MODULE

MEMBERS - NO NAME OVERRIDE REQUIRED FOR SMBTUR(XXXNNN)

LOCAL - NONE

MODULES CALLED - SMBTUR

PROCEDURES CALLED - NONE

CALLING PROCEDURES - DSMBTUR TSMBTUR

PROCEDURE PROCLIB (XSTNJET)

****PROCEDURE XSTNJET

PURPOSE - TO PROVIDE ALTERNATE NAME DESIGNATIONS FOR THE UNIT MEMBERS AND PARAMETERS OF THE STNJET MODULE. THIS ALLOWS PROCEDURES TO BE WRITTEN USING THE SAME DEFAULT NAMES AS THE ORIGINAL MODULE.

AUTHOR - DSW(L03/00/05)

INPUT
USER PARAMETERS

OLD NAME	NEW NAME
IPRINT	PRT

MEMBERS

OLD NAME	NEW NAME
SFIELD(FREQ)	ARRAY (FREQS)
SFIELD(THETA)	ARRAY (THETAS)
SFIELD(PHI)	ARRAY (PHIS)
STNTBL(JDF)	STNDAT(JDFS)
STNTBL(SDF)	STNDAT(SDFS)
STNTBL(FSP)	STNDAT(FSPS)

OUTPUT
USER PARAMETERS - SEE STNJET MODULE

MEMBERS - NO NAME OVERRIDE REQUIRED FOR STNJET(XXXNNN)

LOCAL - NONE

MODULES CALLED - STNJET

PROCEDURES CALLED - NONE

CALLING PROCEDURES - DSTNJET TSTNJET

6 ANOPP PERMANENT DATA BASE

6.1 OVERVIEW

The noise prediction methods used in ANOPP are largely empirical and require tables of data that are used during execution. In order to provide an efficient method of handling these tables the ANOPP permanent data base was developed. In addition to the data required by the functional modules, standard tables of output data are available to substitute for the functional modules which created them. Use of these tables would reduce costs. The ANOPP procedure library is available and should be used when possible; however, there may be cases where the standard procedures are not sufficient. For efficiency, additional user data and procedures can and should be maintained with the ANOPP data base management system. There are at present many user-defined data bases that are available to the ANOPP user. For information concerning the user data bases the ANOPP user is referred to the specific application.

6.2 ANOPP PERMANENT DATA BASE DESCRIPTIONS

The following table describes the data available at the time of publication of this manual. It describes the data available for Level 03/00/00.

Data unit	Data member	Description
ATMOS	ABS	Data table of atmospheric absorption for ANSI standard absorption method[a] at standard conditions as output by FM ABS.
ATMOS	SAEABS	Data table of atmospheric absorption for SAE ARP 866A[b] method at standard conditions as output by FM ABS.
ATMOS	STRD	Data table of atmospheric model data for standard conditions as required for input to FM ABS.
ICAOSAE	MTH	Data table of forward velocity index as a function of directivity angle in degrees as described in appendix A of ARP 876.[c]
ICAOSAE	NDF	Data table of common logarithms of the normalized directivity function for circular jets as described in appendix A of ARP 876.[c]
ICAOSAE	OM	Data table of jet variable density exponent as described in appendix A of ARP 876.[c]
ICAOSAE	PDF	Data table of common logarithms of the power deviation factor as described in appendix A of ARP 876.[c]
ICAOSAE	SCF	Data table of Strouhal number correction factor as described in appendix A of ARP 876.[c]

[a]Reference 3.
[b]Reference 4.
[c]Reference 5.

ORIGINAL PAGE IS
OF POOR QUALITY

Data unit	Data member	Description
ICAOSAE	SJC	Data table of normalized spectral distribution factor as described in appendix A of ARP 876.[a]
JWRCOAN	ALPHA	Table of common logarithms of the peak Strouhal number as described in section 4.8.7.
JWRCOAN	CBF	Table of overall power level benefit for coannular nozzle as described in section 4.8.7.
JWRCOAN	DIR	Table for 10 times the common logarithms of normalized directivity function for coannular jets as function of directivity angle and common logarithms of ratio of jet velocity to speed of sound as described in section 4.8.7.
JWRCOAN	MTH	Table of forward flight effect correction as a function of directivity angle in degrees.
JWRCOAN	NSF	Table of common logarithms of normalized spectrum function as function of common logarithm of Strouhal number and directivity angle in degrees.
JWRCOAN	OM	Table of jet variable density exponent as function of common logarithm of ratio of jet velocity to speed of sound as described in section 4.8.7.
JWRCOAN	PDF	Table of common logarithms of power deviation factor of SAE method as described in section 4.8.7.
JWRCOAN	S1	Table of common logarithms of peak Strouhal number for first component of spectrum as function of directivity angle and common logarithm of ratio of jet velocity to speed of sound as described in section 4.8.7.
JWRCOAN	S2	Table of peak Strouhal number for second component of spectrum as function of directivity angle and common logarithm of ratio of jet velocity to speed of sound as described in section 4.8.7.
PROCLIB	ATMDAT	See section 5.2.
PROCLIB	ATMSTD	
PROCLIB	DCNLJET	
PROCLIB	DFNKAFM	
PROCLIB	DGECOR	
PROCLIB	DGETUR	

[a]Reference 5.

ORIGINAL PAGE IS
OF POOR QUALITY

Data unit	Data member	Description
PROCLIB	DHDNFAN	See section 5.2.
PROCLIB	DSAESHK	
PROCLIB	DSGLJET	
PROCLIB	DSMBTUR	
PROCLIB	DSTNJET	
PROCLIB	LISTALL	
PROCLIB	LISTONE	
PROCLIB	PROMUL	
PROCLIB	PROSGL	
PROCLIB	TCNLJET	
PROCLIB	TFNKAFM	
PROCLIB	TGECOR	
PROCLIB	TGETUR	
PROCLIB	THDNFAN	
PROCLIB	TSAESHK	
PROCLIB	TSGLJET	
PROCLIB	TSMBTUR	
PROCLIB	TSTNJET	
PROCLIB	XABS	
PROCLIB	XATM	
PROCLIB	XCNLJET	
PROCLIB	XEFF	
PROCLIB	XFNKAFM	
PROCLIB	XGECOR	
PROCLIB	XGENSUP	
PROCLIB	XGETUR	
PROCLIB	XGRA	
PROCLIB	XHDNFAN	
PROCLIB	XLEV	
PROCLIB	XPREAFM	
PROCLIB	XPRECOR	
PROCLIB	XPREFAN	
PROCLIB	XPREJET	

ORIGINAL PAGE IS
OF POOR QUALITY

Data unit	Data member	Description
PROCLIB	XPRETUR	See section 5.2.
PROCLIB	XPRO	
PROCLIB	XPROSGL	
PROCLIB	XSAESHK	
PROCLIB	XSGLJET	
PROCLIB	XSMBTUR	
PROCLIB	XSTNJET	
SAE	MTH	Table of forward-velocity index as a function of directivity angle in degrees.
SAE	NDF	Table of common logarithms of normalized directivity function for circular jets as a function of (1) directivity angle in degrees and (2) common logarithm of ratio of jet velocity to speed of sound.
SAE	OM	Table of jet variable-density exponent as function of common logarithms of ratio of jet velocity to speed of sound.
SAE	PDF	Table of common logarithms of power deviation factor from V^8 law as a function of common logarithm of ratio of jet velocity to speed of sound.
SAE	SCF	Table of Strouhal number correction factor as a function of ratio of jet velocity to speed of sound and directivity angle.
SAE	SJC	Table of normalized spectral distribution factor as a function of (1) common logarithm of corrected Strouhal number, (2) ratio of jet velocity to speed of sound, (3) directivity angle from jet inlet in degrees, and (4) ratio of jet temperature to ambient temperature.
SFIELD	FREQ	Data member containing 1/3-octave-band center frequencies from 50 Hz to 10 kHz.
SFIELD	PHI	Data member containing azimuthal directivity angle in degrees most commonly used. It contains only one value, zero.
SFIELD	THETA	Data member containing polar directivity angles in degrees most commonly used. It contains the values from 20° to 170° in 10° increments.
STNTBL	FSP	Data table containing frequency shift parameter as described in section 4.8.6.
STNTBL	SDF	Data table containing shock noise spectral distribution factor as described in section 4.8.6.
STNTBL	JDF	Data table of jet mixing noise spectral distribution factor as described in section 4.8.6.

7 REFERENCES

1. Zorumski, William E.: Aircraft Noise Prediction Program Theoretical Manual. NASA TM-83199, 1981.

2. Gillian, Ronnie E.; Brown, Christine G.; Bartlett, Robert W.; and Baucom, Patricia H.: ANOPP Programmers' Reference Manual for the Executive System. NASA TM X-74029, 1977.

3. American National Standard Method for the Calculation of the Absorption of Sound by the Atmosphere. ANSI S1.26-1978 (ASA 23-1978), American Natl. Stand. Inst., Inc., June 23, 1978.

4. Standard Values of Atmospheric Absorption as a Function of Temperature and Humidity. ARP 866A, Soc. Automot. Eng., Mar. 15, 1975.

5. Gas Turbine Jet Exhaust Noise Prediction. ARP 876, Soc. Automot. Eng., Mar. 1978.

APPENDIX A

GLOSSARY OF ANOPP TERMS AND ACRONYMS

A.1 TERMS

Alternate names - The set of names, established on the EXECUTE control statement, which corresponds to a set of reference names. The set of alternate names is available for retrieval by a functional or an executive system module during the execution of that module.

ANOPP run - One execution of ANOPP, the sequence of actions defined by a user's card-image primary input stream.

Archive - The read-only status, established with the ARCHIVE control statement, whereby the future generation of additional members on a specified data unit is prohibited.

Checkpoint run - An ANOPP run during which a checkpoint file is built. Each cycle generated on the file preserves a run operating environment that may be reestablished during the initiation of a subsequent restart run.

Control statement (CS) - One or more card images which define a particular action to be performed by the executive system during an ANOPP run.

DATA - The name of the data unit created by the ANOPP executive system and used to store data members created with DATA control statements encountered in the primary input stream.

Data base - A collection of data units residing on random-access mass-storage files or sequential library files. (See permanent data base, run-life data base, and user's data base.)

Data element - One or more words residing on a formatted data record. Its data type and number of words are determined by the corresponding element code in the member format.

Data member - Any member other than a procedure member or a table member. (See member.)

Data record - An ordered set of data elements or words residing on a member. The record may be unformatted or it may be formatted as fixed, variable, or card image according to the member format.

Data table - A table of data available to the functional module for processing. It resides on a one-record member having an internal format corresponding to a defined data table type.

Data unit - The highest level of the ANOPP data base structure that can be referenced directly (by name) during an ANOPP run. It is a set of named numbers and it resides alone on an external random-access mass-storage file.

APPENDIX A

Data unit directory (DUD) - The executive system directory containing an entry for each data unit currently established in a run, that is, the run-life data base.

Delimiter - A character on a control statement card image that terminates a field. Special delimiters may constitute a single-character field. The comma and space are valid delimiters on all control statements. Specific control statements allow or require the use of the following special delimiter characters: ( ) = / * and ;.

Element - A word or group of words on a record. (See data element.)

Element code - The descriptor within a data member format specification used to identify the data type of an element within the data member's records.

Established data unit - A run-life data unit.

Established file - A rotating mass-storage file or tape file assigned to the user's job in the external system.

Executive system - The ANOPP modules that manage the sequence of processing during a run based on the user's control statement streams.

External control card - The card-image directives required by the external system to define and execute a job on the host computer.

External file - A file residing on magnetic tape or on rotating mass-storage devices under the management of the external system.

External system - The operating system that controls all job processing on the host computer.

Functional module (FM) - A named set of one or more modules recognized by the ANOPP executive system that performs a specific utility or noise prediction task. A functional module is called into execution during a run with an EXECUTE control statement.

Global dynamic storage (GDS) - A section of free core storage defined and maintained by the executive system for storage of directories and tables. It resides at the end of a user's job field throughout the life of an ANOPP run. Storage size is controlled by an initialization parameter that may be assigned a value with the ANOPP control statement or the RSTRT control statement.

APPENDIX A

Initialization parameter - Run control parameter maintained by the executive system and used to initialize an ANOPP run's operating environment. The user may assign values with an ANOPP control statement or with the RSTRT control statement in a restart run.

Input stream - A set of control statements that defines a sequence of actions to occur during an ANOPP run. (See primary input stream and secondary input stream.)

Inserted primary input stream - A control statement stream optionally inserted and executed in a restart run. (See primary input stream.)

Job deck - The entire card-image deck (or file) consisting of the external control cards and primary input stream(s) required to execute an ANOPP job.

Library file - See sequential library file.

Local dynamic storage (LDS) - A section of free core storage maintained by the executive system that begins with the word following the longest segment currently being executed and ends at the start of global dynamic storage.

Member - The lowest level of the ANOPP data base structure that can be referenced directly (by name) during an ANOPP run. It is a logically contiguous set of records and it resides on a data unit.

Member format - A specification which describes the composition of data records residing on a member.

Module - A FORTRAN or COMPASS subprogram that is part of the ANOPP system.

Parameter - See initialization parameter, system parameter, and user parameter.

Permanent data base - The collection of data unit files and sequential library files established, maintained, and installed as part of the ANOPP system.

Permanent file - An external file established permanently at the host computer installation.

Primary edit phase - The section of an ANOPP run when the primary input stream is checked for correct syntax. Continuation of the run depends on successful completion of this phase; otherwise, an error message is written and the run is terminated.

Primary input stream - The control statement card deck or card-image file required as input to each ANOPP run.

APPENDIX A

Procedure member - A member containing a set of card-image records called a secondary input stream. Each record is a valid control statement card image. (See secondary input stream.)

Random-access file - The type of external mass-storage file required as the residence of an ANOPP run-life data unit. It allows the direct addressing of the unit's members.

Record - An ordered set of elements or words residing on a member. (See data record.)

Restart run - An ANOPP run in which the RSTRT control statement is the first control statement in the primary input stream. At run initiation, the operating environment that existed in a previous ANOPP run and was preserved on an external checkpoint file is reestablished.

Run-control parameter - An initialization or system parameter maintained by the executive system and used to control operating environment during an ANOPP run.

Run-life data base - All data units established in an ANOPP run at a given point in time. Each unit resides on a separate external random-access file and has an entry in the data unit directory.

Run-life data unit - A data unit currently established in the run-life data base.

Run-life member - A member residing on a run-life unit.

Scratch file - A temporary file on which a data unit or library that will exist only during the current ANOPP job or run is established.

Secondary edit phase - The section of an ANOPP run when the secondary input stream is checked for correct syntax. Continuation of the run depends on successful completion of this phase; otherwise, an error message is written and the run is terminated.

Secondary input stream - A set of valid control statements that resides on a card-image procedure member and to which run processing control may be temporarily transferred via the execution of the CALL control statement.

Sequential library file - A sequential file (tape or mass-storage) on which copies of run-life data units are stored with an UNLOAD control statement and from which run-life data units may subsequently be generated with a LOAD control statement.

APPENDIX A

System parameter - Run control parameter maintained by the executive system used to control characteristics of an ANOPP run's operating environment. They may be assigned new values repeatedly during a run with the SETSYS control statement. Several system parameters may also be assigned initial values with the ANOPP control statement (or RSTRT control statement in restart runs).

Table member - A member containing a one-record member corresponding to a defined data table type.

User parameter - Array of one or more values which, once established, remains available to the user throughout the ANOPP run. The value(s) of a user parameter may be established or changed via the PARAM control statement, the EVALUATE control statement, or during the execution of a functional module. The value(s) of a user parameter may be retrieved during functional module processing and may be tested by the user with the IF control statement.

User parameter table (UPT) - A table built and maintained by the executive system which links the user parameter name with its current value, type, and number of elements.

User's data base - The collection of data unit files and sequential library files established and maintained by the user at the host computer installation.

XRUNIT - A data unit created on a scratch file by the ANOPP executive system. It is used during checkpoint and restart runs.

XSUNIT - A data unit created on a scratch file by the ANOPP executive system. It is reserved for ANOPP executive system usage.

A.2 ACRONYMS

ANOPP	Aircraft Noise Prediction Program
CS	control statement
DUD	data unit directory
EPNL	effective perceived noise level
FM	functional module
GDS	global dynamic storage
LDS	local dynamic storage
OASPL	overall sound pressure level
PNL	perceived noise level

APPENDIX A

PNLT	tone-corrected perceived noise level
SPL	sound pressure level
UPT	user parameter table

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX B

CONTROL STATEMENT INDEX

Control statement name	Section
ANOPP	3.1.3.2
ARCHIVE	3.5.3
ATTACH	3.5.1.1
CALL	3.9.2
CATALOG	3.5.2.1
CKPNT	3.10.2
CONTINUE	3.8.4
CREATE	3.5.1.2
DATA	3.7.2
DETACH	3.5.1.3
DROP	3.6.3
ENDCS	3.1.4.1.2
EXECUTE	3.4.2
EVALUATE	3.3.3
GOTO	3.8.2
IF	3.8.3
LOAD	3.6.1
MEMLIST	3.5.2.2
PARAM	3.3.2
PROCEED	3.8.5
PURGE	3.5.1.4
RSTRT	3.10.3
SETSYS	3.2.2
STARTCS	3.1.4.1.1
TABLE	3.7.3
TABLIST	3.5.2.3
TITLE	3.2.3
UNLOAD	3.6.2
UPDATE	3.7.4.2
UPLIST	3.3.4

APPENDIX C

ERROR DIAGNOSTICS AND RECOVERY TECHNIQUES

C.1 INTRODUCTION

Error messages generated during an ANOPP job can be any one of the following:

1. An external system diagnostic

2. An ANOPP functional module error message

3. An ANOPP executive system diagnostic

The diagnostics produced by the external operating system that concern the user are usually fatal, resulting in job termination. They often relate to an error condition previously reported by ANOPP messages in the same job. If an ANOPP run recovery technique (discussed below) is not appropriate, the user may require either local computer site assistance or personal knowledge of the host operating system. The applicable section of appendix D covers the user's host computer operating system and includes a list of the manuals required for interrogation of external system diagnostics.

Error messages produced by specific functional modules are documented in section 4. They report the occurrence of a condition meaningful to the noise prediction or utility function of the executing module. Errors produced by functional modules are nonfatal and result in subsequent run processing (following termination of the FM) being under the control of the JCON system parameter. (See section 3.2.) Recovery techniques discussed below for ANOPP executive system diagnostics are also applicable for many FM-produced error messages.

C.2 ANOPP EXECUTIVE SYSTEM DIAGNOSTICS

The major executive subsystems that control ANOPP run initialization, CS editing, and CS and FM processing phases produce diagnostics. They report fatal errors resulting in run termination and nonfatal errors resulting in the current value of the JCON system parameter controlling subsequent run processing.

All ANOPP executive system diagnostics have the following general form:

prefix (ERROR NUMBER v) *** (CALLER sysrtn) error message

where prefix identifies the executive subsystem that produced the diagnostic, v is the number of the specified system error, sysrtn is the system subroutine that called the error processor, and error message is the error condition that occurred. The diagnostics prefixes and the executive subsystems that produce them are:

```
*** EXEC     ERROR - the executive management system
*** DBM      ERROR - the data base management (DBM) member manager
*** LOAD     ERROR - the DBM LOAD CS processor
*** UNLOAD   ERROR - the DBM UNLOAD CS processor
*** DTM      ERROR - the table manager system
*** XTB      ERROR - the table manager system utilities
*** DSM      ERROR - the dynamic storage manager
*** UPDATE   ERROR - the UPDATE CS processor
*** UTILITY  ERROR - the internal system utilities
*** EVALUATE ERROR - the evaluate CS processor
```

Most error messages the normal ANOPP user encounters relate to a specific condition that the user can correct through changes to his ANOPP run input control statements and/or to his external control cards. However, some error messages are applicable to internal execution problems. These are meaningful to and require the attention of a programmer maintaining the ANOPP system or a specific functional module. The following table lists the executive system diagnostics and suggests recovery techniques to be used:

Executive system diagnostics	Type of error	Recovery technique
*** EXEC ERROR...	Input stream edit phase errors	Add or correct specific CS(CS's)
	Insufficient dynamic storage area	Increase job field length via external control card specification; increase run's GDS area (controlled by initialization parameter LENGL) via ANOPP CS or RSTRT CS
*** DBM ERROR...	Insufficient DUD space	Increase DUD size (controlled by initialization parameter NAEUD) via ANOPP CS or RSTRT CS
	Insufficient data table directory (DTD) space	Increase DTD size (controlled by initialization parameter NAETD) via ANOPP CS or RSTRT CS
	Insufficient dynamic storage area	Increase job field length via external control card specification; increase run's GDS area (controlled by initialization parameter LENGL) via ANOPP CS or RSTRT CS
	Missing or erroneously identified run-life data base unit or member	Add or correct CS's and/or external control card(s) that establish or specify the data base component
	Erroneous data found on specified unit or member	Reexamine data requirements; interrogate content of erroneous component via CATALOG CS, MEMLIST CS, or TABLIST CS

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX C

Executive system diagnostics	Type of error	Recovery technique
*** LOAD ERROR... and *** UNLOAD ERROR...	Insufficient DDB space	Increase DDB size (controlled by initialization parameter NAKDD) via ANOPP CS or RSTRT CS
	Insufficient LDS area	Increase job field length via external control card specification
	Missing or erroneously identified external file, data unit, or member	Recheck data requirements; add or correct applicable CS(CS's) and/or external control card(s) establishing or specifying data component; verify current run-life data base via CATALOG CS; verify stored content of library file via separate LOAD CS/CATALOG CS sequence.
*** DTM ERROR... and *** XTM ERROR...	Insufficient dynamic storage area	Increase job field length via external control card specification; increase run's GDS area (controlled by initialization parameter LENGL) via ANOPP CS or RSTRT CS
	TABLE CS and/or table description card-image set syntax errors	Correct applicable CS and/or directives
	Erroneous table identification, definition, or content	Verify table requirements of run or FM; verify content of table member via TABLIST CS; add or correct any CS, directive, or external control card establishing or defining table; assistance of personnel maintaining system or functional module may be required
*** DSM ERROR...	Insufficient core or GDS/LDS overlap	Increase job field length via external control card specification
	Others	Most DSM error diagnostics were intended to assist in system development and debugging; contact ANOPP system maintenance personnel for assistance
*** UPDATE ERROR...	The omission of, or errors in, update member-level and record-level directive(s)	Add or correct the UPDATE CS and/or specific directive(s)
	Insufficient core	Increase job field length via external control card specification
	Missing or erroneously identified run-life data units and members required in the update process	Add or correct the external control cards, CS's, and update directives that establish or identify the data components
	Erroneous data found on specified unit or member	Reexamine data requirements, interrogate content of erroneous components via CATALOG CS, MEMLIST CS, or TABLIST CS
*** UTILITY ERROR...	Invalid or erroneous user parameter or member data type or values	Recheck data requirements of the executed FM; assistance from ANOPP system or FM maintenance personnel may be required
*** EVALUATE ERROR...	Invalid or erroneous user parameter	Recheck user parameters

The user is reminded that ANOPP checkpoint and restart processing capabilities allow for the debugging of large ANOPP applications without requiring reexecution of the entire run. XTRACE reports are printed as a result of all fatal and some nonfatal system errors. They are intended to help in the debugging of internal ANOPP system code and should be referenced when the user seeks the assistance of ANOPP system maintenance personnel.

ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX D

GUIDE TO ANOPP OPERATING SYSTEMS

D.1 OVERVIEW

This appendix presents the external control cards used to run an ANOPP job on a particular operating system. No attempt is made to fully document each control card; the user must either possess personal knowledge of the operating system or gain assistance from someone familiar with the operating system. In most cases reference manuals are available to provide complete documentation of the control cards presented.

D.2 GUIDE FOR CONTROL DATA NETWORK OPERATING SYSTEM (NOS)

D.2.1 The Basic ANOPP Job Deck

The basic job deck used to run ANOPP on the Control Data NOS is presented below. The card numbers (cc1, cc2, and so forth) are used to reference the specific control cards in the discussion which follows.

```
cc1     JOBNAME,CM170000,T300.
cc2     USER.
cc3     ATTACH,ANOPP.
cc4     RFL,170000.
cc5     REDUCE(-)
cc6     ANOPP.
cc7     7-8-9      (card-deck record-separator card)
cc8       STARTCS     $
             :     (other ANOPP control statements)
cc9       ENDCS $
cc10    6-7-8-9    (card-deck end-of-information card)
```

cc1 - The job statement is required as the first card in all NOS job decks.

The CM parameter establishes the maximum central memory required to run the job. ANOPP runs normally execute within a CM field length of 170000 (octal) words. The largest run in the job dictates the CM value. The LDS and GDS requirements of a run often require an increase in this value.

The T parameter establishes the maximum run time allowed for the job expressed in number (octal) of computer central processing seconds.

cc2 - The USER card establishes a user's run permission. It may be a USER, ACCOUNT, or CHARGE card. Local site directives must be referenced to determine which is appropriate.

cc3 - This ATTACH card establishes the job assignment of external program file ANOPP and must precede the execution of the first ANOPP run in the job.

cc4 to cc6 - These three cards control a load and execution of ANOPP. The RFL and REDUCE cards establish a central memory field length that will remain under the control of ANOPP during its execution.

cc7 - The record-separator card (7-8-9 multipunch in column 1) separates the NOS control cards record (cc1 to cc6) from the primary input stream (cc8 to cc9) required by the ANOPP run executed via card cc6.

cc8 to cc9 - The user's primary input stream to the single ANOPP run in this job.

cc10 - The end-of-information card (6-7-8-9 multipunch in column 1) required to terminate a NOS job deck.

D.2.2 Additional NOS Control Cards

Additional NOS control cards besides those included in the basic job deck are required when data base items residing on external files are involved. A job deck containing multiple ANOPP runs is presented below. The permanent and user's data base requirements reside on sequential tape and mass-storage files and on random-access mass-storage files. The second ANOPP run in the job creates a new library tape file and a new data unit residence file that is to be permanently retained on the mass-storage devices of the host system.

```
cc1     NEWJOB,CM200000,T500.
cc2     USER.
cc3     ATTACH,ANOPP.
cc4     REQUEST(JETTAPE,VSN=NU9876,...etc)
cc5     ATTACH,JETUN1=ANOPF1.
cc6     RFL,170000.
cc7     REDUCE(-)
cc8     ANOPP.
cc9     RETURN,JETTAPE,JETUN1.
cc10    REQUEST(NEWTAPE,VSN=NU9877,...etc)
cc11    DEFINE(JETUNIT,...etc).
cc12    RFL,200000.
cc13    REDUCE(-)
cc14    ANOPP.
cc15    UNLOAD,NEWTAPE.
cc16    7-8-9    (card-deck record-separator card)
cc17       STARTCS $
cc18       LOAD/JETTAPE/ UNIT2, UNIT3 $
cc19       ATTACH UNIT1/JETUN1/ $
cc20       CREATE FM3DATA/TEMP1/ $
cc21       EXECUTE FM2 $
cc22    7-8-9    (card-deck record-separator card)
cc23       ANOPP LENGL=9000 $
cc24       STARTCS $
cc25       ATTACH FM3DATA/TEMP1/ $
cc26       CREATE UNIT7/JETUNIT/ $
cc27       EXECUTE FM3 $
```

cc28 UNLOAD/NEWTAPE/ $
cc29 6-7-8-9 (card-deck end-of-information card)

cc4 - The REQUEST card assigns sequential library tape file JETTAPE. Data units stored on the library will be loaded in the first ANOPP run (card cc18).

cc5 - This ATTACH card assigns permanent external file ANOPF1 to the job under a local file name JETUN1. The resident data unit UNIT1 will be established in the first ANOPP run via card cc19.

cc6 to cc8 - Loads and executes the first ANOPP run. An RFL (field length) of 170000 is specified. The primary input stream for this run are cards cc17 to cc21.

cc9 - The RETURN card drops the assignment of the files no longer required in the job.

cc10 - The REQUEST card assigns tape NEWTAPE to the job. An ANOPP sequential library will be built on this file via cc28.

cc11 - This DEFINE card establishes a new permanent external file under the name JETUNIT. The data unit to reside on the file will be established in the ANOPP run via card cc26 and built by functional module FM3.

cc12 to cc14 - Load and execute the second ANOPP run. Because this run requires a larger GDS area (set via card cc23) the RFL for the job must be increased to 200000 octal words.

cc15 - The UNLOAD control card completes the external creation of the new sequential library tape NEWTAPE.

cc17 to cc21 - The primary input stream to the first ANOPP run (executed by card cc8) is entered.

cc20 and cc25 - Note that no external NOS control cards establish and maintain the job assignment of temporary file TEMP1. Its resident data unit FM3DATA, built in the first ANOPP run, may still be accessed in the second run if the external file name is known.

cc23 to cc28 - The primary input stream to the second ANOPP run (executed by card cc14) is entered.

APPENDIX E

SUMMARY OF ANOPP FUNCTIONAL MODULES FOR LEVEL 03/00/00

Module name	Module title	Brief description (a)	Calling procedures
ABS	Atmospheric Absorption Module	Computes atmospheric absorption coefficient as function of altitude and frequency using either ANSI or SAE method	ATMDAT
ATM	Atmospheric Module	Computes atmospheric properties as function of altitude using hydrostatic model	ATMDAT
CNLJET	Dual Stream Coannular Jet Noise Module	Predicts noise for a double stream coannular jet with inverted velocity profile using Russell and Pao method	DCNLJET TCNLJET
EFF	Effective Noise Module	Computes EPNL	PROMUL PROSGL
FLI	Flight Dynamics Module	Computes two-degree-of-freedom take-off for aircraft	
FNKAFM	Airframe Noise Module	Predicts noise for airframe using Fink method	DFNKAFM TFNKAFM
GECOR	Combustion Noise Module	Predicts noise for combustor using SAE method	DGECOR TGECOR
GENSUP	General Suppression Module	Applies noise suppression factor to a noise table produced by any noise source module	SDLDIR
GEO	Geometry Module	Calculates source-to-observer geometry	
GETUR	Turbine Noise Module	Predicts noise for turbine using GE method	DGETUR TGETUR
GRA	Ground Reflection and Attenuation Module	Computes ground effects factor using Chien-Soroka theory (method is also directly computed in the Propagation Module)	

[a]For more complete descriptions of the modules and the methods used, see reference 1.

APPENDIX E

Module name	Module title	Brief description (a)	Calling procedures
HDNFAN	Fan Noise Module	Predicts noise for fan using the Heidman method	DHDNFAN THDNFAN
LEV	Noise Levels Module	Computes OASPL, A-weighted SPL, D-weighted SPL, PNL, and PNLT	PROMUL PROSGL
PREAFM	Airframe Noise Parameters Module	Generates noise prediction parameters for airframe noise	TFNKAFM
PRECOR	Core Noise Parameters Module	Generates noise prediction parameters for combustion noise	TGECOR
PREFAN	Fan Noise Parameters Module	Generates noise prediction parameters for fan noise	THDNFAN
PREJET	Jet Noise Parameters Module	Generates noise prediction parameters for jet noise	TCNLJET TSAESHK TSGLJET TSTNJET
PRETUR	Turbine Noise Parameters Module	Generates noise prediction parameters for turbine noise	TGETUR TSMBTUR
PRO	Propagation Module	Generates sound spectra at observer from source tables with application of atmospheric absorption and ground effects	PROMUL PROSGL
SAESHK	Circular Jet Shock Cell Noise Module	Predicts shock noise using SAE method	DSAESHK TSAESHK
SGLJET	Single Stream Circular Jet Noise Module	Predicts jet noise using SAE method	DSGLJET TSGLJET
SMBTUR	Smith and Bushell Turbine Noise Module	Predicts turbine noise using Smith and Bushell method	DSMBTUR TSMBTUR
STNJET	Stone Jet Noise Module	Predicts far-field mean-square acoustic pressure for single stream and coaxial circular jets using Stone method	DSTNJET TSTNJET

[a]For more complete descriptions of the modules and the methods used, see reference 1.