

(NASA-CR-170899) MOLECULAR CONTAMINATION
MAIN MODEL SUPPORT Final report (Martin
Marietta Corp.) 97 p HC A05/MF A01 CSCL 22B

N84-10174

Unclass
G3/18 42323



MARTIN MARIETTA

MCR-83-640

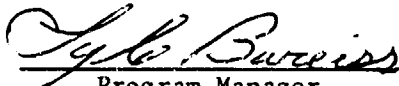
FINAL REPORT

MOLECULAR CONTAMINATION MATH MODEL SUPPORT

Contract NAS8-34945

Author: Russell Wells

Programmer: Joe Pizzicaroli


Program Manager

September 30, 1983

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GEORGE C. MARSHALL SPACE FLIGHT CENTER
MARSHALL SPACE FLIGHT CENTER, AL. 35812

MARTIN MARIETTA DENVER AEROSPACE
BOX 179
DENVER, COLORADO 80201

Table of Contents

	<u>Page</u>
1.0 SUMMARY	1
2.0 INTRODUCTION	2
3.0 PROGRAM DESCRIPTION	3
3.1 Program Overview	3
3.2 Program Subroutines	5
3.3 Input Description	8
3.4 Output Description	14
3.5 Resource Estimates	14
4.0 FUTURE ENHANCEMENTS	15
References	17
Appendices	
A - Minimum Input Sample Case	18
B - All Program Options Sample Case	27
C - Program Listing	46

List of Figures

		<u>Page</u>
1	Program Overview	4
2	Rotations Defining Receive Orientation	9

List of Tables

1	Program Control Parameters	10
2	Geometric Parameters	12
3	Contaminant Parameters	12
4	Mission Parameters	13

1.0 SUMMARY

The purpose of this study is to provide user's training/liaison with personnel at NASA MSFC onto the operation and utilization of the SPACE II contamination analysis program and to develop a preliminary preprocessor for SPACE II to streamline its operation and enhance its "user friendliness". All tasks have been completed within the constraints of the contract. This report summarizes the preliminary SPACE II preprocessor and documents its current status. Although it is not completely operational, it should provide sufficient utility to allow evaluation by MSFC. If the preprocessor discussed herein is deemed viable, it is recommended that further refinements and enhancements be made to insure complete, user friendly operation.

This computer program users' manual describes the operation and features of a preprocessor for the Shuttle/Payload Contamination Evaluation Program Version II - The SPACE II Computer Program (Reference 1). The purpose of a preprocessor is to collect input from a user by an interactive dialogue. The preprocessor then generates an input file that will execute the SPACE II program for a case of interest. The interactive dialogue includes logical questions, multiple choice questions and definitive questions; the latter requires a numerical response.

The preprocessor is built as a baseline model so the basic principle can be evaluated before proceeding with the additional development work that will make it fully operational. As a baseline model it will address the first order input instructions and create an input file that will execute the SPACE II program. The input instructions that the preprocessor will not address and the scope of a fully operational preprocessor are also discussed.

2.0 INTRODUCTION

SPACE II was developed to provide the user with a flexible analytical tool with which to predict the external self-induced molecular contaminant environment of a space vehicle during its orbital operations. It mathematically synthesizes the contaminant environment due to material sources of Spacelab, the Space Shuttle Orbiter and any other spacecraft configuration. It predicts surface deposition by direct and return flux transport on surfaces and molecular column density within any selected line-of-sight. The user has an option to modify geometric configurations, outgassing data and mission parameters through the proper program commands.

The current flexible design of SPACE II makes it possible to modify the basic geometric model; Spacelab in the Shuttle Orbiter bay. Spacelab configuration data is input to the program through mass transport factor files known as TAPE 12, 14 or 15. Another method is through formatted input cards but this is so seldom used that it is not a feature of the preprocessor. These tapes are made using the TRASYS II computer program (Reference 2) using a radiation analogue to Lambertian mass emission from outgassing materials.

A user should first conduct an audit of existing Orbiter, Spacelab and payload input tapes. If the appropriate tapes are available then the user can proceed directly to the baseline preprocessor and construct a SPACE II input file. If a user is to evaluate a new payload in the Orbiter payload bay, then the existing Orbiter TAPE14 will apply. The user then must develop a new TAPE15 if there is to be a column density or return flux calculation and a new TAPE12 when direct flux results are of interest. References 1 and 2 will assist the user in developing these additional TAPES.

The baseline preprocessor does not address the development of any input TAPE but requires corresponding information such as geometric node numbers. It is best to have all input TAPES available to the user before using the preprocessor. There is additional information such as node locations and orientations that are available in TRASYS II output which the preprocessor may require depending on the type of calculation results expected from SPACE II. For these reasons it is recommended that all input TAPES be made and computer outputs be available before a user attempts to use the preprocessor.

The following sections discuss the preprocessor program once a user is ready to proceed with developing a SPACE II input file. Section 3 defines the individual subroutines, input instructions and output results. Section 4 describes future work that will enhance the operation and utility of a SPACE II preprocessor. Within the appendices are two sample case outputs; one that develops a limited input file and another that exercises all possible program options, and in addition a listing of the preprocessor source code in FORTRAN IV.

3.0 PROGRAM DESCRIPTION

3.1 Program Overview

The SPACE II preprocessor is an interactive dialogue program that collects input from a user in a real time question and answer session to create an input file for the SPACE II program. There are six principal sections in the program which are utilized in part or whole. These six sections are:

- 1) Initialize - Give default values to all input variables before program execution.
- 2) Control data - Define the program control parameters to determine which contaminant analysis is of interest. These inputs are generally associated with the namelist CONTRL input section of SPACE II.
- 3) Geometric data - Define the surfaces or engines that are sources of contamination and surfaces that are susceptible to contaminants. These inputs are associated with the namelist INPUTA input section of SPACE II.
- 4) Source information - Define the new surfaces by giving its area, material type and location in the payload bay. If the contaminant source in an engine or vent then define its location and orientation. These inputs are associated with the formatted card images at the end of namelist INPUTA input section of SPACE II.
- 5) Mission data - Define the flight mission parameters such as altitude and Orbiter orientation. The inputs are associated with the MPDB namelist input section of SPACE II.
- 6) Build space file - Create an input file for SPACE II using the responses given by the user.

The interrelationship between each of the six principal sections is illustrated in Figure 1. Sections 2 through 5 collect all of the inputs while output is performed in Section 6. There are numerous other input features to the SPACE II program that are not dealt with by the current baseline preprocessor. These include namelist INPUTB, the formatted card images after INPUTB, namelist INPUTC and the formatted card images after namelist MPDB.

The user can also refer to Figure 6-4 of the SPACE II Users Manual for the logical flow that is used to develop a SPACE II input file.

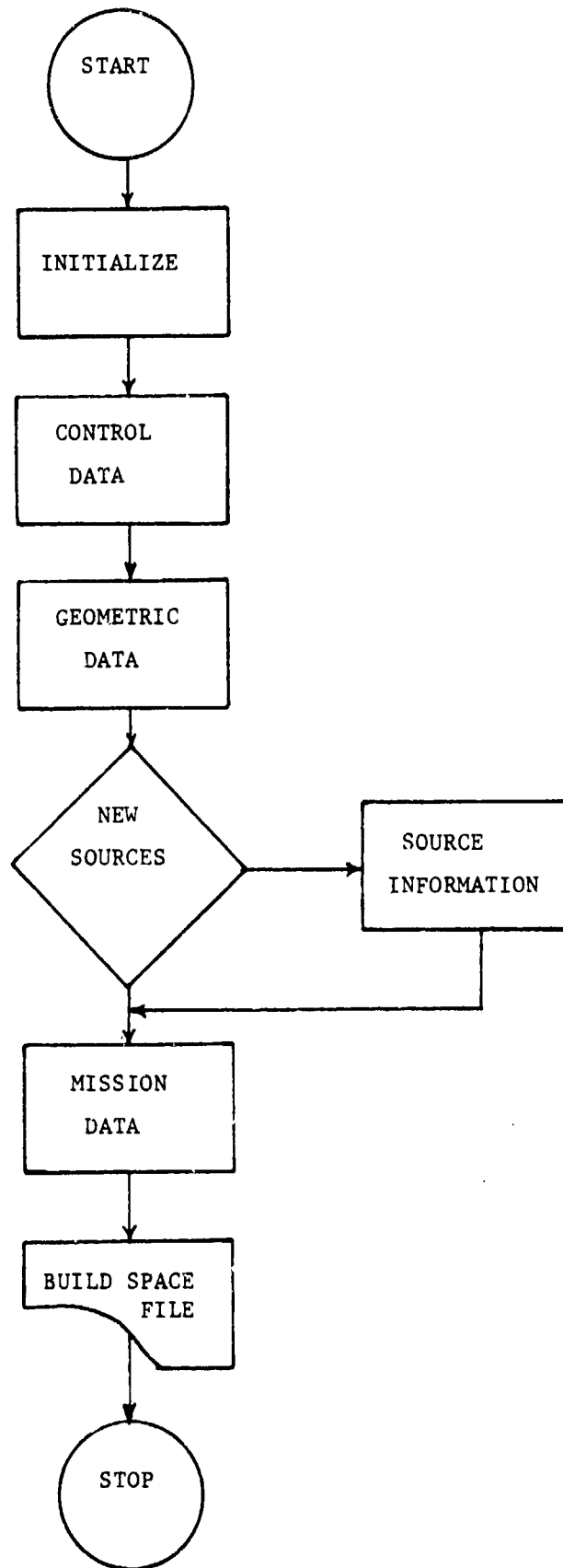


Figure 1 Program Overview

3.2 Program Subroutines

The following subroutines are complete and operational on a FORTRAN IV compiler. They were written with the intent to compile as a simpler FORTRAN source (ANSI) on a mini-computer. There is one subroutine PLUMEX that is not ANSI standard FORTRAN IV but can be made compatible by moving all data statements forward of any executable code and also by separating the mixed mode assignments into separate data statements. The functional responsibilities of each subroutine is given below:

- SPACEP - As a main program it calls these subroutines; INIT, CONTRL, INPUTA, ADDCON, INPUTB, ADDATA, ADDTMP, MPDB and ADDVFS, which collect input from the user for each of the namelist input sections. It then creates a file using the BUILD subroutine that will execute the SPACE computer program for a particular case of interest. Default values are given to the input variables by the subroutine INIT.
- CONTRL - Collect inputs from the user by an interactive dialogue for the CONTRL namelist input section in the SPACE computer program. These inputs will specify the type of analysis, source of contamination, geometric configuration and whether there is new information that is not a part of block data in the SPACE computer program.
- INPUTA - Collect inputs from the user by an interactive dialogue for the INPUTA namelist input section. These inputs identify the node number of surfaces and points that are sources of contamination and also the surfaces that as a receiver are susceptible to a contaminant flux.
- MPDB - Collect input from the user by an interactive dialogue for the MPDB namelist input section. These inputs will determine the flight mission parameters such as orbital altitude, orientation, and velocity. If there is to be a column density or return flux analysis, then the receiver location, orientation and field-of-view are requested from the user.
- ADDCON - Collect input from the user by an interactive dialogue for the formatted input section after INPUTA namelist. These inputs will specify the contaminant source characteristics. If it is a surface source then the type of material, its surface area and the vehicle it belongs on is important. If it is a point source then the type of engine or vent, its location and orientation is requested from the user. The surface source information is usually attached as TAPE4 with the input file while point source characteristics for Orbiter engines are a part of block data in the SPACE computer program.
- BUILD - Create a file that will execute the SPACE computer program using inputs given by the user.

- KYBDIN - Check whether the response made by the user is appropriate with regard to it being an integer number, floating point number, character string or an unrecognizable input.
- ORBTR - This is the Shuttle Orbiter block data which contains descriptive information such as surface area, material name and material location for each of the geometric shapes that collectively simulate the Orbiter.
- INIT - Default values are given to the input variables that result in a simple baseline case.
- PLUMEX - This is the Orbiter engine block data which contains descriptive information such as species mass fractions, engine location and type. It also contains the plume coefficients which describe the emission pattern from each engine with respect to radial distance and angle from the centerline. The structure is not ANSI standard FORTRAN and may need to be modified for use with a simpler compiler.
- CLEAR - Clears the terminal screen on a Beehive terminal for a new question and a user response. The escape code is machine dependant and may need to be modified to an appropriate code for use with a different terminal.
- HEADER - Places a header on the terminal screen before a new question is given to the user. Each header describes which section of the input listing is currently being considered.

Due to time and budget constraints on this task there are some subroutines that are incomplete but these do not adversely affect the operating characteristics of this preprocessor as a baseline model. After these subroutines are complete then the full capability of SPACE input instructions will be in place. The functional responsibility of each subroutine is given below:

- INPUTB - Collect inputs from the user by an interactive dialogue for the INPUTB namelist input section. These inputs will modify either the mass loss characteristics of a surface material or the distribution coefficients of an Orbiter engine. There are default values in the SPACE computer program for these variables. Mass loss characteristics include outgassing rate and time constant while plume distribution coefficients comprise species mass fractions, molecular weight and molecular diameter.
- ADDDATA - Collect inputs from the user by an interactive dialogue for the format data after INPUTB namelist input section. These inputs describe how many changes to the preset values and the new data input for contaminant species, kind of materials, location of materials and type of engines.

- ADDTMP - Collect inputs from the user by an interactive dialogue for the formatted data after INPUTB namelist input section or namelist data in INPUTC input section. These inputs define the temperature of each surface node which is normally attached as TAPE10 with the input file. A temperature file is available as a part of the SPACE II data base for the Orbiter and Skylab nodes.
- ADDVFS - Collect inputs from the user by an interactive dialogue for the formatted data after MPDB namelist input section. These inputs are the mass transport factors between any two surfaces having a view of each other. As a geometric dependent parameter of distance from the source and angle off the centerline, each establishes the fraction of mass that can impinge on a surface. A mass transport factor file is usually attached as TAPE12 with the input file.
- LMOPX - This is the Long Module - One Pallet configuration of Spacelab block data which contains descriptive information such as surface area, material name and material location for each of the 69 geometric shapes and one vent that collectively simulate this Orbiter payload.
- SMPX - This is the Short Module - Three Pallet configuration of Spacelab block data which contains the same type of information as the previous block data on 91 surfaces and one vent.
- FIVPX - This is the Five Pallet configuration of Spacelab block data comprising 82 surfaces and is another payload that is a part of the SPACE data base.
- P801X - This is a P80-1 satellite configuration in block data of the Space Test Program and includes 67 surface nodes.
- DSPIUS - This is a Defense Satellite Program satellite and Inertial Upper Stage transfer vehicle configuration in block data using 59 surfaces to collectively simulate another Orbiter payload.
- TEACH - Instructions are given on request at the beginning of program execution that informs the user about constructing a SPACE input file by using the preprocessor.
- MATLX - This is the material and engine block data which contains information on spacecraft material names and locations, the species that outgas from these materials and also names of Orbiter engines that are a part of the SPACE block data.
- IERROR - Error messages concerning user input instructions that are not compatible or incomplete for a case of interest.

3.3 Input Description

Input Format - User input to an interactive dialogue as a question and answer format is by an integer number, a real number with a decimal point (exponential format is not available), or a character string including a blank line. If the response is unrecognizable or inappropriate due to typographical errors, then the answer prompt returns for additional user input. The baseline preprocessor does not give a descriptive message describing why the user response is inappropriate. But in most cases it is apparent and the location in the source code where descriptive messages will occur is identified at the end of subroutine KYBDIN.

Almost every variable has a default value so not all parameters require a response. Any parameter not given a value will retain the default value. Default values can be found in Reference 1 or in Tables 1 through 4. A sample of user input to the interactive dialogue is shown in the Appendices.

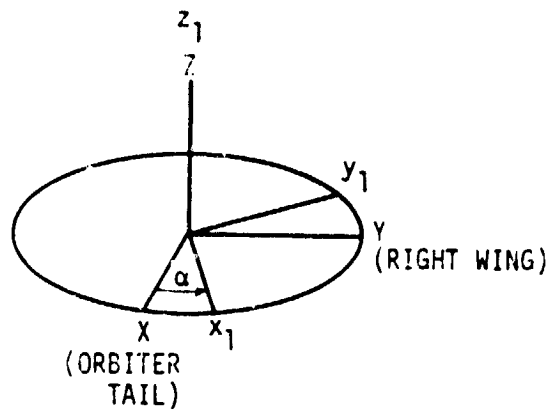
Input Parameters - The input variables that a user will encounter while executing the baseline preprocessor are described in Tables 1 through 4: Table 1 - Program Control Parameters, Table 2 - Geometric Parameters, Table 3 - Contaminant Parameters and Table 4 - Mission Parameters. Under the column heading "Variable Type"; the letter "R" indicates a real number is input which contains a decimal point, the letter "I" indicates an integer number is input which contains no decimal point and the letter "C" indicates a character string is input which contains either alphabetic or numerical symbols. "Default values" are listed under the next column heading as the value of the parameter internally initialized by SPACE before program execution.

There are three basic types of questions in the interactive dialogue; a logical question using a yes or no response, a multiple choice question using an integer input or a definitive question which requires a real number such as orbital altitude or a character string like the case title, for example.

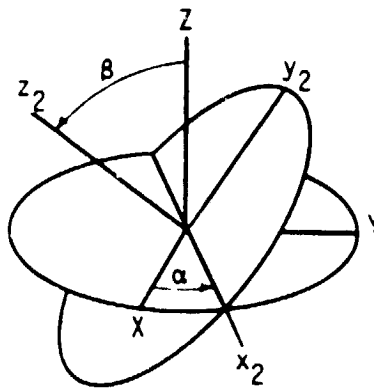
The yes or no response to a logical question can be shortened to the letters "Y" or "N", respectively. Only the first character of the string defines the response so other responses such as "YEA" or "NAY" are acceptable. Logical questions usually coincide with the logical variables in Tables 1 through 4 which have a TRUE or FALSE default value.

A definitive question is the most difficult to answer while working at a computer terminal without some degree of preparation in advance. The response these questions require is simpler when certain items are available. Among the items of information that should be available to the user are: an output listing from TRASYS (Thermal Radiation Analysis System) or similar computer program for the payload configuration, a listing of the additional input files such as TAPE 4 which contains geometric and material information, a blueprint of the payload configuration, a protractor and a scale. These items will provide a basis for defining geometric node numbers, node locations and node orientations. Locations are given with respect to the standard Orbiter coordinate system so it is best to construct the geometric payload model in this system. Orientations are given with respect to the spherical coordinate system shown in Figure 2.

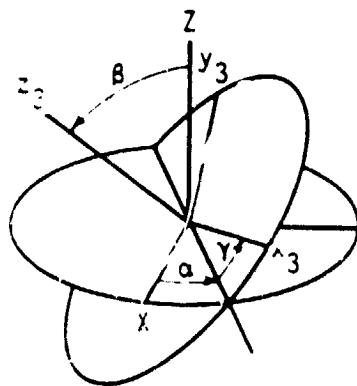
ORIGINAL PAGE IS
OF POOR QUALITY



ROTATION #1



ROTATION #2



ROTATION #3

Figure 2 Rotations Defining Receiver Orientation

There are sixty-five (65) input variables listed in Table 1 through 4 which a user will encounter while executing the baseline preprocessor. Four of these variables will query the user for an input value but subsequently will have no impact on the SPACE II input file. This occurs because other subroutines are incomplete. A star "*" identifies these variables. The description beside each variable is brief and should be used as a guide because the original user manual (Reference 1) for SPACE II contains a complete description. The parameter names are the same as the corresponding parameter in Reference 1 so the extent of the preprocessor can be easily determined by direct comparison.

A virtue of the preprocessor is that it will eliminate the need to input some variables because previous input or information from the built-in data base has resolved their value. REPORT is an example of a variable that is defined according to the user choice of a contaminant transport mechanism. Other variables include SERIES when a built-in spacecraft configuration is chosen, ICCODE, I and K sequence numbers, MOUT1, MOUT2, MED1, MED2, M1 and M2.

A second and equally important benefit of a preprocessor is that it will make the SPACE II program user friendly by creating a manageable set of input variables. The user manual of Reference 1 lists 150 input variables. This development work has reduced that number by more than half to 65 input variables by either neglecting seldom used variables or determining their value from previous input instructions. Future work on a preprocessor concept will eventually shrink the input variable listing to a concise and manageable set.

Table 1 - PROGRAM CONTROL PARAMETERS

<u>Parameter</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
TITLE	C	--	Description of analysis
ORBTR	C	True	There is an STS Orbiter Configuration
PAYLOD	C	False	There is a payload
NEWCON	C	False	New geometric information not found on TAPE4
SPCRFT	C	--	Payload name
SERIES	I	1000	Numbering scheme for payload nodes
NTAPE4	C	False	Geometric data on TAPE4
OUT	C	True	Outgassing contaminants
ED	C	False	Early desorption contaminants

Table 1 - PROGRAM CONTROL PARAMETERS (Cont'd)

<u>Parameter</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
LEAK	C	False	Crew cabin leaks contaminants
REFLCT	C	False	Contaminants can reflect on other surfaces
NRFLCT	I	1	Number of reflections
PLUME	C	False	Engine or vent contaminants
NEWPL	C	False	New engine or vent information not found on TAPES
MCD	C	True	Compute mass/number column density
DIRECT	C	False	Compute direct flux transport
PFAS2	C	False	Compute return flux transport from ambient scattering
RFSS	C	False	Compute return flux transport from contaminant scattering
MACH*	R	1.0	Engine flow Mach for contaminant scattering
TSTARR*	R	--	Engine flow temperature for contaminant scattering
MAXTMP	C	True	Maximum temperature of surfaces
MINTMP	C	False	Minimum temperature of surfaces
ATCODE	I	0	Temperature is on Tape 10 in column one through 5
NEWTNL*	C	False	Temperature input permanently updates Tape 10
NEWTCD*	C	False	Temperature input temporarily updates Tape 10
NEWMFP	C	False	Combine tape 14 and 15 and generate a RMS file

* Although there are inputs for these variables they have no impact on final SPACE II input file.

Table 2 - GEOMETRIC PARAMETERS

<u>Parameter</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
SURFSC	I	--	Eliminate surface nodes that are contaminant sources
SSURFS	I	--	Eliminate surface nodes that reflect contaminant
PNTSC	I	--	Identify engines/vents that are on
ONTIME	F	0.0	Time duration of engine/vent
RECEVR	I	1234	Identify surface nodes susceptible to contaminants
FOVANG	F	180.	Field-of-view half angle for direct flux receivers

Table 3 - CONTAMINANT PARAMETERS

<u>Parameter</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
IDENT	I	--	Surface node number
SECT	C	--	Location of surface
MATRL	C	--	Name of surface material
AREA	F	--	Area of surface (in ²)
CLOC	C	--	Location of engine/vent
CTYPE	C	--	Name of engine/vent
CXLOC	F	--	Orbiter X location of engine/vent (in)
CYLOC	F	--	Orbiter Y location of engine/vent (in)
CZLOC	F	--	Orbiter Z location of engine/vent (in)
CTHETA	F	--	Angle of nozzle centerline to Z-axis (deg)
CPHI	F	--	Angle of nozzle centerline in X-Y plane from +X axis (deg)

Table 4 - MISSION PARAMETERS

<u>Parameter</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
PITCH	F	0.0	Orbiter pitch angle (deg)
YAW	F	0.0	Orbiter yaw angle (deg)
ROLL	F	0.0	Orbiter roll angle (deg)
ALT	F	400.	Orbiter altitude (km)
VA	F	7650.	Orbiter velocity (m/sec)
SUNL	C	False	Low sunspot activity
SUNM	C	True	Medium sunspot activity
SUNH	C	False	High sunspot activity
XO	F	1107.	Receiver X-location (in)
YO	F	0.0	Receiver Y-location (in)
ZO	F	507.	Receiver Z-location (in)
Mass Column Density Calculation Only			
THETAL	F	--	Angle of receiver centerline to Z-axis (deg)
PHIL	F	--	Angle of receiver centerline in X-Y plane from +X axis (deg)
Return Flux Calculation Only			
ALPHA	F	0.0	Angle of receiver centerline to Z-axis (deg)
BETA	F	0.0	Angle of receiver centerline to local +X axis (deg)
GAMMA	F	0.0	Angle of receiver centerline to local +Z axis (deg)
THETA1	F	0.0	Angle off surface Z-axis where field-of-view (FOV) begins (deg)
THETA2	F	10.24	Angle off surface Z-axis where FOV ends (deg)

Table 4 - MISSION PARAMETERS (Cont'd)

<u>Parameter</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
PHI1	F	0.0	Angle off surface X-axis where FOV begins (deg)
PHI2	F	360.	Angle off surface λ -axis where FOV ends (deg)
DTHETA	F	10.24	Increment in THETA angle
DPHI	F	45.	Increment in PHI angle

3.4 Output Description

Output from the baseline preprocessor is in two formats; one is to the terminal screen and one as a local data file. The former has an organization that is similar to the four sections in Section 3.3 "Input Description". Presently the HEADER subroutine is incomplete but in some instances will write a header on the terminal screen before each question is given to the user. The headers that are operational include: "Space II input deck preprocessor" at the beginning of program execution, "Program control parameters" which corresponds to Table 1 input listing while Table 2 input listing is associated with "Contamination source definition" heading. The printout of a sample case in Appendix A and B illustrate this output format.

There is a special character; >E, that is written to the terminal screen to make it completely clear. Whenever this escape E code appears then the previous information on the screen disappears so a new question can be written. Although this code is machine dependent, it can be easily changed to the appropriate code in any particular application.

The local data file called TAPE1 that appears at the end of program execution is shown at the end of each sample problem. This is compatible with the SPACE II input structure and will execute the contamination case of interest.

3.5 Resource Estimates

The SPACE II preprocessor can be compiled and executed with a central memory limit of 70500.

There is one external reference; EOF that is not a part of the preprocessor to determine when there is an end of a data file.

The central-processor-unit (CPU) time to process a particular case depends on the complexity of the input instructions. Experience shows it has always been less than a second.

4.0 Future Enhancements

The genesis of a preprocessor has led to a baseline computer program that creates a limited input data file for the SPACE II computer program. It has also provided an equally important opportunity to consider the needs of a user who has to deal with the intensive input demand of SPACE II. Although the preprocessor is incomplete, it does demonstrate that a software system that can handle every input function leading to a successful SPACE II execution will necessarily be complex. There are a number of ideas that have emerged throughout the development of this preprocessor. These are listed below for consideration:

- 1) User Input - The structure of a preprocessor to SPACE II will be somewhere between simple and complex depending on the needs of each user. If an analysis involves simple geometric configurations, such as an empty Orbiter payload bay and can use previously built data bases, then a simple preprocessor will adequately handle the task. However, if new geometric configurations or data base information is necessary to accomplish a task, such as program particular thermal and outgassing data, then the preprocessor will necessarily have to be complex. A formal definition from the user community with respect to their needs is an important aspect of any future enhancement.
- 2) Planning - A plan is necessary to assure that the work done during an early stage of development will be applicable to the final product at a later date. The plan should consider the demands on a user community, the resources of hardware components that these people utilize, the structure of software systems that manage their data bases and also the interrelationship between those many input resources that a user has at their disposal such as blueprints and other computer programs.
- 3) Tape Development - In addition to the input data file are several tapes which are attached to the job stream before executing the SPACE II program. These include surface materials data (TAPE4), surface temperature data (TAPE10), mass transport factors data between adjacent surfaces (TAPE12) and from surfaces to points in space (TAPE 14 & 15). A preprocessor should help the user build these tapes using results from a radiation computer program like TRASYS.
- 4) Receiver Information - SPACE II input requires the location and orientation of each receiver when a user wants to determine either deposition from return flux contaminant transport or the effect of molecular column density within the field-of-view of an optical component. This information is typically available within the output of their geometric model. A preprocessor should access all relevant information that is a part of the geometric model so a user can be confident that the information is consistent and free of mistakes.
- 5) Graphics - An interactive plotting capability is the best method that a user has to determine whether the input values are correct. It is easy to mistakenly input the wrong values unless there is a means for feedback that will quickly show the user how the computer program will interpret their input instructions.

6) Data base - The SPACE II data base is an integral part of molecular contamination analysis. A preprocessor should show the user what data is available and help them determine whether it is applicable. Although the data base is extensive, it must be functional to save the user time and eliminate redundant work.

It is recommended that MSFC review the concept of a SPACE II preprocessor and in particular the baseline processor which is presented in this report. A preprocessor can train a new engineer in contamination analysis while being an assistance to an engineer with experience by quickly doing work that had previously been done at the expense of hours or days on a computer terminal. It is our impression that a preprocessor is necessary but at this time we would appreciate MSFC comments about their needs and direction on future work.

REFERENCES

- 1) Bareiss, L. E., Jarossy, F. J., Pizzicaroli, J. C., and Owen, N., "Shuttle/Payload Contamination Evaluation Program - The SPACE Computer Program Users Manual", MCR-81-509, Martin Marietta Aerospace, 1981.
- 2) "Thermal Radiation Analysis System (TRASYS)", JSC NAS9-14318, MCR 713-105, Martin Marietta Denver Aerospace, 1975.

APPENDIX A - MINIMUM INPUT SAMPLE CASE

This appendix shows the interactive dialogue that a user will encounter while developing a SPACE II input file for a limited input case. The case corresponds to the first sample problem of Section 5 in Reference 1 which demonstrates the operation of SPACE II when almost all input parameters are given their default value. The problem involves outgassing from the Space lab Long Module/One Pallet configuration at ten hours into a mission. The mass and number column density of outgassing species along a line-of-sight parallel to the Z-axis is computed. After the interactive dialogue is a listing of the input file for SPACE II that was developed by the preprocessor.

Although a minimum input case may seem to imply that very little response is necessary. In practice, however, the user must answer about 25 questions. The preprocessor has this feature because it will query a user about all options in the control parameters input section and then ask a user only about those options in the latter input sections that are relevant to an analysis.

It needs to be mentioned that there is one anomaly in the namelist section INPUTA of the input file. It shows only one Orbiter surface is taken out of the analysis instead of all Orbiter surfaces outside of the payload bay. This discrepancy can be easily removed in later work by allowing for a user to input a range of node numbers. Presently the baseline preprocessor requires that the user input each individual node number. It is cumbersome to sequentially input the numbers one through 155, so only the number 155 was input. Later the file will need editing from one surface with node number 155 to 155 surfaces starting with node number one.

↑>E

ORIGINAL PAGE IS
OF POOR QUALITY

-
- SPACE I I -
- INPUT DECK -
- PREPROCESSOR -

=====

DO YOU NEED INSTRUCTIONS (YES/NO)?

? N
↑>E

INPUT TITLE FOR THIS CASE (UP TO 72 CHARACTERS):

? *** SAMPLE CASE NO. 1 MINIMUM INPUT CASE (DEFAULT PARAMETERS) ***
↑>E

-
- PROGRAM CONTROL -
- PARAMETERS -

DO YOU WANT THE ORBITER CONFIGURATION ACTIVATED?

? Y
↑>E

-
- PROGRAM CONTROL -
- PARAMETERS -

DO YOU WANT A PAYLOAD CONFIGURATION ACTIVATED?

? Y
↑>E

```
-----  
- PROGRAM CONTROL -  
- PARAMETERS -  
-----
```

WHICH OF THE FOLLOWING PAYLOAD CONFIGURATIONS DO YOU WISH:

- 1) LMOP
- 2) SMTP
- 3) FIVP
- 4) P80-1
- 5) DSP/IUS
- 6) NEW CONFIGURATION

? 1
↑>E

```
-----  
- PROGRAM CONTROL -  
- PARAMETERS -  
-----
```

ARE SURFACE SOURCES TO BE ACTIVATED?

? Y
↑>E

```
-----  
- PROGRAM CONTROL -  
- PARAMETERS -  
-----
```


SELECT FROM THE FOLLOWING LIST:

- 1) OUTGASSING ONLY
- 2) EARLY DESORPTION ONLY
- 3) CABIN LEAKAGE ONLY
- 4) OUTGASSING + EARLY DESORPTION
- 5) OUTGASSING + CABIN LEAKAGE
- 6) EARLY DESORPTION + CABIN LEAKAGE
- 7) OUTGASSING + EARLY DESORPTION + CABIN LEAKAGE

? 1
↑>E

```
-----  
- PROGRAM CONTROL -  
- PARAMETERS -  
-----
```

DO YOU WANT MULTIPLE REFLECTIONS EVALUATED?

? N
↑>E

```
-----  
- PROGRAM CONTROL -  
- PARAMETERS -  
-----
```

DO YOU WANT ANY ORBITER ENGINES OR VENTS ACTIVATED?

? N
↑>E

```
-----  
- PROGRAM CONTROL -  
- PARAMETERS -  
-----
```

DO YOU WANT TO INPUT NEW POINT SOURCES?

ORIGINAL PAGE IS
OF POOR QUALITY

? N
↑>E

- PROGRAM CONTROL -
- PARAMETERS -

SELECT TYPE(S) OF ANALYSIS DESIRED:
(TYPE 0 WHEN DONE)

- 1) MASS COLUMN DENSITY (MCD) ALONG A LINE OF SIGHT
- 2) DIRECT FLUX AND/OR DEPOSITION ON A RECEIVING SURFACE
- 3) RETURN FLUX/DEPOSITION DUE TO AMBIENT SCATTERING

? 1
?
↑>E

- PROGRAM CONTROL -
- PARAMETERS -

WHICH OF THE FOLLOWING THERMAL PROFILES FROM
TAPE 10 SHOULD BE USED FOR THIS CASE:

- 1) COLUMN 1 (MAXTMP)
- 2) COLUMN 2 (MINTMP)
- 3) COLUMN 3 (ATCODE=1)
- 3) COLUMN 4 (ATCODE=2)
- 5) COLUMN 5 (ATCODE=3)
- 6) COLUMN 6 (ATCODE=4)
- 7) COLUMN 7 (ATCODE=5)

? 1
↑>E

PROGRAM CONTROL
PARAMETERS

DO YOU WANT TO INPUT THERMAL DATA IN ADDITION TO
THAT CONTAINED ON TAPE 10?

? N
↑>E

PROGRAM CONTROL
PARAMETERS

DO YOU WANT THIS RUN TO GENERATE A TAPE 13 FROM
THE ORBITER TAPE 14 AND PAYLOAD TAPE 15?

? Y
↑>E

CONTAMINATION SOURCE
DEFINITION

DO YOU WANT ANY SURFACES DELETED?

? Y
↑>E

CONTAMINATION SOURCE

- D E F I N I T I O N -

YOU HAVE TWO OPTIONS FOR ZEROING OUT SURFACES.

- 1) IF YOU KNOW THE SEQUENCE NUMBER (NOT THE NODE NUMBER) OF THE SURFACE(S) TO BE ELIMINATED, YOU CAN INPUT THE NUMBER(S) DIRECTLY, TERMINATING THE LIST WITH "0"
- 2) IF YOU KNOW THE NODE NUMBER, BUT NOT THE SEQUENCE NUMBER, YOU CAN ELECT TO REVIEW ALL NODES AND THEIR SEQUENCE NUMBERS, 1 SCREENFUL AT A TIME. IF YOU SELECT THIS OPTION, REVIEW THE LIST AND JOT DOWN THE APPROPRIATE SEQUENCE NUMBERS. AT THE END OF THE REVIEW, THE PROGRAM WILL PROMPT YOU AS IN 1)

PLEASE SELECT EITHER OPTION 1) OR 2)

? 1
↑>E

- C O N T A M I N A T I O N S O U R C E -
- D E F I N I T I O N -

INPUT SEQUENCE NUMBER(S) OF NODES TO BE
ELIMINATED (TYPE 0 WHEN DONE)

? 155
?
↑>E

- C O N T A M I N A T I O N S O U R C E -
- D E F I N I T I O N -

PLEASE ENTER RECEIVING SURFACE NODE NUMBERS (UP
TO 25). ENTER 0 WHEN DONE:

NODE?

? 1234

NODE?

? 0
↑>E

DO YOU WANT TO CHANGE THE ORBITER DEFAULT ATTITUDE?
(PITCH = 0.00, YAW = 0.00, ROLL = 0.00)

? N
↑>E

DO YOU WANT TO CHANGE THE ORBITER DEFAULT ALTITUDE?
(ALT = 400.00 KM)

? N
↑>E

DO YOU WANT TO CHANGE THE ORBITER DEFAULT VELOCITY?
(VEL = 7650. M/SEC)

? N
↑>E

SELECT DESIRED ATMOSPHERE DENSITY:

- 1) LOW
- 2) MEDIUM
- 3) HIGH

ORIGINAL IMAGE IS
OF POOR QUALITY

? 2

*** SAMPLE CASE NO. 1 MINIMUM INPUT CASE (DEFAULT PARAMETERS) ***

```
$CONTRL
ORBITR=.TRUE.,
PAYLOD=.TRUE.,
OUT=.TRUE.,
ED=.FALSE.,
LEAK=.FALSE.,
PLUME=.FALSE.,
MCD=.TRUE.,
DIRECT=.FALSE.,
RFAS2=.FALSE.,
RFSS=.FALSE.,
REFLCT=.FALSE.,
NEWCON=.FALSE.,
NTAPE4=.FALSE.,
NEWTCD=.FALSE.,
NEWTNL=.FALSE.,
NEWMFS=.FALSE.,
NEWMFP=.TRUE.,
MAXTMP=.TRUE.,
REPORT(7)=.TRUE.,REPORT(51)=.TRUE.,
REPORT(35)=.TRUE.,REPORT(37)=.TRUE.,
GO=.TRUE.,
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
$SEND
LMOP 1000
```

```
$INPUTA
SURFSC(155)=0.,
RECEVR(1)=1234,
ICCODE(1)=1*1,
GO=.TRUE.,
```

```
$SEND
```

```
$INPUTB
```

```
$SEND
```

```
$MPDB
```

```
THETAL(1)=0.00,PHIL(1)=0.00,
GO=.TRUE.,
```

```
$SEND
```

```
STOP
```

APPENDIX B - ALL PROGRAM OPTIONS SAMPLE CASE

This appendix shows the interactive dialogue that a user will encounter while developing a SPACE II input file using all program options within the preprocessor. The problem involves all contaminant sources from an Orbiter and new payload called DSCS at ten hours into a mission. The results will include number column density, direct flux, return flux, and second surface flux transport deposition. All surfaces will outgas and show early desorption of light gases. Additional light gases will originate from the crew cabin and two evaporator will be expelling water overboard. There is a new surface and point source not found within the data TAPES that must accompany the input file. The Orbiter is 380 km above Earth and has a ten degree pitch angle below local vertical with a zero yaw and roll angle. After the interactive dialogue is a listing of the input file for SPACE II that was made by the preprocessor.

There are about 75 questions that that user must answer to create this input file. Although this is an unusual case because the different transport mechanisms are typically handled as separate cases. It does demonstrate the variety of questions that a user will encounter. The upper limit on the number of questions will generally depend on how many surface or point sources and receiver locations are defined in the input file. If, however, all source characteristics are defined by the alternate TAPE format then the number of questions will be on the order of this sample problem.

↑>E

ORIGINAL WORK IS
OF POOR QUALITY

```
-----  
-           S P A C E   I I           -  
-           I N P U T   D E C K       -  
-           P R E P R O C E S S O R   -  
-----
```

=====

DO YOU NEED INSTRUCTIONS (YES/NO)?

? N
↑>E

INPUT TITLE FOR THIS CASE (UP TO 72 CHARACTERS):

? *** SAMPLE CASE - EXERCISE ALL PROGRAM OPTIONS ***
↑>E

```
-----  
-           P R O G R A M   C O N T R O L   -  
-           P A R A M E T E R S           -  
-----
```

DO YOU WANT THE ORBITER CONFIGURATION ACTIVATED?

? Y
↑>E

```
-----  
-           P R O G R A M   C O N T R O L   -  
-           P A R A M E T E R S           -  
-----
```


DO YOU WANT A PAYLOAD CONFIGURATION ACTIVATED?

? Y
↑>E

```
-----  
- PROGRAM CONTROL -  
- PARAMETERS -  
-----
```

WHICH OF THE FOLLOWING PAYLOAD CONFIGURATIONS DO YOU WISH:

- 1) LMOP
- 2) SMTP
- 3) FIVP
- 4) P80-1
- 5) DSP/IUS
- 6) NEW CONFIGURATION

? 6
↑>E

```
-----  
- PROGRAM CONTROL -  
- PARAMETERS -  
-----
```

WHAT IS THE NAME OF THE PAYLOAD (6 LETTERS MAX)?

? DSCS
↑>E

```
-----  
- PROGRAM CONTROL -  
- PARAMETERS -  
-----
```

IN WHAT RANGE ARE THE PAYLOAD NODE NUMBERS?

- 1) 1000-1999
- 2) 2000-2999
- 3) 3000-3999
- 4) 4000-4999

? 1
↑>E

```
-----  
- PROGRAM CONTROL -  
- PARAM E T E R S -  
-----
```

HOW WILL PAYLOAD CONFIGURATION BE INPUT?

- 1) VIA TAPE 4
- 2) VIA USER INPUT (THIS SESSION)

? 2
↑>E

```
-----  
- PROGRAM CONTROL -  
- PARAM E T E R S -  
-----
```

ARE SURFACE SOURCES TO BE ACTIVATED?

? Y
↑>E

```
-----  
- PROGRAM CONTROL -  
- PARAM E T E R S -  
-----
```

SELECT FROM THE FOLLOWING LIST:

- 1) OUTGASSING ONLY
- 2) EARLY DESORPTION ONLY
- 3) CABIN LEAKAGE ONLY
- 4) OUTGASSING + EARLY DESORPTION
- 5) OUTGASSING + CABIN LEAKAGE
- 6) EARLY DESORPTION + CABIN LEAKAGE
- 7) OUTGASSING + EARLY DESORPTION + CABIN LEAKAGE

? 7
↑>E

- PROGRAM CONTROL -
- PARAMETERS -

DO YOU WANT MULTIPLE REFLECTIONS EVALUATED?

? Y

HOW MANY REFLECTIONS ARE DESIRED?

? 4
↑>E

- PROGRAM CONTROL -
- PARAMETERS -

DO YOU WANT ANY ORBITER ENGINES OR VENTS ACTIVATED?

? Y
↑>E

- PROGRAM CONTROL -
- PARAMETERS -

DO YOU WANT TO INPUT NEW POINT SOURCES?

? Y

HOW MANY NEW POINT SOURCES WILL THERE BE?

? 1
↑>E

- PROGRAM CONTROL -
- PARAMETERS -

SELECT TYPE(S) OF ANALYSIS DESIRED:
(TYPE 0 WHEN DONE)

- 1) MASS COLUMN DENSITY (MCD) ALONG A LINE OF SIGHT
- 2) DIRECT FLUX AND/OR DEPOSITION ON A RECEIVING SURFACE
- 3) RETURN FLUX/DEPOSITION DUE TO AMBIENT SCATTERING
- 4) RETURN FLUX/DEPOSITION DUE TO SELF SCATTERING

? 1
?
? 2
?
? 3
?
? 0
↑>E

- PROGRAM CONTROL -
- PARAMETERS -

WHICH OF THE FOLLOWING THERMAL PROFILES FROM
TAPE 10 SHOULD BE USED FOR THIS CASE:

- 1) COLUMN 1 (MAXTMP)
- 2) COLUMN 2 (MINTMP)
- 3) COLUMN 3 (ATCODE=1)
- 3) COLUMN 4 (ATCODE=2)
- 5) COLUMN 5 (ATCODE=3)
- 6) COLUMN 6 (ATCODE=4)
- 7) COLUMN 7 (ATCODE=5)

? 1
↑>E

```
-----  
-  
- PROGRAM CONTROL -  
-  
- PARAMETERS -  
-----
```

DO YOU WANT TO INPUT THERMAL DATA IN ADDITION TO
THAT CONTAINED ON TAPE 10?

? N
↑>E

```
-----  
-  
- PROGRAM CONTROL -  
-  
- PARAMETERS -  
-----
```

DO YOU WANT THIS RUN TO GENERATE A TAPE 13 FROM
THE ORBITER TAPE 14 AND PAYLOAD TAPE 15?

? Y
↑>E

ORIGINAL PAGE IS
OF POOR QUALITY

CONTAMINATION SOURCE
DEFINITION

DO YOU WANT ANY SURFACES DELETED?

? Y
↑>E

CONTAMINATION SOURCE
DEFINITION

YOU HAVE TWO OPTIONS FOR ZEROING OUT SURFACES:

- 1) IF YOU KNOW THE SEQUENCE NUMBER (NOT THE NODE NUMBER) OF THE SURFACE(S) TO BE ELIMINATED, YOU CAN INPUT THE NUMBER(S) DIRECTLY, TERMINATING THE LIST WITH "0"
- 2) IF YOU KNOW THE NODE NUMBER, BUT NOT THE SEQUENCE NUMBER, YOU CAN ELECT TO REVIEW ALL NODES AND THEIR SEQUENCE NUMBERS, 1 SCREENFUL AT A TIME. IF YOU SELECT THIS OPTION, REVIEW THE LIST AND JOT DOWN THE APPROPRIATE SEQUENCE NUMBERS. AT THE END OF THE REVIEW, THE PROGRAM WILL PROMPT YOU AS IN 1)

PLEASE SELECT EITHER OPTION 1) OR 2)

? 2
↑>E

* REVIEW CONFIGURATION - PRESS RETURN TO CONTINUE *

SEQ #	NODE	PLACE	MATL	AREA
1	20	RADOOR	TEFLON	12200.0
2	22	RADOOR	TEFLON	12200.0
3	24	RADOOR	TEFLON	12200.0
4	26	RADOOR	TEFLON	12200.0
5	30	RADOOR	TEFLON	12200.0
6	32	RADOOR	TEFLON	12200.0
7	34	RADOOR	TEFLON	12200.0
8	36	RADOOR	TEFLON	12200.0
9	40	RADOOR	TEFLON	25580.0
10	42	RADOOR	TEFLON	25580.0
11	44	RADOOR	TEFLON	25580.0
12	46	RADOOR	TEFLON	25580.0
13	50	RADOOR	TEFLON	25580.0
14	52	RADOOR	TEFLON	25580.0
15	54	RADOOR	TEFLON	25580.0

ORIGINAL PAGE IS
OF POOR QUALITY

16	56	RADOOR	TEFLON	25580.0
17	21	FUSLAG	LRSI	12200.0
18	23	FUSLAG	LRSI	12200.0
19	25	FUSLAG	LRSI	12200.0
20	27	FUSLAG	LRSI	12200.0

?

* REVIEW CONFIGURATION - PRESS RETURN TO CONTINUE *

SEQ #	NODE	PLACE	MATL	AREA
21	31	FUSLAG	LRSI	12200.0
22	33	FUSLAG	LRSI	12200.0
23	35	FUSLAG	LRSI	12200.0
24	37	FUSLAG	LRSI	12200.0
25	41	FUSLAG	LRSI	25580.0
26	43	FUSLAG	LRSI	25580.0
27	45	FUSLAG	LRSI	25580.0
28	47	FUSLAG	LRSI	25580.0
29	51	FUSLAG	LRSI	25580.0
30	53	FUSLAG	LRSI	25580.0
31	55	FUSLAG	LRSI	25580.0
32	57	FUSLAG	LRSI	25580.0
33	202	FUSLAG	LRSI	32520.0
34	203	FUSLAG	LRSI	32520.0
35	230	FUSLAG	LRSI	25730.0
36	240	FUSLAG	LRSI	16340.0
37	241	FUSLAG	LRSI	16340.0
38	250	FUSLAG	LRSI	19580.0
39	260	FUSLAG	LRSI	20240.0
40	301	FUSLAG	LRSI	26600.0

?

* REVIEW CONFIGURATION - PRESS RETURN TO CONTINUE *

SEQ #	NODE	PLACE	MATL	AREA
41	305	FUSLAG	LRSI	30930.0
42	306	FUSLAG	NOMEX	30930.0
43	307	FUSLAG	NOMEX	24770.0
44	311	FUSLAG	LRSI	26600.0
45	315	FUSLAG	LRSI	30930.0
46	316	FUSLAG	NOMEX	30930.0
47	317	FUSLAG	NOMEX	24770.0
48	420	FUSLAG	LRSI	1312.0
49	425	FUSLAG	LRSI	1312.0
50	60	OMS	LRSI	1145.0
51	62	OMS	LRSI	7850.0
52	64	OMS	LRSI	37920.0
53	66	OMS	LRSI	1991.0
54	67	OMS	LRSI	2028.0
55	68	OMS	LRSI	415.0
56	70	OMS	LRSI	895.0
57	72	OMS	LRSI	1406.0
58	74	OMS	LRSI	1312.0
59	76	OMS	LRSI	715.0
60	77	OMS	LRSI	600.0

?

* REVIEW CONFIGURATION - PRESS RETURN TO CONTINUE *

SEQ #	NODE	PLACE	MATL	AREA
61	80	OMS	LRSI	1145.0
62	82	OMS	LRSI	35 7813.0

63	84	OMS	LRSI	37740.0
64	86	OMS	LRSI	1991.0
65	87	OMS	LRSI	2028.0
66	88	OMS	LRSI	415.0
67	90	OMS	LRSI	895.0
68	92	OMS	LRSI	1406.0
69	94	OMS	LRSI	1312.0
70	96	OMS	LRSI	715.0
71	97	OMS	LRSI	601.0
72	100	WING	NOMEX	6356.0
73	102	WING	NOMEX	29590.0
74	104	WING	NOMEX	9125.0
75	110	WING	NOMEX	23340.0
76	112	WING	NOMEX	19280.0
77	115	WING	LRSI	19280.0
78	117	WING	HRSI	5650.0
79	118	WING	HRSI	2508.0
80	119	WING	LRSI	3302.0

?

* REVIEW CONFIGURATION - PRESS RETURN TO CONTINUE *

SEQ #	NODE	PLACE	MATL	AREA
81	121	WING	RCC	2251.0
82	122	WING	RCC	3123.0
83	130	WING	NOMEX	6356.0
84	132	WING	NOMEX	29590.0
85	134	WING	NOMEX	9125.0
86	140	WING	NOMEX	23340.0
87	142	WING	NOMEX	19280.0
88	145	WING	LRSI	19280.0
89	147	WING	HRSI	5650.0
90	148	WING	HRSI	2508.0
91	149	WING	LRSI	3302.0
92	151	WING	RCC	2251.0
93	152	WING	RCC	3123.0
94	106	ELEVON	NOMEX	6499.0
95	107	ELEVON	NOMEX	17210.0
96	136	ELEVON	NOMEX	6499.0
97	137	ELEVON	NOMEX	9125.0
98	450	ELEVON	NOMEX	138.0
99	451	ELEVON	NOMEX	415.0
100	452	ELEVON	NOMEX	692.0

?

* REVIEW CONFIGURATION - PRESS RETURN TO CONTINUE *

SEQ #	NODE	PLACE	MATL	AREA
101	453	ELEVON	NOMEX	960.0
102	454	ELEVON	NOMEX	1246.0
103	455	ELEVON	NOMEX	1523.0
104	456	ELEVON	NOMEX	1800.0
105	457	ELEVON	NOMEX	2076.0
106	458	ELEVON	NOMEX	2353.0
107	459	ELEVON	NOMEX	2630.0
108	460	ELEVON	NOMEX	138.0
109	461	ELEVON	NOMEX	415.0
110	462	ELEVON	NOMEX	692.0
111	463	ELEVON	NOMEX	969.0
112	464	ELEVON	NOMEX	1246.0
113	465	ELEVON	NOMEX	1523.0
114	466	ELEVON	NOMEX	1800.0

115	467	ELEVON	NOMEX	2076.0
116	468	ELEVON	NOMEX	2353.0
117	469	ELEVON	NOMEX	2630.0
118	160	CREW	RCC	7191.0
119	161	CREW	LRSI	9348.0
120	162	CREW	LRSI	9348.0

?

* REVIEW CONFIGURATION - PRESS RETURN TO CONTINUE *

SEQ #	NODE	PLACE	MATL	AREA
121	163	CREW	LRSI	3380.0
122	164	CREW	LRSI	3380.0
123	165	CREW	LRSI	4253.0
124	166	CREW	LRSI	4253.0
125	167	CREW	HRSI	12590.0
126	168	CREW	HRSI	12590.0
127	169	CREW	HRSI	9600.0
128	170	CREW	HRSI	9600.0
129	171	CREW	HRSI	3705.0
130	172	CREW	HRSI	3705.0
131	174	CREW	LRSI	20720.0
132	175	CREW	LRSI	10150.0
133	177	CREW	LRSI	10150.0
134	180	CREW	WINDOW	1424.0
135	181	CREW	WINDOW	1424.0
136	182	CREW	WINDOW	1424.0
137	183	CREW	WINDOW	1424.0
138	184	CREW	WINDOW	1424.0
139	185	CREW	WINDOW	1424.0
140	190	CREW	LRSI	10250.0

?

* REVIEW CONFIGURATION - PRESS RETURN TO CONTINUE *

SEQ #	NODE	PLACE	MATL	AREA
141	380	TAIL	LRSI	16920.0
142	381	TAIL	LRSI	16920.0
143	382	TAIL	LRSI	8833.0
144	383	TAIL	LRSI	8833.0
145	384	TAIL	LRSI	13940.0
146	385	TAIL	LRSI	13940.0
147	386	TAIL	LRSI	6116.0
148	387	TAIL	LRSI	6116.0
149	388	TAIL	LRSI	2744.0
150	389	TAIL	LRSI	2744.0
151	390	TAIL	LRSI	1160.0
152	391	TAIL	LRSI	1160.0
153	392	TAIL	LRSI	3081.0
154	393	TAIL	LRSI	3081.0
155	399	TAIL	HRSI	3823.0

?

↑>E

CONTAMINATION SOURCE
DEFINITION

INPUT SEQUENCE NUMBER(S) OF NODES TO BE
ELIMINATED (TYPE 0 WHEN DONE)

? 101

? 0

↑>E

- CONTAMINATION SOURCE -
- DEFINITION -

DO YOU WANT ANY SURFACES TO BE DELETED AS
SECOND SURFACE SOURCES?

? Y

↑ E

- CONTAMINATION SOURCE -
- DEFINITION -

YOU HAVE TWO OPTIONS FOR ZEROING OUT SURFACES:

- 1) IF YOU KNOW THE SEQUENCE NUMBER (NOT THE NODE NUMBER) OF THE SURFACE(S) TO BE ELIMINATED, YOU CAN INPUT THE NUMBER(S) DIRECTLY, TERMINATING THE LIST WITH "0"
- 2) IF YOU KNOW THE NODE NUMBER, BUT NOT THE SEQUENCE NUMBER, YOU CAN ELECT TO REVIEW ALL NODES AND THEIR SEQUENCE NUMBERS, 1 SCREENFUL AT A TIME. IF YOU SELECT THIS OPTION, REVIEW THE LIST AND JOT DOWN THE APPROPRIATE SEQUENCE NUMBERS. AT THE END OF THE REVIEW, THE PROGRAM WILL PROMPT YOU AS IN 1)

PLEASE SELECT EITHER OPTION 1) OR 2)

? 1

↑>E

 - CONTAMINATION SOURCE -
 - DEFINITION -

INPUT SEQUENCE NUMBER(S) OF NODES TO BE
 ELIMINATED (TYPE 0 WHEN DONE)

? 101
 ? 0
 ↑>E

 - CONTAMINATION SOURCE -
 - DEFINITION -

*** SELECT ACTIVE ORBITER POINT SOURCES ***

YOU HAVE TWO OPTIONS FOR ACTIVATING ORBITER POINT
 SOURCES:

- 1) IF YOU KNOW THE NODE NUMBER OF THE ENGINE
 OR VENT YOU WANT TO ACTIVATE, YOU CAN
 SIMPLY INPUT THE NODE NUMBER AND THE
 DESIRED ON TIME (IN SECONDS)
- 2) IF YOU DON'T KNOW THE NODE NUMBER, YOU CAN ELECT
 TO REVIEW ALL ORBITER PREDEFINED POINT SOURCES,
 JOT DOWN THE DESIRED NODE NUMBERS AND THEN
 BRANCH TO OPTION 1

PLEASE SELECT EITHER OPTION 1) OR 2)

? 2
 ↑>E

* REVIEW ENGINES/VENTS - PRESS RETURN TO CONTINUE *

NODE	LOC	TYPE	X	Y	Z	THETA	PHI
7112	FLF -X	RCS	332.0	-14.0	389.0	0.0	0.0
7122	FCF -X	RCS	332.0	0.0	391.0	0.0	0.0
7132	FRF -X	RCS	332.0	14.0	389.0	0.0	0.0
7123	FLS +Y	RCS	360.0	-47.0	368.0	0.0	0.0
7113	FLS +Y	RCS	360.0	-47.0	354.0	0.0	0.0
7115	FLU +Z	RCS	350.0	-13.0	395.0	0.0	0.0
7125	FCU +Z	RCS	350.0	0.0	395.0	0.0	0.0

7135	FRU +Z	RCS	350.0	13.0	395.0	0.0	0.0
7116	FLD -Z	RCS	333.0	-41.0	381.0	0.0	0.0
7126	FLD -Z	RCS	347.0	-45.0	386.0	0.0	0.0
7144	FRS -Y	RCS	362.0	47.0	368.0	0.0	0.0
7134	FRS -Y	RCS	362.0	47.0	354.0	0.0	0.0
7136	FRD -Z	RCS	333.0	41.0	381.0	0.0	0.0
7146	FRD -Z	RCS	347.0	45.0	386.0	0.0	0.0
7211	ALA +X	RCS	1557.0	-119.0	473.0	0.0	0.0
7231	ALA +X	RCS	1557.0	-132.0	473.0	0.0	0.0
7243	ALS +Y	RCS	1516.0	-123.0	459.0	0.0	0.0
7223	ALS +Y	RCS	1529.0	-123.0	459.0	0.0	0.0
7233	ALS +Y	RCS	1542.0	-122.0	459.0	0.0	0.0
7213	ALS +Y	RCS	1555.0	-122.0	459.0	0.0	0.0

?

* REVIEW ENGINES/VENTS - PRESS RETURN TO CONTINUE *

NODE	LOC	TYPE	X	Y	Z	THETA	PHI
7245	ALU +Z	RCS	1516.0	-132.0	481.0	0.0	0.0
7225	ALU +Z	RCS	1529.0	-132.0	481.0	0.0	0.0
7215	ALU +Z	RCS	1542.0	-132.0	481.0	0.0	0.0
7246	ALD -Z	RCS	1516.0	-112.0	437.0	0.0	0.0
7226	ALD -Z	RCS	1529.0	-111.0	440.0	0.0	0.0
7236	ALD -Z	RCS	1542.0	-110.0	443.0	0.0	0.0
7311	ARA +X	RCS	1557.0	119.0	473.0	0.0	0.0
7331	ARA +X	RCS	1557.0	132.0	473.0	0.0	0.0
7344	ARS -Y	RCS	1516.0	123.0	459.0	0.0	0.0
7324	ARS -Y	RCS	1529.0	123.0	459.0	0.0	0.0
7334	ARS -Y	RCS	1542.0	123.0	459.0	0.0	0.0
7314	ARS -Y	RCS	1555.0	123.0	459.0	0.0	0.0
7345	ARU +Z	RCS	1516.0	132.0	481.0	0.0	0.0
7325	ARU +Z	RCS	1529.0	132.0	481.0	0.0	0.0
7315	ARU +Z	RCS	1542.0	132.0	481.0	0.0	0.0
7346	ARD -Z	RCS	1516.0	112.0	437.0	0.0	0.0
7326	ARD -Z	RCS	1529.0	111.0	440.0	0.0	0.0
7336	ARD -Z	RCS	1542.0	110.0	443.0	0.0	0.0
8116	FLD -Z	VCS	324.0	-46.0	374.0	0.0	0.0
8136	FRD -Z	VCS	324.0	46.0	374.0	0.0	0.0

?

* REVIEW ENGINES/VENTS - PRESS RETURN TO CONTINUE *

NODE	LOC	TYPE	X	Y	Z	THETA	PHI
8257	ALD -Z	VCS	1565.0	-144.0	459.0	0.0	0.0
8258	ALS +Y	VCS	1565.0	-118.0	457.0	0.0	0.0
8357	ARD -Z	VCS	1565.0	144.0	459.0	0.0	0.0
8358	ARS +Y	VCS	1565.0	-118.0	457.0	0.0	0.0
6877	ARS +Y	EVAP1	1506.0	127.0	305.0	0.0	0.0
6879	ALS -Y	EVAP1	1506.0	-127.0	305.0	0.0	0.0
9000	SMALL	OMS	80.0	0.0	0.0	0.0	0.0
9002	LARGE	OMS	180.0	0.0	0.0	0.0	0.0

0

::::

:: 0.0 0.0 0.0 0.0 0.0

?

↑>E

 - CONTAMINATION⁴⁰ SOURCE -

-
- C O N T A M I N A T I O N S O U R C E -
- D E F I N I T I O N -
-

PLEASE ENTER RECEIVING SURFACE NODE NUMBERS (UP TO 25). ENTER 0 WHEN DONE:

NODE?

? 1101
 NODE?

? 0
↑>E

-
- C O N T A M I N A T I O N S O U R C E -
- D E F I N I T I O N -
-

DO YOU WANT THE FIELD OF VIEW OF ANY OF THE RECEIVERS LIMITED TO LESS THAN 90 DEGREES (HALF ANGLE)?

? Y

INPUT RECEIVER NODE NUMBER (0 TO END)

? 1101
INPUT DESIRED FOV LIMITING HALF ANGLE

? 45.

INPUT RECEIVER NODE NUMBER (0 TO END)

? 0
↑>E

*** NEW SURFACE CONFIGURATION INPUTS ***

INPUT NEW NODE NR. ("0" IF DONE)

? 1101

42

INPUT LOCATION (6 LETTERS MAX)

? IUS

INPUT MATERIAL (6 LETTERS MAX)

? PAINT

INPUT SURFACE AREA (SQ. IN.)

? 200.

INPUT NEW NODE NR. ("0" IF DONE)

? 0

↑>E

*** NEW POINT SOURCE INPUTS ***

NEW POINT SOURCE NUMBER 1 - NODE NUMBER 7200:

INPUT POINT SOURCE LOCATION (6 LETTERS MAX)

? GAS

INPUT POINT SOURCE TYPE (6 LETTERS MAX)

? VENT

INPUT POINT SOURCE X-COORDINATE (INCHES)

? 1000.

INPUT POINT SOURCE Y-COORDINATE (INCHES)

? 20.

INPUT POINT SOURCE Z-COORDINATE (INCHES)

? 400.

INPUT POINT SOURCE ORIENTATION ANGLE (THETA)

? 180.

INPUT POINT SOURCE ORIENTATION ANGLE (PHI)

? 45.

↑>E

DO YOU WANT TO CHANGE THE ORBITER DEFAULT ATTITUDE?

(PITCH = 0.00, YAW = 0.00, ROLL = 0.00)

? Y

ENTER PITCH ANGLE IN DEGREES -

? -10.

ENTER YAW ANGLE -

? 0.

ENTER ROLL ANGLE -

? 0.

↑>E

DO YOU WANT TO CHANGE THE ORBITER DEFAULT ALTITUDE?

(ALT = 400.00 KM)

? Y

ENTER DESIRED ALTITUDE (IN KM)

? 380.

↑>E

DO YOU WANT TO CHANGE THE ORBITER DEFAULT VELOCITY?

(VEL = 7650. M/SEC)

? Y

ENTER DESIRED VELOCITY (IN M/SEC)

? 7700.

↑>E

SELECT DESIRED ATMOSPHERE DENSITY:

- 1) LOW
- 2) MEDIUM
- 3) HIGH

? 2

*** SAMPLE CASE - EXERCISE ALL PROGRAM OPTIONS ***

\$CONTRL
ORBTR=.TRUE.,
PAYLOD=.TRUE.,
OUT=.TRUE.,
ED=.TRUE.,
LEAK=.TRUE.,
PLUME=.TRUE.,
MCD=.TRUE.,
DIRECT=.TRUE.,
RFAS2=.TRUE.,
RFSS=.FALSE.,
REFLCT=.TRUE.,
NRFLCT= 4,
NEWCON=.TRUE.,
NTAPE4=.FALSE.,
NEWTCD=.FALSE.,
NEWTNL=.FALSE.,
NEWMFS=.FALSE.,
NEWMFP=.TRUE.,
MAXTMP=.TRUE.,
REPORT(7)=.TRUE.,REPORT(51)=.TRUE.,
REPORT(21)=4*.TRUE.,
REPORT(35)=.TRUE.,REPORT(37)=.TRUE.,
REPORT(45)=.TRUE.,
GO=.TRUE.,

ORIGINAL PAGE 19
OF POOR QUALITY

\$SEND
DSCS 1000
\$INPUTA
SURFSC(101)=0.,
SSURFS(101)=0.,
PNTSC(1)=6877,
PNTSC(2)=6879,
PNTSC(3)=7200,
ONTIME(1)= 1.00,
ONTIME(2)= 1.00,
ONTIME(3)= 1.00,
NEWPL(3)=.TRUE.,
RECEVR(1)=1101,
ICCODE(1)= 1*2,
FOVANG(1)= 45.00,
GO=.TRUE.,

\$SEND
156 1101 IUS PAINT 200.0
3 7200 GAS VENT 1000.0 20.0 400.0 180.0

99999
\$INPUTB
\$SEND
\$MPDB
PITCH=-10.00,
YAW= 0.00,
ROLL= 0.00,
ALT=380.00,
VA=7700.0,
THETAL(1)= 0.00,PHIL(1)= 0.00,
GO=.TRUE.,
\$SEND
STOP

APPENDIX C - PROGRAM LISTING

This appendix contains a listing of the SPACE II preprocessor source code.

PROGRAM SPACEP(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1)

```
*****
*
* SPACE II INPUT DECK PREPROCESSOR *
*
*****
```

MARTIN MARIETTA AEROSPACE
DENVER DIVISION
P.O. BOX 179
DENVER, COLORADO 80201

- DESIGNED AND CODED BY
J. C. PIZZICAROLI
AUGUST 1983

```
=====0002
COMMON /CNTRL/ ORBITR, PAYLOD, OUT, ED, LEAK, PLUME, 0002
+ MCD, DIRECT, RFAS2, RFSS, REFLCT, NRFLCT, 0002
+ NEWCON, NTAPE4, NEWTCD, NEWTNL, NEWMFS, NEWMFP, 0003
+ MINTMP, MAXTMP, ATCODE, R41DEP, GO, REPORT(70) 0003
COMMON/CONFIG/ SURFSC(300), SSURFS(300), PNTSC(50), NEWPL(50), 0003
+ ONTIME(50), RECEVR(25), ICCODE, FOVANG(25), 0003
+ SEPT, NEWDAT, ADSURF, NNEWPL, 0003
+ AL, KTOTAL, NORBPL, ISURF(300), 0003
+ ISSURF(300) 0003
COMMON/CHANGE/ CHNGES, CHNGEK, CHNGEP, CHNGPL 0003
COMMON /RATES/ RATE(25,10), TAU(25,10), TSTART(3), TSTOP(3), 0003
+ AGEORB, AGEPLD, MOUT1, MOUT2, MED1, MED2, 0003
+ M1, M2, AMBWT, AMBDIA, TSTARR, MACH, 0004
+ TIMEOO 0004
COMMON /ORBIT/ ALT, VA, PITCH, YAW, ROLL, SUNL, 0004
+ SUNM, SUNH 0004
COMMON /GEOM/ XORGIN, YORGIN, ZORGIN, X0(25), Y0(25), Z0(25), 0004
+ ALPHA(25), BETA(25), GAMMA(25) 0004
COMMON /INTEG/ THETA(25), PHIL(25), THETA1(25), THETA2(25), 0004
+ DTHETA(25), PHI1(25), PHI2(25), DPHI(25), 0004
+ DOMEGA(25), DS(25), RMAX, NTHETA, 0004
+ NPHI 0004
COMMON /TEMPS/ TEMP(2000) 0005
COMMON/PLUMES/ PLUMEC(10,25), SPECMF(10,25), CXLOC(50), CYLOC(50), 0005
+ CZLOC(50), CTHETA(50), CPHI(50), 0005
+ CIDENT(50) 0005
COMMON/MOLEC/ MOLWT(10), DIA(10) 0005
COMMON /SURFS/IDENT(300), AREA(300) 0005
COMMON/CHAR/ITITLE(72), IRSPNS(80), NEWNAM(6), LTYPE(25), 0005
+ KINDS(25), PLACE(30), SPECIE(10), SECT(300), 0005
+ MATRL(300), NAMEPL, CLOC(50), CTYPE(50), 0005
+ NPLUME(25) 0005
COMMON/CHFLAG/ CHATT, CHALT, CHVEL, CHSUN, CHDS,CHORIG, 0006
+ CHTIM, CHAGE, CHINDX, CHWT, CHDIA, CHLOC(25), 0006
+ CHVIEW(25), CHRATE(25,10), CHTAU(25,10), 0006
+ CHPLUM(10,25), CHMF(10,25) 0006
COMMON/INDX/ INDXSP(25), INDXK(25), INDXP(30), INDXP(25), 0006
+ INDXJT, INDXXT 0006
COMMON/NUCON/NUSECT(6,300), NUMATL(6,300), NULOC(6,50), 0006
+ NUTYPE(6,50), NUSPEC(6,10), NUKIND(6,25), 0006
+ NUPLAC(6,30), NUNPLM(6,25) 0006
REAL ONTIME, MACH, MOLWT 0006
0007
```

```

INTEGER SURFSC, SSURFS, PNTSC, RECEVR, ITITLE, LTYPE, KINDS, 0007
+ PLACE, SPECIE, SECT, MATRL, ATCODE, SERIES, CIDENT, 0007
+ CLOC, CTYPE, CHNGES, CHNGEK, CHNGEP, CHNGPL 0007
LOGICAL ORBITR, PAYLOD, OUT, ED, LEAK, PLUME, MCD, 0007
+ DIRECT, RFAS2, RFSS, REFLCT, NEWCON, NTAPE4, NEWTCD, 0007
+ NEWINL, NEWMFS, NEWMFP, MINTMP, MAXTMP, REPORT, GO, 0007
+ NEWPL, SUNL, SUNM, SUNH, R41DEP, NEWDAT, ADSURF, 0007
+ CHATT, CHALT, CHVEL, CHSUN, CHDS, CHORIG, CHLOC, 0007
+ CHVIEW, CHTIM, CHRATE, CHTAU, CHAGE, CHPLUM, CHMF, 0007
+ CHINDX, CHWT, CHDIA, INDXSP, INDXK, INDXP, INDXPL 0008
=====0008
..PERFORM INITIALIZATION OF DEFAULT VALUES, CONSTANTS, COEFFICIENTS, ET0008
.. CALL INIT 0008
..WRITE PROGRAM HEADER TO SCREEN 0008
.. CALL HEADER(1) ORIGINAL PAGE IS 0008
OF POOR QUALITY 0009
..PROVIDE PROGRAM INSTRUCTIONS IF DESIRED 0009
.. WRITE(6,6010) 0009
6010 FORMAT(/////1X,35HDO YOU NEED INSTRUCTIONS (YES/NO)? ) 0009
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0009
IF(IRSPNS(IFIRST).EQ.1HY) CALL TEACH 0009
.. - BEGIN THE INTERACTIVE QUESTION/ANSWER DIALOGUE PROCESS - 0009
..ASK FOR THE TITLE OF THE USER'S ANALYSIS INPUT DECK 0010
100 CALL CLEAR 0010
WRITE(6,6020) 0010
6020 FORMAT(/////1X,48HINPUT TITLE FOR THIS CASE (UP TO 72 CHARACTERS): 0010
+//) 0010
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0010
I = IFIRST - 1 0010
DO 110 N=1,72 0010
ITITLE(N) = IRSPNS(I + N) 0011
110 CONTINUE 0011
..COLLECT INPUTS FOR NAMELIST $CONTRL 0011
.. 200 CALL CONTRL 0011
..COLLECT INPUTS FOR NAMELIST $INPUTA 0011
.. 300 CALL INPUTA 0011
..IF NEW CONFIGURATION DATA IS REQUESTED, PROMPT FOR THE APPROPRIATE IN 0012
.. 400 NEWDAT = .FALSE. 0012
DO 410 K=1,50 0012
IF(NEWPL(K)) NEWDAT = .TRUE. 0012
410 CONTINUE 0012
IF(NEWDAT.OR.NEWCON) CALL ADDCON 0012
..COLLECT INPUTS FOR NAMELIST $INPUTB 0012
.. 500 CALL INPUTB 0013
.. 0013

```

```

..MODIFY ANY SPECIES, KINDS, PLACE OR PLUME CARDS PER USER REQUESTS 0013
.. 600 NCHG = CHNGES + CHNGEK + CHNGEP + CHNGPL 0013
   IF(NCHG.GT.0) CALL ADDATA 0013
.. 700 IF(NEWTCD.OR.NEWTNL) CALL ADDTMP 0014
.. 800 CALL MPDB 0014
.. 900 IF(NEWMFS) CALL ADDVFS 0014
.. - DONE - NOW BUILD THE SPACE II INPUT DATA FILE 0015
1000 CALL BUILD 0015

STOP 0015
END 0015
SUBROUTINE CONTRL 0015

***** 0015
* 0016
* COLLECT INPUT FOR CONTRL * 0016
* + 0016
* ADDITIONAL INITIALIZATION * 0016
* 0016
***** 0016

===== 0016

COMMON /CNTRL/ ORBITR, PAYLOD, OUT, ED, LEAK, PLUME, 0016
+ MCD, DIRECT, RFAS2, RFSS, REFLECT, NRFLCT, 0017
+ NEWCON, NTAPE4, NEWTCD, NEWTNL, NEWMFS, NEWMFP, 0017
+ MINTMP, MAXTMP, ATCODE, R41DEP, GO, REPORT(70) 0017
COMMON/CONFIG/ SURFSC(300), SSURFS(300), PNTSC(50), NEWPL(50) 0017
+ ONTIME(50), RECEVR(25), ICCODE, FOVANG(25), 0017
+ SERIES, NEWDAT, ADSURF, NNEWPL, 0017
+ JTOTAL, KTOTAL, NORBPL, ISURF(300), 0017
+ ISSURF(300) 0017
COMMON/CHANGE/ CHNGES, CHNGEK, CHNGEP, CHNGPL 0017
COMMON /RATES/ RATE(25,10), TAU(25,10), TSTART(3), TSTOP(3), 0017
+ AGEORB, AGEPLD, MOUT1, MOUT2, MED1, MED2, 0018
+ M1, M2, AMBWT, AMBDIA, TSTARR, MACH, 0018
+ TIMEOO 0018
COMMON /ORBIT/ ALT, VA, PITCH, YAW, ROLL, SUNL, 0018
+ SUNM, SUNH 0018
COMMON /GEOM/ XORGIN, YORGIN, ZORGIN, X0(25), Y0(25), Z0(25), 0018
+ ALPHA(25), BETA(25), GAMMA(25) 0018
COMMON /INTEG/ THETA(25), PHI(25), THETA1(25), THETA2(25), 0018
+ DTHETA(25), PHI1(25), PHI2(25), DPHI(25), 0018
+ DOMEGA(25), DS(25), RMAX, NTHETA, 0018
+ NPHI 0019
COMMON /TEMPS/ TEMP(2000) 0019
COMMON/PLUMES/ PLUMEC(10,25), SPECMF(10,25), CXLOC(50), CYLOC(50), 0019
+ CZLOC(50), CTHETA(50), CPHI(50), 0019
+ CIDENT(50) 0019

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

COMMON/MOLEC/  MOLWT(10), DIA(10)                                0019
COMMON /SURFS/IDENT(300), AREA(300)                             0019
COMMON/CHAR/ITITLE(72), IRSPNS(80), NEWNAM(6),  LTYPE(25),     0019
+      KINDS(25), PLACE(30), SPECIE(10), SECT(300),           0019
+      MATRL(300), NAMEPL, CLOC(50), CTYPE(50),               0019
+      NPLUME(25)                                             0020
COMMON/CHFLAG/ CHATT, CHALT, CHVEL, CHSUN, CHDS,CHORIG,       0020
+      CHTIM, CHAGE, CHINDX, CHWT, CHDIA, CHLOC(25),         0020
+      CHVIEW(25), CHRATE(25,10), CHTAU(25,10),             0020
+      CHPLUM(10,25), CHMF(10,25)                             0020
COMMON/INDX/  INDXSP(25), INDXK(25), INDXP(30), INDXPL(25),   0020
+      INDXJT, INDXKT                                         0020
COMMON/NUCON/NUSECT(6,300), NUMATL(6,300), NULOC(6,50),      0020
+      NUTYPE(6,50), NUSPEC(6,10), NUKIND(6,25),            0020
+      NUPLAC(6,30), NUNPLM(6,25)                             0020
                                                                0021
REAL ONTIME, MACH, MOLWT                                       0021
INTEGER SURFSC, SSURFS, PNTSC, RECEVR, ITITLE, LTYPE, KINDS,  0021
+      PLACE, SPECIE, SECT, MATRL, ATCODE, SERIES, CIDENT,   0021
+      CLOC, CTYPE, CHNGES, CHNGEK, CHNGEP, CHNGPL           0021
LOGICAL ORBITR, PAYLOD, OUT, ED, LEAK, PLUME, MCD,            0021
+      DIRECT, RFA2, RFSS, REFLCT, NEWCON, NTAPE4, NEWTCD,   0021
+      NEWTNL, NEWMFS, NEWMFP, MINTMP, MAXTMP, REPORT, GO,    0021
+      NEWPL, SUNL, SUNM, SUNH, R41DEP, NEWDAT, ADSURF,      0021
+      CHATT, CHALT, CHVEL, CHSUN, CHDS, CHORIG, CHLOC,     0021
+      CHVIEW, CHTIM, CHRATE, CHTAU, CHAGE, CHPLUM, CHMF,   0022
+      CHINDX, CHWT, CHDIA, INDXSP, INDXK, INDXP, INDXPL    0022
===== 0022
CALL CLEAR                                                     0022
CALL HEADER(2)                                                0022
                                                                0022
DETERMINE IF THE OPBITER IS TO BE ACTIVATED                   0022
                                                                0022
100 WRITE(6,6010)                                             0023
6010 FORMAT(//1X,48HDO YOU WANT THE ORBITER CONFIGURATION    0023
      ACTIVATED?/)
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)      0023
ORBITR = .TRUE.                                              0023
IF(IRSPNS(IFIRST).EQ.1HN) ORBITR=.FALSE.                    0023
                                                                0023
SEE IF A PAYLOAD CONFIGURATION IS TO BE ADDED IN             0023
                                                                0023
CALL CLEAR                                                     0023
CALL HEADER(2)                                                0023
WRITE(6,6020)                                                 0024
6020 FORMAT(//1X,46HDO YOU WANT A PAYLOAD CONFIGURATION      0024
      ACTIVATED?/)
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)      0024
IF(IRSPNS(IFIRST).EQ.1HY) PAYLOD = .TRUE.                   0024
IF(PAYLOD) GO TO 110                                          0024
IF(ORBITR) GO TO 250                                          0024
CALL CLEAR                                                     0024
CALL HEADER(2)                                                0024
WRITE(6,6025)                                                 0024
6025 FORMAT(//1X,38HIS THIS TO BE A POINT SOURCE ONLY RUN?/) 0024
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)      0025
IF(IRSPNS(IFIRST).NE.1HY) GO TO 105                          0025
NEWCON = .TRUE.                                              0025
GO TO 50                                                       0025
105 WRITE(6,6030)                                             0025
6030 FORMAT(/1X,
+60H*** EITHER THE ORBITER AND/OR A PAYLOAD MUST BE SELECTED ***)) 0025

```

**ORIGINAL PAGE IS
OF POOR QUALITY**

GO TO 100	ORIGINAL PAGE IS	0025
110 CALL CLEAR	OF POOR QUALITY	0025
CALL HEADER(2)		0025
NAMEPL = 6H		0026
WRITE(6,6040)		0026
6040 FORMAT(//1X,58HWHICH OF THE FOLLOWING PAYLOAD CONFIGURATIONS DO YOU WISH: /		0026
+5X,7H1) LMOP/5X,7H2) SMTP/5X,7H3) FIVP/5X,8H4) P80-1/		0026
+5X,10H5) DSP/IUS/5X,20H6) NEW CONFIGURATION//)		0026
CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)		0026
IF((IVALUE.LE.0).OR.(IVALUE.GE.7)) GO TO 110		0026
IF(IVALUE.EQ.6) GO TO 120		0026
IF(IVALUE.NE.1) GO TO 112		0026
NAMEPL = 6H LMOP		0026
SERIES = 1000		0027
GO TO 250		0027
112 IF(IVALUE.NE.2) GO TO 114		0027
NAMEPL = 6H SMTP		0027
SERIES = 2000		0027
GO TO 250		0027
114 IF(IVALUE.NE.3) GO TO 116		0027
NAMEPL = 6H FIVP		0027
SERIES = 3000		0027
GO TO 250		0027
116 IF(IVALUE.NE.4) GO TO 118		0028
NAMEPL = 6H P801		0028
SERIES = 1000		0028
GO TO 250		0028
118 IF(IVALUE.NE.5) GO TO 120		0028
NAMEPL = 6HDSPIUS		0028
SERIES = 1000		0028
GO TO 250		0028
120 CALL CLEAR		0028
CALL HEADER(2)		0028
WRITE(6,6050)		0029
605) FORMAT(//1X,49HWHAT IS THE NAME OF THE PAYLOAD (6 LETTERS MAX)? /)		0029
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)		0029
NCHAR = ILAST - IFIRST + 1		0029
IF(NCHAR.GT.6) GO TO 120		0029
IPAD = 6 - NCHAR		0029
IF(IPAD.EQ.0) GO TO 140		0029
DO 130 I=1,IPAD		0029
130 NEWNAM(I) = 1H		0029
140 IPAD1 = IPAD + 1		0029
II = 0		0030
DO 150 I=IPAD1,6		0030
II = II + 1		0030
150 NEWNAM(I) = IRSPNS(IFIRST + II - 1)		0030
160 CALL CLEAR		0030
CALL HEADER(2)		0030
WRITE(6,6055)		0030
6055 FORMAT(//1X,43HIN WHAT RANGE ARE THE PAYLOAD NODE NUMBERS?//		0030
+5X,12H1) 1000-1999/5X,12H2) 2000-2999/5X,12H3) 3000-3999/		0030
+5X,12H4) 4000-4999//)		0030
CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)		0031
IF((IVALUE.LE.0).OR.(IVALUE.GE.5)) GO TO 160		0031
SERIES = IVALUE * 1000		0031
DETERMINE HOW NEW CONFIGURATION DATA WILL BE INPUT TO SPACE		0031
200 CALL CLEAR		0031
CALL HEADER(2)		0031

WRITE(6,6060)	0031
6060 FORMAT(//1X,40HHOW WILL PAYLOAD CONFIGURATION BE INPUT?//	0032
+5X,13H1) VIA TAPE 4/	0032
+5X,32H2) VIA USER INPUT (THIS SESSION//)	0032
CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0032
IF((IVALUE.LE.0).OR.(IVALUE.GE.3)) GO TO 200	0032
IF(IVALUE.EQ.1) NTAPE4 = .TRUE.	0032
IF(IVALUE.EQ.2) NEWCON = .TRUE.	0032
IF(IVALUE.EQ.2) ADSURF = .TRUE.	0032
	0032
SELECT SURFACE SOURCES DESIRED	0032
	0032
250 CALL CLEAR	0033
CALL HEADER(2)	0033
WRITE(6,6070)	0033
6070 FORMAT(//1X,36HARE SURFACE SOURCES TO BE ACTIVATED?//)	0033
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0033
OUT = .FALSE.	0033
IF(IRSPNS(IFIRST).EQ.1HN) GO TO 300	0033
255 IF(.NOT.ORBITR) GO TO 270	0033
260 CALL CLEAR	0033
CALL HEADER(2)	0033
WRITE(6,6080)	0034
6080 FORMAT(//1X,31HSELECT FROM THE FOLLOWING LIST://	0034
+5X,18H1) OUTGASSING ONLY/	0034
+5X,24H2) EARLY DESORPTION ONLY/	0034
+5X,21H3) CABIN LEAKAGE ONLY/	0034
+5X,32H4) OUTGASSING + EARLY DESORPTION/	0034
+5X,29H5) OUTGASSING + CABIN LEAKAGE/	0034
+5X,35H6) EARLY DESORPTION + CABIN LEAKAGE/	0034
+5X,48H7) OUTGASSING + EARLY DESORPTION + CABIN LEAKAGE//)	0034
CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0035
IF((IVALUE.LE.0).OR.(IVALUE.GE.8)) GO TO 255	0035
IF((IVALUE.EQ.1).OR.(IVALUE.EQ.4).OR.(IVALUE.EQ.5).OR.	0035
+(IVALUE.EQ.7))OUT = .TRUE.	0035
IF((IVALUE.EQ.2).OR.(IVALUE.EQ.4).OR.(IVALUE.EQ.6).OR.	0035
+(IVALUE.EQ.7))ED = .TRUE.	0035
IF((IVALUE.EQ.3).OR.(IVALUE.EQ.5).OR.(IVALUE.EQ.6).OR.	0035
+(IVALUE.EQ.7))LEAK = .TRUE.	0035
GO TO 300	0035
270 CALL CLEAR	0035
CALL HEADER(2)	0035
WRITE(6,6085)	0036
6085 FORMAT(//1X,31HSELECT FROM THE FOLLOWING LIST://	0036
+5X,18H1) OUTGASSING ONLY/	0036
+5X,24H2) EARLY DESORPTION ONLY/	0036
+5X,34H3) OUTGASSING AND EARLY DESORPTION//)	0036
CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0036
IF((IVALUE.LE.0).OR.(IVALUE.GE.4)) GO TO 255	0036
IF((IVALUE.EQ.1).OR.(IVALUE.EQ.3)) OUT = .TRUE.	0036
IF((IVALUE.EQ.2).OR.(IVALUE.EQ.3)) ED = .TRUE.	0036
	0036
SEE IF MULTIPLE REFLECTIONS ARE DESIRED	0037
	0037
300 CALL CLEAR	0037
CALL HEADER(2)	0037
WRITE(6,6090)	0037
6090 FORMAT(//1X,43HDO YOU WANT MULTIPLE REFLECTIONS EVALUATED?//)	0037
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0037
IF(IRSPNS(IFIRST).NE.1HY) GO TO 400	0037
REFLECT = .TRUE.	0037
310 WRITE(6,6100)	0037
	0038

ORIGINAL PAGE 19
OF POOR QUALITY


```

6100 FORMAT(//1X,33HHOW MANY REFLECTIONS ARE DESIRED?/)          0038
      CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)      0038
      IF((IVALUE.LE.0).OR.(IVALUE.GT.10)) GO TO 310                0038
      NRFLCT = IVALUE                                              0038

      SEE IF ORBITER POINT SOURCES (ENGINES/VENTS) ARE TO BE     0038
      ACTIVATED                                                    0038

400 IF(.NOT.ORBITR) GO TO 500                                     0038
      CALL CLEAR                                                    0038
      CALL HEADER(2)                                               0039
      WRITE(6,6110)                                                0039
6110 FORMAT(//1X,51HDO YOU WANT ANY ORBITER ENGINES OR VENTS    0039
      +/)                                                           0039
      CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)      0039
      IF(IRSPNS(IFIRST).NE.1HY) GO TO 500                          0039
      PLUME = .TRUE.                                               0039

      SEE IF NEW POINT SOURCES ARE TO BE INPUT VIA NEW           0039
      CONFIGURATION CARDS                                         0039

500 CALL CLEAR                                                    0040
      CALL HEADER(2)                                               0040
      WRITE(6,6130)                                                0040
6130 FORMAT(//1X,39HDO YOU WANT TO INPUT NEW POINT SOURCES?/)   0040
      CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)      0040
      IF(IRSPNS(IFIRST).NE.1HY) GO TO 600                          0040
      NEWCON = .TRUE.                                              0040
510 WRITE(6,6140)                                                 0040
6140 FORMAT(/1X,41HHOW MANY NEW POINT SOURCES WILL THERE BE?/)  0040
      CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)      0040
      IF(IVALUE.LE.0) GO TO 510                                     0041
      IF(IVALUE.GT.50) GO TO 510                                    0041
      NNEWPL = IVALUE                                              0041

      DETERMINE TYPE OF CONTAMINATION ANALYSIS DESIRED           0041

600 CALL CLEAR                                                    0041
      CALL HEADER(2)                                               0041
      WRITE(6,6150)                                                0041
6150 FORMAT(//1X,35HSELECT TYPE(S) OF ANALYSIS DESIRED://       0041
      +1X,18H(TYPE 0 WHEN DONE)//                                  0042
      +5X,50H1) MASS COLUMN DENSITY (MCD) ALONG A LINE OF SIGHT/  0042
      +5X,55H2) DIRECT FLUX AND/OR DEPOSITION ON A RECEIVING     0042
      SURFACE/                                                     0042
      +5X,51H3) RETURN FLUX/DEPOSITION DUE TO AMBIENT SCATTERING) 0042
      IF((.NOT.PLUME).OR.(NNEWPL.EQ.0)) GO TO 610                 0042
      WRITE(6,6160)                                                0042
6160 FORMAT(5X,48H4) RETURN FLUX/DEPOSITION DUE TO SELF          0042
      SCATTERING//)                                               0042
610 CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)      0042
      NHIVAL = 3                                                    0042
      IF(PLUME.OR.(NNEWPL.GT.0)) NHIVAL = 4                       0042
      IF((IVALUE.LT.0).OR.(IVALUE.GT.NHIVAL)) GO TO 600           0042
      IF(IVALUE.EQ.1) MCD = .TRUE.                                  0043
      IF(IVALUE.EQ.2) DIRECT = .TRUE.                              0043
      IF(IVALUE.EQ.3) RFAS2 = .TRUE.                               0043
      IF(IVALUE.EQ.4) RFSS = .TRUE.                               0043
      IF(IVALUE.NE.0) GO TO 610                                    0043

      NEED ADDITIONAL INPUT FOR BGK PLUME SELF SCATTERING MODEL 0043

      IF(.NOT.RFSS) GO TO 700                                       0043
650 CALL CLEAR                                                    0043
      CALL HEADER(2)                                               0044
      WRITE(6,6170)                                                0044

```

```

6170 FORMAT(//1X,48HPLEASE INPUT FOLLOWING POINT SOURCE PARAMETERS -// 0044
+5X,28HMACH NUMBER (MUST BE > 1.0)?//) 0044
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0044
IF(FVALUE.LT.1.) GO TO 650 0044
MACH = FVALUE 0044
660 WRITE(6,6180) 0044
6180 FORMAT(//5X,31HTSTARR (IN DEGREES CENTIGRADE)?//) 0044
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0045
IF(FVALUE.LE.-273.) GO TO 660 0045
TSTARR = FVALUE 0045

DETERMINE HOW TAPE 10 THERMAL DATA IS TO BE USED 0045

700 CALL CLEAR 0045
CALL HEADER(2) 0045
WRITE(6,6190) 0045
6190 FORMAT(//1X,45HWHICH OF THE FOLLOWING THERMAL PROFILES FROM / 0045
+1X,37HTAPE 10 SHOULD BE USED FOR THIS CASE:// 0046
+5X,20H1) COLUMN 1 (MAXTMP)/ 0046
+5X,20H2) COLUMN 2 (MINTMP)/ 0046
+5X,22H3) COLUMN 3 (ATCODE=1)/ 0046
+5X,22H3) COLUMN 4 (ATCODE=2)/ 0046
+5X,22H5) COLUMN 5 (ATCODE=3)/ 0046
+5X,22H6) COLUMN 6 (ATCODE=4)/ 0046
+5X,22H7) COLUMN 7 (ATCODE=5)//) 0046
CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0046
IF((IVALUE.LE.0).OR.(IVALUE.GE.8)) GO TO 700 0046
IF(IVALUE.GE.3) ATCODE = IVALUE - 2 0047
IF(IVALUE.EQ.1) MAXTMP = .TRUE. 0047
IF(IVALUE.EQ.2) MINTMP = .TRUE. 0047

DETERMINE IF THERMAL DATA ARE TO BE INPUT VIA NAMELIST OR CARDS 0047

750 CALL CLEAR 0047
CALL HEADER(2) 0047
WRITE(6,6200) 0047
6200 FORMAT(//1X,48HDO YOU WANT TO INPUT THERMAL DATA IN ADDITION TO/ 0047
+1X,26HTHAT CONTAINED ON TAPE 10?//) 0048
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0048
IF(IRSPNS(IFIRST).NE.1HY) GO TO 800 0048
760 WRITE(6,6210) 0048
6210 FORMAT(//1X,16HSHOULD INPUT BE:// 0048
+5X,31H1) VIA NAMELIST (NEWTNL OPTION)/ 0048
+5X,38H2) VIA FORMATTED CARDS (NEWTCD OPTION)//) 0048
CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0048
IF((IVALUE.LE.0).OR.(IVALUE.GE.3)) GO TO 760 0048
IF(IVALUE.EQ.1) NEWTNL = .TRUE. 0048
IF(IVALUE.EQ.2) NEWTCD = .TRUE. 0049

DETERMINE IF THE USERS CASE SHOULD RESULT IN THE GENERATION 0049
OF A POINT-BODY TAPE 13 VIEWFACTOR FILE (NEWMFP OPTION) 0049

800 IF(.NOT.PAYLOD) GO TO 850 0049
CALL CLEAR 0049
CALL HEADER(2) 0049
WRITE(6,6220) 0049
6220 FORMAT(//1X,47HDO YOU WANT THIS RUN TO GENERATE A TAPE 13 FROM/ 0049
+1X,40HTHE ORBITER TAPE 14 AND PAYLOAD TAPE 15?//) 0050
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0050
IF(IRSPNS(IFIRST).EQ.1HY) NEWMFP = .TRUE. 0050

BEGIN SELECTION PROCESS FOR SPAC 54 ITPUT REPORT OPTIONS 0050

```

ORIGINAL PAGE IS
OF POOR QUALITY


```

+          CHTIM, CHAGE, CHINDX,  CHWT,  CHDIA, CHLOC(25), 0056
+          CHVIEW(25), CHRATE(25,10),  CHTAU(25,10),      0056
+          CHPLUM(10,25), CHMF(10,25)                    0056
COMMON/INDX/  INDXSP(25), INDXK(25),  INDXP(30),  INDXPL(25), 0057
+          INDXJT,  INDXKT                                0057
COMMON/NUCON/NUSECT(6,300), NUMATL(6,300),  NULOC(6,50), 0057
+          NUTYPE(6,50),  NUSPEC(6,10),  NUKIND(6,25),    0057
+          NUPLAC(6,30),  NUNPLM(6,25)                   0057
+          REAL ONTIME, MACH, MOLWT                       0057
+          INTEGER SURFSC, SSURFS,  PNTSC, RECEVR, ITITLE,  LTYPE,  KINDS, 0057
+          PLACE, SPECIE,  SECT,  MATRL, ATCODE, SERIES,  CIDENT, 0057
+          CLOC,  CTYPE,  CHNGES,  CHNGEK,  CHNGEP,  CHNGPL, 0057
LOGICAL ORBITR, PAYLOD,  OUT,  ED,  LEAK,  PLUME,  MCD, 0058
+          DIRECT,  RFAS2,  RFSS,  REFLCT,  NEWCON,  NTAPE4,  NEWTCD, 0058
+          NEWTNL,  NEWMFS,  NEWMFP,  MINTMP,  MAXTMP,  REPORT,  GO, 0058
+          NEWPL,  SUNL,  SUNM,  SUNH,  R41DEP,  NEWDAT,  ADSURF, 0058
+          CHATT,  CHALT,  CHVEL,  CHSUN,  CHDS,  CHORIG,  CHLOC, 0058
+          CHVIEW,  CHTIM,  CHRATE,  CHTAU,  CHAGE,  CHPLUM,  CHMF, 0058
+          CHINDX,  CHWT,  CHDIA,  INDXSP,  INDXK,  INDXP,  INDXPL, 0058
=====0058
IF THE ORBITER AND/OR A PAYLOAD HAVE BEEN ACTIVATED, SEE IF THE USER 0059
DESIRES TO ZERO OUT ANY SURFACES.  IF MULTIPLE REFLECTIONS HAVE BEEN 0059
FLAGGED, SEE IF ANY NODES ARE TO BE ELIMINATED AS SECOND SURFACE SOUR0059
IF(.NOT.(ORBITR.OR.PAYLOD))GO TO 500 0059
CALL CLEAR 0059
CALL HEADER(3) 0059
WRITE(6,6010) 0059
6010 FORMAT(//1X,33HDO YOU WANT ANY SURFACES DELETED?/) 0059
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0059
IF(IRSPNS(IFIRST).NE.1HY) GO TO 250 0060
PROVIDE OPTION TO EITHER 0060
1) INPUT SEQUENCE NUMBERS DIRECTLY 0060
2) REVIEW ALL NODES, 1 SCREENFUL AT A TIME, THEN BRANCH TO 1) 0060
CALL CLEAR 0060
CALL HEADER(3) 0060
WRITE(6,6020) 0060
6020 FORMAT(//1X,46HYOU HAVE TWO OPTIONS FOR ZEROING OUT SURFACES:// 0060
+5X,56H1) IF YOU KNOW THE SEQUENCE NUMBER (NOT THE NODE NUMBER)// 0061
+8X,49HOF THE SURFACE(S) TO BE ELIMINATED, YOU CAN INPUT/ 0061
+8X,53HTHE NUMBER(S) DIRECTLY, TERMINATING THE LIST WITH "0"// 0061
+5X,52H2) IF YOU KNOW THE NODE NUMBER, BUT NOT THE SEQUENCE/ 0061
+8X,51HNUMBER, YOU CAN ELECT TO REVIEW ALL NODES AND THEIR/ 0061
+8X,48HSEQUENCE NUMBERS, 1 SCREENFUL AT A TIME.  IF YOU/ 0061
+8X,48HSELECT THIS OPTION, REVIEW THE LIST AND JOT DOWN/ 0061
+8X,48HTHE APPROPRIATE SEQUENCE NUMBERS.  AT THE END OF/ 0061
+8X,48HTHE REVIEW, THE PROGRAM WILL PROMPT YOU AS IN 1)////) 0061
110 WRITE(6,6030) 0061
6030 FORMAT(1X,36HPLEASE SELECT EITHER OPTION 1) OR 2)/) 0062
CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0062
IF((IVALUE.LE.0).OR.(IVALUE.GE.3)) GO TO 110 0062
IF(IVALUE.EQ.1) GO TO 200 0062
REVIEW CONFIGURATION 0062
CALL CLEAR 0062
WRITE(6,6040) 0062

```

```

6040 FORMAT(1X,51H* REVIEW CONFIGURATION - PRESS RETURN TO CONTINUE */ 0062
      +5X,5HSEQ #,5X,4HNODE,6X,5HPLACE,6X,4HMATL,7X,4HAREA/ 0063
      +5X,5H-----,5X,4H----,5X,6H-----,5X,6H-----,5X,7H-----) 0063
      NEXT = 1 0063
120 LCOUNT = 1 0063
130 WRITE(6,6050)NEXT,SURFSC(NEXT),SECT(NEXT),MATRL(NEXT), 0063
      +AREA(NEXT) 0063
6050 FORMAT(6X,I3,6X,I4,5X,A6,5X,A6,5X,F7.1) 0063
      NEXT = NEXT + 1 0063
      LCOUNT = LCOUNT + 1 0063
      IF(NEXT.GT.JTOTAL) GO TO 150 0064
      IF(LCOUNT.GT.20) GO TO 150 0064
      GO TO 130 0064
150 CALL KYBDIN(IRSPNS,5,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0064
      IF(NEXT.GT.JTOTAL) GO TO 200 0064
      WRITE(6,6040) 0064
      GO TO 120 0064

      PROMPT FOR SURFACES TO BE ZEROED 0064

200 CALL CLEAR 0065
      CALL HEADER(3) 0065
      WRITE(6,6055) 0065
6055 FORMAT(//1X,39HINPUT SEQUENCE NUMBER(S) OF NODES TO BE/ 0065
      +29HELIMINATED (TYPE 0 WHEN DONE)//) 0065
      N = 1 0065
210 CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0065
      IF((IVALUE.LT.0).OR.(IVALUE.GT.300)) CALL IERROR 0065
      IF(IVALUE.EQ.0) GO TO 250 0065
      ISURF(N) = IVALUE 0065
      N = N + 1 0066
      GO TO 210 0066

      IF MULTIPLE REFLECTIONS HAVE BEEN FLAGGED, USE SAME APPROACH AS 0066
      ABOVE FOR ZEROING OUT ANY SURFACES NOT DESIRED AS REFLECTORS 0066

250 IF(.NOT.REFLCT) GO TO 500 0066
      CALL CLEAR 0066
      CALL HEADER(3) 0066
      WRITE(6,6060) 0066
6060 FORMAT(//1X,41HDO YOU WANT ANY SURFACES TO BE DELETED AS/ 0067
      +1X,23HSECOND SURFACE SOURCES?/) 0067
      CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0067
      IF(IRSPNS(IFIRST).NE.1HY) GO TO 500 0067

      PROVIDE OPTION TO EITHER 0067
      1) INPUT SEQUENCE NUMBERS DIRECTLY 0067
      2) REVIEW ALL NODES, 1 SCREENFUL AT A TIME, THEN BRANCH TO 1) 0067

      CALL CLEAR 0067
      CALL HEADER(3) 0068
      WRITE(6,6020) 0068
310 WRITE(6,6030) 0068
      CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0068
      IF((IVALUE.LE.0).OR.(IVALUE.GE.3)) GO TO 310 0068
      IF(IVALUE.EQ.1) GO TO 400 0068

      REVIEW CONFIGURATION 0068

      CALL CLEAR 0068
      WRITE(6,6040) 0069

```

```

NEXT = 1
320 LCOUNT = 1
330 WRITE(6,6050)NEXT,SURFSC(NEXT),SECT(NEXT),MATRL(NEXT),
+AREA(NEXT)
NEXT = NEXT + 1
LCOUNT = LCOUNT + 1
IF(NEXT.GT.JTCTAL) GO TO 350
IF(LCOUNT.GT.20) GO TO 350
GO TO 330
350 CALL KYBDIN(IRSPNS,5,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)
IF(NEXT.GT.JTOTAL) GO TO 400
WRITE(6,6040)
GO TO 320

PROMPT FOR SURFACES TO BE ZEROED

400 CALL CLEAR
CALL HEADER(3)
WRITE(6,6070)
6070 FORMAT(//1X,39HINPUT SEQUENCE NUMBER(S) OF NODES TO BE/
+29HELIMINATED (TYPE 0 WHEN DONE)//)
N = 1
410 CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)
IF((IVALUE.LT.0).OR.(IVALUE.GT.300)) CALL IERROR
IF(IVALUE.EQ.0) GO TO 500
ISSURF(N) = IVALUE
N = N + 1
GO TO 410

IF THE USER WANTS ORBITER OR OTHER (NEW) PLUMES ACTIVATED,
PROMPT FOR THEIR NODE NUMBERS AND ONTIMES. PROVIDE OPTION
TO REVIEW ORBITER POINT SOURCES.

500 IF((.NOT.PLUME).AND.(NNEWPL.EQ.0)) GO TO 800
N = 1
IF(.NOT.PLUME) GO TO 700
CALL CLEAR
CALL HEADER(3)
WRITE(6,6075)
6075 FORMAT(/9X,43H*** SELECT ACTIVE ORBITER POINT SOURCES ***/)
WRITE(6,6080)
6080 FORMAT(//1X,49HYOU HAVE TWO OPTIONS FOR ACTIVATING ORBITER POINT/
+1X,8HSOURCES://5X,44H1) IF YOU KNOW THE NODE NUMBER OF THE ENGINE/
+8X,37HOR VENT YOU WANT TO ACTIVATE, YOU CAN/
+8X,36HSIMPLY INPUT THE NODE NUMBER AND THE/
+8X,28HDESIRED ON TIME (IN SECONDS)//
+5X,51H2) IF YOU DON'T KNOW THE NODE NUMBER, YOU CAN ELECT/
+8X,47HTO REVIEW ALL ORBITER PREDEFINED POINT SOURCES,/
+8X,42HJOT DOWN THE DESIRED NODE NUMBERS AND THEN/
+8X,18HBRANCH TO OPTION 1///)
510 WRITE(6,6030)
CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)
IF((IVALUE.LE.0).OR.(IVALUE.GE.3)) GO TO 510
IF(IVALUE.EQ.1) GO TO 600

REVIEW PREDEFINED ORBITER ENGINES AND VENTS

CALL CLEAR
WRITE(6,6090)
6090 FORMAT(1X,51H* REVIEW ENGINES/VENTS - PRESS RETURN TO CONTINUE */
+2X,4HNODE,4X,3HLOC,3X,4HTYPE,7X,1HX,9X,1HY,8X,1HZ,6X,5HTHETA,

```

+6X,3HPhi/2X,4H----,2X,6H-----,2X,6H-----,3X,6H-----,3X,
+6H-----,3X,6H-----,3X,6H-----,3X,6H-----)

```

NEXT = 1
520 LCOUNT = 1
530 WRITE(6,6100)CIDENT(NEXT),CLOC(NEXT),CTYPE(NEXT),CXLOC(NEXT),
+
+ CYLOC(NEXT),CZLOC(NEXT),CTHETA(NEXT),CPHI(NEXT)
6100 FORMAT(2X,I4,2X,A6,2X,A6,5(2X,F7.1))
NEXT = NEXT + 1
LCOUNT = LCOUNT + 1
IF(NEXT.GT.KTOTAL) GO TO 550
IF(LCOUNT.GT.20) GO TO 550
GO TO 530
550 CALL KYBDIN(IRSPNS,5,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)
IF(NEXT.GT.KTOTAL) GO TO 600
WRITE(6,6090)
GO TO 520

PROMPT FOR ORBITER ENGINES/VENTS TO BE ACTIVATED

600 CALL CLEAR
CALL HEADER(3)
610 WRITE(6,6110)
6110 FORMAT(/1X,47HINPUT NODE NUMBER OF ORBITER ENGINE OR VENT YOU/
+1X,37HWANT ACTIVATED (ENTER 0 TO TERMINATE)/)
CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)
IF(IVALUE.EQ.0) GO TO 700
IF((IVALUE.LT.5000).OR.(IVALUE.GT.9999)) GO TO 610
PNTSC(N) = IVALUE
620 WRITE(6,6120)
6120 FORMAT(/1X,39HINPUT POINT SOURCE ON TIME (IN SECONDS)/)
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)
IF(FVALUE.LE.0.) GO TO 620
ONTIME(N) = FVALUE
N = N + 1
GO TO 610

PROMPT FOR NEW PLUME INPUTS, IF REQUESTED BACK IN CONTRL

700 NORBPL = N - 1
IF(NNEWPL.EQ.0) GO TO 800
CALL CLEAR
CALL HEADER(3)
WRITE(6,6130)
6130 FORMAT(/1X,25HNEW POINT SOURCE INPUTS -/)
710 WRITE(6,6140)
6140 FORMAT(/1X,43HINPUT NODE NUMBER OF NEW ENGINE OR VENT YOU/
+1X,37HWANT ACTIVATED (ENTER 0 TO TERMINATE)/)
CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)
IF(IVALUE.EQ.0) GO TO 800
IF((IVALUE.LT.5000).OR.(IVALUE.GT.9999)) GO TO 710
PNTSC(N) = IVALUE
720 WRITE(6,6150)
6150 FORMAT(/1X,39HINPUT POINT SOURCE ON TIME (IN SECONDS)/)
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)
IF(FVALUE.LE.0) GO TO 720
ONTIME(N) = FVALUE
N = N + 1
IF(N.GT.(NORBPL + NNEWPL)) GO TO 800
GO TO 710

DETERMINE RECEIVING SURFACES DESIRED FOR ANALYSIS. IF DIRECT
```

ORIGINAL PAGE IS
OF POOR QUALITY

```

FLUX ANALYSIS WAS REQUESTED, SEE IF USER WANTS TO LIMIT FIELD OF VIEW OF RECEIVERS.                                0081
                                                                                               0081
800 KTOTAL = N - 1                                       0081
    CALL CLEAR                                           0081
    CALL HEADER(3)                                       0082
    N = 1                                               0082
    WRITE(6,6160)                                       0082
6160 FORMAT(//1X,47HPLEASE ENTER RECEIVING SURFACE NODE NUMBERS (UP/ 0082
+1X,27HTO 25). ENTER 0 WHEN DONE://)                   0082
810 WRITE(6,6170)                                       0082
6170 FORMAT(5X,6HNODE? )                               0082
820 CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0082
    IF((IVALUE.LT.0).OR.(IVALUE.GE.5000)) GO TO 820    0082
    IF(IVALUE.EQ.0) GO TO 900                          0082
    RECEVR(N) = IVALUE                                 0083
    N = N + 1                                          0083
    GO TO 810                                           0083
                                                                                               0083
900 IF(.NOT.DIRECT) GO TO 999                          0083
    CALL CLEAR                                           0083
    CALL HEADER(3)                                       0083
    WRITE(6,6180)                                       0083
6180 FORMAT(//1X,43HDO YOU WANT THE FIELD OF VIEW OF ANY OF THE/ 0083
+1X,55HRECEIVERS LIMITED TO LESS THAN 90 DEGREES (HALF ANGLE)?/) 0083
    CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0084
    IF(IRSPNS(IFIRST).NE.1HY) GO TO 999                0084
910 WRITE(6,6190)                                       0084
6190 FORMAT(/1X,37HINPUT RECEIVER NODE NUMBER (0 TO END)//) 0084
920 CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0084
    IF(IVALUE.EQ.0) GO TO 999                          0084
    IF((IVALUE.LE.0).OR.(IVALUE.GE.5000)) GO TO 920   0084
    DO 930 J=1,25                                       0084
    IF(IVALUE.EQ.RECEVR(J)) GO TO 940                 0084
930 CONTINUE                                           0084
    WRITE(6,6200)                                       0085
6200 FORMAT(1X,45H***** THAT NODE DOES NOT CORRESPOND TO ANY OF/ 0085
+ 7X,28HTHE RECEIVERS YOU IDENTIFIED)                 0085
    GO TO 910                                           0085
940 WRITE(6,6210)                                       0085
6210 FORMAT(1X,37HINPUT DESIRED FOV LIMITING HALF ANGLE) 0085
950 CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0085
    IF((FVALUE.LE.0.).OR.(FVALUE.GT.180.)) GO TO 950  0085
    FOVANG(J) = FVALUE                                 0085
    GO TO 910                                           0085
                                                                                               0086
***** ALL DONE WITH NAMELIST $INPUT DATA COLLECTIONS ***** 0086
                                                                                               0086
999 RETURN                                             0086
    END                                               0086
    SUBROUTINE INPUTB                                  0086
                                                                                               0086
*****                                                                                               0086
*                                                                                               0086
*   C O L L E C T   D A T A   F O R   I N P U T B   *   0086
*                                                                                               0087
*****                                                                                               0087
                                                                                               0087
RETURN                                               0087
END                                                 0087
SUBROUTINE MPDB                                       0087

```



```

*****
* COLLECT DATA FOR MPDB * ORIGINAL PAGE IS OF POOR QUALITY
*****

```

```

=====
COMMON /CNTRL/ ORBITR, PAYLOD, OUT, ED, LEAK, PLUME,
+ MCD, DIRECT, RFAS2, RFSS, REFLCT, NRFLCT,
+ NEWCON, NTAPE4, NEWTCD, NEWTNL, NEWMFS, NEWMFP,
+ MINTMP, MAXTMP, ATCODE, R41DEP, GO, REPORT(70)
COMMON/CONFIG/ SURFSC(300), SSURFS(300), PNTSC(50), NEWPL(50),
+ ONTIME(50), RECEVR(25), ICCODE, FOVANG(25),
+ SERIES, NEWDAT, ADSURF, NNEWPL,
+ JTOTAL, KTOTAL, NORBPL, ISURF(300),
+ ISSURF(300)
COMMON/CHANGE/ CHNGES, CHNGEK, CHNGEP, CHNGPL
COMMON /RATES/ RATE(25,10), TAU(25,10), TSTART(3), TSTOP(3),
+ AGEORB, AGEPLD, MOUT1, MOUT2, MED1, MED2,
+ M1, M2, AMBWT, AMBDIA, TSTARR, MACH,
+ TIMEOO
COMMON /ORBIT/ ALT, VA, PITCH, YAW, ROLL, SUNL,
+ SUNM, SUNH
COMMON /GEOM/ XORGIN, YORGIN, ZORGIN, X0(25), Y0(25), Z0(25),
+ ALPHA(25), BETA(25), GAMMA(25)
COMMON /INTEG/ THETA(25), PHIL(25), THETA1(25), THETA2(25),
+ DTHETA(25), PHI1(25), PHI2(25), DPHI(25),
+ OMEGA(25), DS(25), RMAX, NTHETA,
+ NPHI
COMMON /TEMPS/ TEMP(2000)
COMMON/PLUMES/ PLUMEC(10,25), SPECMF(10,25), CXLOC(50), CYLOC(50),
+ CZLOC(50), CTHETA(50), CPHI(50),
+ CIDENT(50)
COMMON/MOLEC/ MOLWT(10), DIA(10)
COMMON /SURFS/IDENT(300), AREA(300)
COMMON/CHAR/ITITLE(72), IRSPNS(80), NEWNAM(6), LTYPE(25),
+ KINDS(25), PLACE(30), SPECIE(10), SECT(300),
+ MATRL(300), NAMEPL, CLOC(50), CTYPE(50),
+ NPLUME(25)
COMMON/CHFLAG/ CHATT, CHALT, CHVEL, CHSUN, CHDS, CHORIG,
+ CHTIM, CHAGE, CHINDX, CHWT, CHDIA, CHLOC(25),
+ CHVIEW(25), CHRATE(25,10), CHTAU(25,10),
+ CHPLUM(10,25), CHMF(10,25)
COMMON/INDX/ INDXSP(25), INDXK(25), INDXP(30), INDXPL(25),
+ INDXJT, INDXKT
COMMON/NUCON/NUSECT(6,300), NUMATL(6,300), NULOC(6,50),
+ NUTYPE(6,50), NUSPEC(6,10), NUKIND(6,25),
+ NUPLAC(6,30), NUNPLM(6,25)

REAL ONTIME, MACH, MOLWT
INTEGER SURFSC, SSURFS, PNTSC, RECEVR, ITITLE, LTYPE, KINDS,
+ PLACE, SPECIE, SECT, MATRL, ATCODE, SERIES, CIDENT,
+ CLOC, CTYPE, CHNGES, CHNGEK, CHNGEP, CHNGPL
LOGICAL ORBITR, PAYLOD, OUT, ED, LEAK, PLUME, MCD,
+ DIRECT, RFAS2, RFSS, REFLCT, NEWCON, NTAPE4, NEWTCD,
+ NEWTNL, NEWMFS, NEWMFP, MINTMP, MAXTMP, REPORT, GO,
+ NEWPL, SUNL, SUNM, SUNH, R41DEP, NEWDAT, ADSURF,
+ CHATT, CHALT, CHVEL, CHSUN, CHDS, CHORIG, CHLOC,
+ CHVIEW, CHTIM, CHRATE, CHTAU, CHAGE, CHPLUM, CHMF,
+ CHINDX, CHWT, CHDIA, INDXSP, INDXK, INDXP, INDXPL

```

```

=====0093
SEE IF THE USER WANTS TO CHANGE THE DEFAULT ORBITER ALTITUDE, VELOCIT0094
OR ATTITUDE. IF SO, SET CHANGE FLAGS AND PROMPT FOR INPUTS. 0094
10 CALL CLEAR 0094
CALL HEADER(7) 0094
WRITE(6,6010)PITCH,YAW,ROLL 0094
6010 FORMAT(/1X,51HDO YOU WANT TO CHANGE THE ORBITER DEFAULT ATTITUDE? 0094
+//2X,9H(PITCH = ,F6.2,8H, YAW = ,F6.2,9H, ROLL = ,F6.2,1H)//) 0094
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0094
IF(IRSPNS(IFIRST).NE.1HY) GO TO 50 0095
CHATT = .TRUE. 0095
WRITE(6,6020) 0095
6020 FORMAT(/1X,30HENTER PITCH ANGLE IN DEGREES -/25X) 0095
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0095
PITCH = FVALUE 0095
WRITE(6,6030) 0095
6030 FORMAT(/1X,17HENTER YAW ANGLE -/25X) 0095
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0095
YAW = FVALUE 0095
WRITE(6,6040) 0096
6040 FORMAT(/1X,18HENTER ROLL ANGLE -/25X) 0096
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0096
ROLL = FVALUE 0096
50 CALL CLEAR 0096
CALL HEADER(7) 0096
WRITE(6,6050)ALT 0096
6050 FORMAT(/1X,51HDO YOU WANT TO CHANGE THE ORBITER DEFAULT ALTITUDE? 0096
+//2X,7H(ALT = ,F6.2,4H KM)) 0096
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0097
IF(IRSPNS(IFIRST).NE.1HY) GO TO 100 0097
CHALT = .TRUE. 0097
60 WRITE(6,6060) 0097
6060 FORMAT(/1X,30HENTER DESIRED ALTITUDE (IN KM)/25X) 0097
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0097
IF(FVALUE.LE.0.) GO TO 60 0097
ALT = FVALUE 0097
100 CALL CLEAR 0097
CALL HEADER(7) 0098
WRITE(6,6070)VA 0098
6070 FORMAT(/1X,51HDO YOU WANT TO CHANGE THE ORBITER DEFAULT VELOCITY? 0098
+//2X,7H(VEL = ,F6.0,7H M/SEC)) 0098
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0098
IF(IRSPNS(IFIRST).NE.1HY) GO TO 150 0098
CHVEL = .TRUE. 0098
110 WRITE(6,6080) 0098
6080 FORMAT(/1X,33HENTER DESIRED VELOCITY (IN M/SEC)/25X) 0098
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0098
IF(FVALUE.LE.0.) GO TO 110 0099
VA = FVALUE 0099
SEE WHICH ATMOSPHERIC DENSITY MODEL (SUNSPOT ACTIVITY) IS DESIRED 0099
150 CALL CLEAR 0099
CALL HEADER(7) 0099
160 WRITE(6,6090) 0099
6090 FORMAT(/1X,34HSELECT DESIRED ATMOSPHERE DENSITY:// 0099
+5X,6H1) LOW/5X,9H2) MEDIUM/5X,7H3) HIGH//25X) 0099
CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST) 0100

```

```

IF((IVALUE.LE.0).OR.(IVALUE.GE.4)) GO TO 160                                0100
IF(IVALUE.EQ.1) SUNL=.TRUE.                                                0100
IF(IVALUE.EQ.2) SUNM=.TRUE.                                                0100
IF(IVALUE.EQ.3) SUNH=.TRUE.                                                0100
IF((IVALUE.EQ.1).OR.(IVALUE.EQ.3)) CHSUN=.TRUE.                            0100
IF THE USER HAS REQUESTED AN MCD, RFAS2 OR RFSS ANALYSIS, PROMPT          0100
FOR LINE(S)-OF-SIGHT AND RECEIVER LOCATION(S)/ORIENTATION(S)                0100
200 IF(.NOT.(MCD.AND.RFAS2.AND.RFSS)) GO TO 800                            0100
    CALL CLEAR                                                                0101
    CALL HEADER(7)                                                            0101
    WRITE(6,6100)                                                             0101
6100 FORMAT(/1X,35H***** INPUT RECEIVER LOCATION *****//)              0101
    DO 250 N=1,25                                                            0101
        IF(RECEVR(N).EQ.0) GO TO 250                                         0101
        WRITE(6,6110)N,RECEVR(N),X0(N),Y0(N),Z0(N)                          0101
6110 FORMAT(/1X,20H*** RECEIVER NUMBER ,I2,8H - NODE ,I4//1X,             0101
+52HDO YOU WANT TO CHANGE THE RECEIVER DEFAULT LOCATION?/                   0101
+5X,5H(X = ,F6.2,6H, Y = ,F6.2,6H, Z = ,F6.2,1H)/25X)                    0102
        CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)              0102
        IF(IRSPNS(IFIRST).NE.1HY) GO TO 250                                  0102
        CHLOC(N) = .TRUE.                                                    0102
        WRITE(6,6115)                                                         0102
.6115 FORMAT(/1X,39HENTER RECEIVER X-COORDINATE (IN INCHES)/25X)           0102
        CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)              0102
        X0(N) = FVALUE                                                        0102
        WRITE(6,6120)                                                         0102
6120 FORMAT(/5X,27HENTER RECEIVER Y-COORDINATE/25X)                        0102
        CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)              0103
        Y0(N) = FVALUE                                                        0103
        WRITE(6,6130)                                                         0103
6130 FORMAT(/5X,27HENTER RECEIVER Z-COORDINATE/25X)                        0103
        CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)              0103
        Z0(N) = FVALUE                                                        0103
250 CONTINUE                                                                  0103
IF AN MCD ANALYSIS WAS REQUESTED, PROMPT FOR LINE-OF-SIGHT DEFINITION      0103
300 IF(.NOT.MCD) GO TO 500                                                  0104
    CALL CLEAR                                                                0104
    CALL HEADER(7)                                                            0104
    WRITE(6,6180)                                                             0104
6180 FORMAT(///1X,49H*** LINE-OF-SIGHT DEFINITION FOR MCD ANALYSIS ***     0104
+//1X,56H(ONE MCD LINE-OF-SIGHT MAY BE INPUT FOR EACH RECEIV_LR -)         0104
+//)                                                                           0104
    DO 400 N=1,25                                                            0104
        IF(RECEVR(N).EQ.0) GO TO 400                                         0104
        WRITE(6,6190)N,RECEVR(N)                                             0104
6190 FORMAT(///1X,20H*** RECEIVER NUMBER ,I2,8H - NODE ,I4//1X,           0105
+26HINPUT ANGLE THETA (IN DEG)/25X)                                         0105
        CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)              0105
        THETA(N) = FVALUE                                                    0105
        WRITE(6,6200)                                                         0105
6200 FORMAT(/1X,24HINPUT ANGLE PHI (IN DEG)/25X)                            0105
        CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)              0105
        PHIL(N) = FVALUE                                                      0105
400 CONTINUE                                                                  0105
IF AN RFAS2 OR RFSS ANALYSIS WAS REQUESTED, PROMPT FOR LOS DEFINITION      0106
500 IF(.NOT.(RFAS2.AND.RFSS)) GO TO 800                                    0106

```

CALL CLEAR	0106
CALL HEADER(7)	0106
WRITE(6,6210)	0106
6210 FORMAT(/1X,46H*** LOS DEFINITION FOR RFAS2/RFSS ANALYSIS ***//)	0106
DO 550 N=1,25	0106
IF(RECEVR(N).EQ.0) GO TO 550	0106
WRITE(6,6220)N,RECEVR(N)	0106
6220 FORMAT(/1X,20H*** RECEIVER NUMBER ,I2,8H - NODE ,I4//)	0107
WRITE(6,6140)	0107
6140 FORMAT(/1X,51HDO YOU WANT TO CHANGE RECEIVER DEFAULT ORIENTATION?	0107
+ /1X,47H(DEFAULT IS WITH RECEIVER NORMAL ALONG +Z AXIS)//25X)	0107
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0107
IF(IRSPNS(IFIRST).NE.1HY) GO TO 520	0107
CHVIEW(N) = .TRUE.	0107
WRITE(6,6150)	0107
6150 FORMAT(/1X,41HINPUT 1ST ROTATION (ALPHA) - CCW ABOUT +Z/25X)	0107
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0107
ALPHA(N) = FVALUE	0108
WRITE(6,6160)	0108
6160 FORMAT(/1X,44HINPUT 2ND ROTATION (BETA) - CCW ABOUT NEW +X/25X)	0108
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0108
BETA(N) = FVALUE	0108
WRITE(6,6170)	0108
6170 FORMAT(/1X,45HINPUT 3RD ROTATION (GAMMA) - CCW ABOUT NEW +Z/25X)	0108
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0108
GAMMA(N) = FVALUE	0108
520 WRITE(6,6225)	0108
6225 FORMAT(/1X,24HINPUT ANGLE THETA1 (DEG)//)	0109
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0109
THETA1(N) = FVALUE	0109
WRITE(6,6230)	0109
6230 FORMAT(/1X,18HINPUT ANGLE THETA2//)	0109
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0109
THETA2(N) = FVALUE	0109
WRITE(6,6240)	0109
6240 FORMAT(/1X,22HINPUT INCREMENT DTHETA//)	0109
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0109
DTHETA(N) = FVALUE	0110
WRITE(6,6250)	0110
6250 FORMAT(/1X,16HINPUT ANGLE PHI1//)	0110
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0110
PHI1(N) = FVALUE	0110
WRITE(6,6260)	0110
6260 FORMAT(/1X,16HINPUT ANGLE PHI2//)	0110
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0110
PHI2(N) = FVALUE	0110
WRITE(6,6270)	0110
6270 FORMAT(/1X,20HINPUT INCREMENT DPHI//)	0111
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0111
DPHI(N) = FVALUE	0111
550 CONTINUE	0111
	0111
	0111
	0111
BASICALLY DONE - COULD ADD AT A FUTURE TIME THE OPTION TO MODIFY	0111
THE DS(I) ARRAY, RMAX, DOMEGA(I), RFSTK, ETC. - IT WAS FELT THAT	0111
THESE OPTIONS ARE SO SELDOM USED THAT IMPLEMENTATION WAS NOT	0111
WARRANTED AT THIS TIME.	0112
	0112
	0112
800 CONTINUE	0112
RETURN	0112

```

END 0112
SUBROUTINE ADDCON 0112
0112
***** 0112
* 0112
* COLLECT NEW CONFIGURATION DATA * 0113
* 0113
***** 0113
===== 0113
COMMON /CNTRL/ ORBITR, PAYLOD, OUT, ED, LEAK, PLUME, 0113
+ MCD, DIRECT, RFAS2, RFSS, REFLCT, NRFLCT, 0113
+ NEWCON, NTAPE4, NEWTCD, NEWTNL, NEWMFS, NEWMFP, 0113
+ MINTMP, MAXTMP, ATCODE, R41DEP, GO, REPORT(70) 0113
COMMON/CONFIG/ SURFSC(300), SSURFS(300), PNTSC(50), NEWPL(50), 0114
+ ONTIME(50), RECEVR(25), ICCODE, FOVANG(25), 0114
+ SERIES, NEWDAT, ADSURF, NNEWPL, 0114
+ JTOTAL, KTOTAL, NORBPL, ISURF(300), 0114
+ ISSURF(300) 0114
COMMON/CHANGE/ CHNGES, CHNGEK, CHNGEP, CHNGPL 0114
COMMON /RATES/ RATE(25,10), TAU(25,10), TSTART(3), TSTOP(3), 0114
+ AGEORB, AGEPLD, MOUT1, MOUT2, MED1, MED2, 0114
+ M1, M2, AMBWT, AMBDIA, TSTARR, MACH, 0114
+ TIMEOO 0114
COMMON /ORBIT/ ALT, VA, PITCH, YAW, ROLL, SUNL, 0115
+ SUNM, SUNH 0115
COMMON /GEOM/ XORGIN, YORGIN, ZORGIN, XO(25), YO(25), ZO(25), 0115
+ ALPHA(25), BETA(25), GAMMA(25) 0115
COMMON /INTEG/ THETA(25), PHIL(25), THETA1(25), THETA2(25), 0115
+ DTHETA(25), PHI1(25), PHI2(25), DPHI(25), 0115
+ DOMEGA(25), DS(25), RMA, NTHETA, 0115
+ NPHI 0115
COMMON /TEMPS/ TEMP(2000) 0115
COMMON/PLUMES/ PLUMEC(10,25), SPECMF(10,25), CXLOC(50), CYLOC(50), 0115
+ CZLOC(50), CTHETA(50), CPHI(50), 0116
+ CIDENT(50) 0116
COMMON/MOLEC/ MOLWT(10), DIA(10) 0116
COMMON /SURFS/IDENT(300), AREA(300) 0116
COMMON/CHAR/ITITLE(72), IRSPNS(80), NEWNAM(6), LTYPE(25), 0116
+ KINDS(25), PLACE(30), SPECIE(10), SECT(300), 0116
+ MATRL(300), NAMEPL, CLOC(50), CTYPE(50), 0116
+ NPLUME(25) 0116
COMMON/CHFLAG/ CHATT, CHALT, CHVEL, CHSUN, CHDS, CHORIG, 0116
+ CHTIM, CHAGE, CHINDX, CHWT, CHDIA, CHLOC(25), 0116
+ CHVIEW(25), CHRATE(25,10), CHTAU(25,10), 0117
+ CHPLUM(10,25), CHMF(10,25) 0117
COMMON/INDX/ INDXSP(25), INDXK(25), INDXP(30), INDXPL(25), 0117
+ INDXJT, INDXKT 0117
COMMON/NUCON/NUSECT(6,300), NUMATL(6,300), NULCC(6,50), 0117
+ NUTYPE(6,50), NUSPEC(6,10), NUKIND(6,25), 0117
+ NUPLAC(6,30), NUNPLM(6,25) 0117
REAL ONTIME, MACH, MOLWT 0117
INTEGER SURFSC, SSURFS, PNTSC, RECEVR, ITITLE, LTYPE, KINDS, 0117
+ PLACE, SPECIE, SECT, MATRL, ATCODE, SERIES, CIDENT, 0118
+ CLOC, CTYPE, CHNGES, CHNGEK, CHNGEP, CHNGPL 0118
LOGICAL ORBITR, PAYLOD, OUT, ED, LEAK, PLUME, MCD, 0118
+ DIRECT, RFAS2, RFSS, REFLCT, NEWCON, NTAPE4, NEWTCD, 0118
+ NEWTNL, NEWMFS, NEWMFP, MINTMP, MAXTMP, REPORT, GO, 0118
+ NEWPL, SUNL, SUNM, SUNH, R41DEP, NEWDAT, ADSURF, 0118
+ CHATT, CHALT, CHVEL, CHSUN, CHDS, CHORIG, CHLOC, 0118

```

+	CHVIEW, CHTIM, CHRATE, CHTAU, CHAGE, CHPLUM, CHMF,	0118
+	CHINDX, CHWT, CHDIA, INDXSP, INDXK, INDXP, INDXP	0118
		0118

=====0119

SEE IF THE USER DESIRES TO INPUT A NEW SURFACE CONFIGURATION AS PART OF THE SPACE II INPUT DECK (NEWCON OPTION). IF SO, PROMPT FOR NODE NUMBER, LOCATION (PLACE), MATERIAL AND SURFACE AREA. THE PROGRAM AUTOMATICALLY DETERMINES THE SEQUENCE NUMBER FROM THE CURRENT VALUE OF JTOTAL, INCREMENTING JTOTAL AS SURFACES ARE ADDED.

```

IF(.NOT.ADSURF) GO TO 500
CALL CLEAR
CALL HEADER(4)
INDXJT = JTOTAL
WRITE(6,6010)
6010 FORMAT(/40H*** NEW SURFACE CONFIGURATION INPUTS ***/)
10 WRITE(6,6020)
6020 FORMAT(/32HINPUT NEW NODE NR. ("0" IF DONE)/)
CALL KYBDIN(IRSPNS,0,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)
IF(IVALUE.EQ.0) GO TO 500
IF((IVALUE.LT.0).OR.(IVALUE.GE.5000)) GO TO 10
JTOTAL = JTOTAL + 1
IDENT(JTOTAL) = IVALUE

20 WRITE(6,6030)
6030 FORMAT(/30HINPUT LOCATION (6 LETTERS MAX)/)
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)
NCHAR = ILAST - IFIRST + 1
IF(NCHAR.GT.6) GO TO 20
IPAD = 6 - NCHAR
IF(IPAD.EQ.0) GO TO 40
DO 30 I=1,IPAD
30 NUSECT(I,JTOTAL) = 1H
40 IPAD1 = IPAD + 1
II = 0
DO 50 I=IPAD1,6
II = II + 1
50 NUSECT(I,JTOTAL) = IRSPNS(IFIRST + II - 1)

120 WRITE(6,6040)
6040 FORMAT(/30HINPUT MATERIAL (6 LETTERS MAX)/)
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)
NCHAR = ILAST - IFIRST + 1
IF(NCHAR.GT.6) GO TO 120
IPAD = 6 - NCHAR
IF(IPAD.EQ.0) GO TO 140
DO 130 I=1,IPAD
130 NUMATL(I,JTOTAL) = 1H
140 IPAD1 = IPAD + 1
II = 0
DO 150 I=IPAD1,6
II = II + 1
150 NUMATL(I,JTOTAL) = IRSPNS(IFIRST + II - 1)

160 WRITE(6,6050)
6050 FORMAT(/28HINPUT SURFACE AREA (SQ. IN.)/)
CALL KYBDIN(IRSFNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)
IF(FVALUE.LE.0.) GO TO 160
AREA(JTOTAL) = FVALUE

```

GO TO 10	0124
INPUT NEW POINT SOURCE DATA	0125
500 IF(NNEWPL.EQ.0) GO TO 999	0125
CALL CLEAR	0125
CALL HEADER(4)	0125
INDXKT = KTOTAL - NNEWPL	0125
WRITE(6,6060)	0125
6060 FORMAT(/31H*** NEW POINT SOURCE INPUTS ***/)	0125
DO 800 N=1,NNEWPL	0126
K = NQRBPL + N	0126
WRITE(6,6070) N,PNTSC(K)	0126
6070 FORMAT(/24HNEW POINT SOURCE NUMBER ,I2,15H - NODE NUMBER ,I4,1H:)	0126
520 WRITE(6,6080)	0126
6080 FORMAT(/43HINPUT POINT SOURCE LOCATION (6 LETTERS MAX)/)	0126
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0126
NCHAR = ILAST - IFIRST + 1	0126
IF(NCHAR.GT.6) GO TO 520	0126
IPAD = 6 - NCHAR	0126
IF(IPAD.EQ.0) GO TO 540	0127
DO 530 I=1,IPAD	0127
530 NULOC(I,K) = 1H	0127
540 IPAD1 = IPAD + 1	0127
II = 0	0127
DO 550 I=IPAD1,6	0127
II = II + 1	0127
550 NULOC(I,K) = IRSPNS(IFIRST + II - 1)	0127
620 WRITE(6,6090)	0127
6090 FORMAT(/39HINPUT POINT SOURCE TYPE (6 LETTERS MAX)/)	0128
CALL KYBDIN(IRSPNS,3,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0128
NCHAR = ILAST - IFIRST + 1	0128
IF(NCHAR.GT.6) GO TO 620	0128
IPAD = 6 - NCHAR	0128
IF(IPAD.EQ.0) GO TO 640	0128
DO 630 I=1,IPAD	0128
630 NUTYPE(I,K) = 1H	0128
640 IPAD1 = IPAD + 1	0128
II = 0	0128
DO 650 I=IPAD1,6	0129
II = II + 1	0129
650 NUTYPE(I,K) = IRSPNS(IFIRST + II - 1)	0129
WRITE(6,6100)	0129
6100 FORMAT(/40HINPUT POINT SOURCE X-COORDINATE (INCHES)/)	0129
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0129
CXLOC(K) = FVALUE	0129
WRITE(6,6110)	0129
6110 FORMAT(/40HINPUT POINT SOURCE Y-COORDINATE (INCHES)/)	0129
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0130
CYLOC(K) = FVALUE	0130
WRITE(6,6120)	0130
6120 FORMAT(/40HINPUT POINT SOURCE Z-COORDINATE (INCHES)/)	0130
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0130
CZLOC(K) = FVALUE	0130
WRITE(6,6130)	0130
6130 FORMAT(/44HINPUT POINT SOURCE ORIENTATION ANGLE (THETA)/)	0130
CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)	0130
CTHETA(K) = FVALUE	0130
WRITE(6,6140)	0131

```

6140 FORMAT(/42HINPUT POINT SOURCE ORIENTATION ANGLE (PHI)/)          0131
      CALL KYBDIN(IRSPNS,1,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)          0131
      CPHI(K) = FVALUE                                                0131
800 CONTINUE                                                         0131
ALL DONE                                                             0131
999 RETURN                                                           0131
      END                                                            0132
      SUBROUTINE ADDATA                                              0132
*****                                                                0132
*                                                                    0132
*  P R O C E S S   C H A N G E   C A R D S   *                       0132
*                                                                    0132
*****                                                                0132
      RETURN                                                         0132
      END                                                            0133
      SUBROUTINE ADDTMP                                              0133
*****                                                                0133
*                                                                    0133
*  P R O C E S S   N E W   T E M P E R A T U R E   D A T A   *      0133
*                                                                    0133
*****                                                                0133
      RETURN                                                         0133
      END                                                            0134
      SUBROUTINE ADDVFS                                              0134
*****                                                                0134
*                                                                    0134
*  C O L L E C T   N E W   V F   D A T A   *                       0134
*                                                                    0134
*****                                                                0134
      RETURN                                                         0134
      END                                                            0135
      SUBROUTINE BUILD                                              0135
*****                                                                0135
*                                                                    0135
*  B U I L D   S P A C E   I I   I N P U T   D A T A   F I L E   *  0135
*                                                                    0135
*****                                                                0135
===== 0135
COMMON /CNTRL/ ORBITR, PAYLOD, OUT, ED, LEAK, PLUME,                0136
+             MCD, DIRECT, RFAS2, RFSS, REFLCT, NRFLCT,           0136
+             NEWCON, NTAPE4, NEWTCD, NEWTNL, NEWMFS, NEWMFP,      0136
+             MINTMP, MAXTMP, ATCODE, R41DEP, GO, REPORT(70)      0136
COMMON/CONFIG/ SURFSC(300), SSURFS(300), PNTSC(50), NEWPL(50),    0136
+             ONTIME(50), RECEVR(25), ICCODE, FOVANG(25),         0136
+             SERIES, NEWDAT, ADSURF, NNEWPL,                    0136
+             JTOTAL, KTOTAL, NORBPL, ISURF(300),                 0136
+             ISSURF(300)                                         0136
COMMON/CHANGE/ CHNGES, CHNGEK, CHNGEP, CHNGPL                     0137
COMMON /RATES/ RATE(25,10), TAU(25,10), TSTART(3), TSTOP(3),      0137
+             AGEORB, AGEPLD, MOUT1, WOUT2, MED1, MED2,           0137

```



```

+           M1,      M2,  AMBWT, AMBDIA, TSTARR,  MACH,      0137
+           TIMEOO
COMMON /ORBIT/  ALT,      VA,  PITCH,  YAW,  ROLL,  SUNL,      0137
+           SUNM,      SUNH
COMMON /GEOM/  XORGIN, YORGIN, ZORGIN, X0(25), Y0(25), Z0(25), 0137
+           ALPHA(25), BETA(25), GAMMA(25)
COMMON /INTEG/ THETA(25), PHIL(25), THETA1(25), THETA2(25), 0137
+           DTHETA(25), PHI1(25), PHI2(25), DPHI(25),
+           DOMEGA(25), DS(25), RMAX, NTHETA, 0138
+           NPHI
COMMON /TEMPS/ TEMP(2000) 0138
COMMON/PLUMES/ PLUMEC(10,25), SPECMF(10,25), CXLOC(50), CYLOC(50), 0138
+           CZLOC(50), CTHETA(50), CPHI(50),
+           CIDENT(50) 0138
COMMON/MOLEC/ MOLWT(10), DIA(10) 0138
COMMON /SURFS/IDENT(300), AREA(300) 0138
COMMON/CHAR/ITITLE(72), IRSPNS(80), NEWNAM(6), LTYPE(25), 0138
+           KINDS(25), PLACE(30), SPECIE(10), SECT(300),
+           MATRL(300), NAMEPL, CLOC(50), CTYPE(50), 0139
+           NPLUME(25) 0139
COMMON/CHFLAG/ CHATT, CHALT, CHVEL, CHSUN, CHDS, CHORIG, 0139
+           CHTIM, CHAGE, CHINDX, CHWT, CHDIA, CHLOC(25), 0139
+           CHVIEW(25), CHRATE(25,10), CHTAU(25,10), 0139
+           CHPLUM(10,25), CHMF(10,25) 0139
COMMON/INDX/  INDXSP(25), INDXK(25), INDXP(30), INDXPL(25), 0139
+           INDXJT, INDXKT 0139
COMMON/NUCON/NUSECT(6,300), NUMATL(6,300), NULOC(6,50), 0139
+           NUTYPE(6,50), NUSPEC(6,10), NUKIND(6,25), 0140
+           NUPLAC(6,30), NUNPLM(6,25) 0140
REAL ONTIME, MACH, MOLWT 0140
INTEGER SURFSC, SSURFS, PNTSC, RECEVR, ITITLE, LTYPE, KINDS, 0140
+           PLACE, SPECIE, SECT, MATRL, ATCODE, SERIES, CIDENT, 0140
+           CLOC, CTYPE, CHNGES, CHNGEK, CHNGEP, CHNGPL 0140
LOGICAL ORBITR, PAYLOD, OUT, ED, LEAK, PLUME, MCD, 0140
+           DIRECT, RFAS2, RFSS, REFLCT, NEWCON, NTAPE4, NEWTCD, 0140
+           NEWTNL, NEWMFS, NEWMFP, MINTMP, MAXTMP, REPORT, GO, 0140
+           NEWPL, SUNL, SUNM, SUNH, R41DEP, NEWDAT, ADSURF, 0141
+           CHATT, CHALT, CHVEL, CHSUN, CHDS, CHORIG, CHLOC, 0141
+           CHVIEW, CHTIM, CHRATE, CHTAU, CHAGE, CHPLUM, CHMF, 0141
+           CHINDX, CHWT, CHDIA, INDXSP, INDXK, INDXP, INDXPL 0141

```

```

=====0141
THE SPACE II INPUT DECK IS BUILT AND OUTPUT TO TAPE 1 0141
OUTPUT THE USER'S TITLE FOR THIS CASE 0141
REWIND 1 0142
WRITE(1,1010)(ITITLE(I),I=1,72) 0142
1010 FORMAT(72A1) 0142
GENERATE OUTPUT FOR NAMELIST $CONTRL 0142
(SHOW ALL VALUES, WHETHER .T. OR .F., DEFAULT OR NOT) 0142
WRITE(1,1020) 0142
1020 FORMAT(8H $CONTRL) 0143
IF(ORBITR) WRITE(1,1030) 0143
1030 FORMAT(16H ORBITR=.TRUE.,) 0143
IF(.NOT.ORBITR) WRITE(1,1040) 0143

```

1040	FORMAT(17H ORBITR=.FALSE.,)	0143
	IF(PAYLOD) WRITE(1,1050)	0143
1050	FORMAT(16H PAYLOD=.TRUE.,)	0143
	IF(.NOT.PAYLOD) WRITE(1,1060)	0143
1060	FORMAT(17H PAYLOD=.FALSE.,)	0144
	IF(OUT) WRITE(1,1070)	0144
1070	FORMAT(13H OUT=.TRUE.,)	0144
	IF(.NOT.OUT) WRITE(1,1080)	0144
1080	FORMAT(14H OUT=.FALSE.,)	0144
	IF(ED) WRITE(1,1090)	0144
1090	FORMAT(12H ED=.TRUE.,)	0144
	IF(.NOT.ED) WRITE(1,1100)	0144
1100	FORMAT(13H ED=.FALSE.,)	0145
	IF(LEAK) WRITE(1,1110)	0145
1110	FORMAT(14H LEAK=.TRUE.,)	0145
	IF(.NOT.LEAK) WRITE(1,1120)	0145
1120	FORMAT(15H LEAK=.FALSE.,)	0145
	IF(PLUME) WRITE(1,1130)	0145
1130	FORMAT(15H PLUME=.TRUE.,)	0145
	IF(.NOT.PLUME) WRITE(1,1140)	0145
1140	FORMAT(16H PLUME=.FALSE.,)	0146
	IF(MCD) WRITE(1,1150)	0146
1150	FORMAT(13H MCD=.TRUE.,)	0146
	IF(.NOT.MCD) WRITE(1,1160)	0146
1160	FORMAT(14H MCD=.FALSE.,)	0146
	IF(DIRECT) WRITE(1,1170)	0146
1170	FORMAT(16H DIRECT=.TRUE.,)	0146
	IF(.NOT.DIRECT) WRITE(1,1180)	0146
1180	FORMAT(17H DIRECT=.FALSE.,)	0147
	IF(RFAS2) WRITE(1,1190)	0147
1190	FORMAT(15H RFAS2=.TRUE.,)	0147
	IF(.NOT.RFAS2) WRITE(1,1200)	0147
1200	FORMAT(16H RFAS2=.FALSE.,)	0147
	IF(RFSS) WRITE(1,1210)	0147
1210	FORMAT(14H RFSS=.TRUE.,)	0147
	IF(.NOT.RFSS) WRITE(1,1220)	0147
1220	FORMAT(15H RFSS=.FALSE.,)	0148
	IF(REFLCT) WRITE(1,1222)	0148
1222	FORMAT(16H REFLCT=.TRUE.,)	0148
	IF(.NOT.REFLCT) WRITE(1,1224)	0148
1224	FORMAT(17H REFLCT=.FALSE.,)	0148
	IF(REFLCT) WRITE(1,1225) NRFLCT	0148
1225	FORMAT(9H NRFLCT=,I2,1H,)	0148
	IF(NEWCON) WRITE(1,1226)	0148
1226	FORMAT(16H NEWCON=.TRUE.,)	0149
	IF(.NOT.NEWCON) WRITE(1,1228)	0149
1228	FORMAT(17H NEWCON=.FALSE.,)	0149
	IF(NTAPE4) WRITE(1,1230)	0149
1230	FORMAT(16H NTAPE4=.TRUE.,)	0149
	IF(.NOT.NTAPE4) WRITE(1,1240)	0149

1240	FORMAT(17H NTAPE4=.FALSE.,)	0149
	IF(NEWTCD) WRITE(1,1250)	0149
1250	FORMAT(16H NEWTCD=.TRUE.,)	0149
	IF(.NOT.NEWTCD) WRITE(1,1260)	0150
1260	FORMAT(17H NEWTCD=.FALSE.,)	0150
	IF(NEWTNL) WRITE(1,1270)	0150
1270	FORMAT(16H NEWTNL=.TRUE.,)	0150
	IF(.NOT.NEWTNL) WRITE(1,1280)	0150
1280	FORMAT(17H NEWTNL=.FALSE.,)	0150
	IF(NEWMFS) WRITE(1,1290)	0150
1290	FORMAT(16H NEWMFS=.TRUE.,)	0151
	IF(.NOT.NEWMFS) WRITE(1,1300)	0151
1300	FORMAT(17H NEWMFS=.FALSE.,)	0151
	IF(NEWMFP) WRITE(1,1310)	0151
1310	FORMAT(16H NEWMFP=.TRUE.,)	0151
	IF(.NOT.NEWMFP) WRITE(1,1320)	0151
1320	FORMAT(17H NEWMFP=.FALSE.,)	0151
	IF(MINTMP) WRITE(1,1330)	0151
1330	FORMAT(16H MINTMP=.TRUE.,)	0152
	IF(MAXTMP) WRITE(1,1340)	0152
1340	FORMAT(16H MAXTMP=.TRUE.,)	0152
	IF(ATCODE.NE.0) WRITE(1,1350)ATCODE	0152
1350	FORMAT(9H ATCODE=,I1,1H,)	0152
	HANDLE REPORTS HERE	0152
	WRITE(1,1360)	0152
1360	FORMAT(37H REPORT(7)=.TRUE.,REPORT(51)=.TRUE.,)	0152
	IF (DIRECT) WRITE(1,1370)	0153
1370	FORMAT(22H REPORT(21)=4*.TRUE.,)	0153
	IF (MCD) WRITE(1,1380)	0153
1380	FORMAT(38H REPORT(35)=.TRUE.,REPORT(37)=.TRUE.,)	0153
	IF (RFSS .OR. RFAS2) WRITE(1,1390)	0153
1390	FORMAT(20H REPORT(45)=.TRUE.,)	0153
	WRITE(1,1500)	0153
1500	FORMAT(12H GO=.TRUE.,/5H \$END)	0153
	HANDLE PAYLOAD CARD, IF APPROPRIATE	0153
	IF(.NOT.PAYLOD) GO TO 100	0154
	IF(NAMEPL.NE.6H) WRITE(1,1510)NAMEPL,SERIES	0154
1510	FORMAT(A6,4X,I4)	0154
	IF(NAMEPL.EQ.6H) WRITE(1,1520)(NEWNAM(I),I=1,6),SERIES	0154
1520	FORMAT(6A1,4X,I4)	0154
	HANDLE OUTPUT FOR NAMELIST \$INPUTA (ONLY MODS OR SPECIFIC INPUTS)	0154
100	WRITE(1,1525)	0154
1525	FORMAT(8H \$INPUTA)	0155
	ZEROED SURFACES	0155
	DO 110 I=1,JTOTAL	0155
	IF(ISURF(I).NE.0) WRITE(1,1530) ISURF(I)	0155
1530	FORMAT(9H SURFSC(.I4,4H)=0,)	0155
110	CONTINUE	0155

ZEROED SECOND SURFACES	0155
DO 120 I=1, JTOTAL	0156
IF(ISSURF(I).NE.0) WRITE(1,1540) ISSURF(I)	0156
1540 FORMAT(9H SSURFS(,I4,4H)=0,)	0156
120 CONTINUE	0156
HANDLE PNTSC, ONTIME AND NEWPL INPUTS	0156
IF(.NOT.(PLUME.AND.(NNEWPL.GT.0))) GO TO 250	0156
DO 150 K=1,KTOTAL	0157
IF(PNTSC(K).NE.0) WRITE(1,1550) K,PNTSC(K)	0157
1550 FORMAT(8H PNTSC(,I2,2H)=,I4,1H,)	0157
150 CONTINUE	0157
DO 160 K=1,KTOTAL	0157
IF(PNTSC(K).NE.0) WRITE(1,1560) K,ONTIME(K)	0157
1560 FORMAT(9H ONTIME(,I2,2H)=,F6.2,1H,)	0157
160 CONTINUE	0157
200 IF(NNEWPL.EQ.0) GO TO 250	0158
KFIRST = NORBPL + 1	0158
KLAST = NORBPL + NNEWPL	0158
DO 210 K=KFIRST,KLAST	0158
WRITE(1,1565) K	0158
1565 FORMAT(8H NEWPL(,I2,9H)=.TRUE.,)	0158
210 CONTINUE	0158
HANDLE RECEVR, ICCODE AND FOVANG INPUTS	0158
250 NRECVR = 0	0159
DO 260 I=1,25	0159
IF(RECEVR(I).EQ.0) GO TO 260	0159
NRECVR = NRECVR + 1	0159
WRITE(1,1570) I,RECEVR(I)	0159
1570 FORMAT(9H RECEVR(,I2,2H)=,I4,1H,)	0159
260 CONTINUE	0159
IF(MCD.OR.DIRECT) ICCODE = 1	0159
IF(RFAS2.OR.RFSS) ICCODE = 2	0159
WRITE(1,1580) NRECVR,ICCODE	0160
1580 FORMAT(12H ICCODE(1)=,I2,1H*,I1,1H,)	0160
DO 270 I=1,25	0160
IF(FOVANG(I).NE.180.) WRITE(1,1590) I,FOVANG(I)	0160
1590 FORMAT(9H FOVANG(,I2,2H)=,F6.2,1H,)	0160
270 CONTINUE	0160
CLOSE OUT NAMELIST \$INPUTA	0160
WRITE(1,1600)	0161
1600 FORMAT(12H GO=.TRUE.,/5H \$END)	0161
PROCESS NEW CONFIGURATION INPUTS (SURFACES AND/OR POINT SOURCES)	0161
IF(.NOT.NEWCON) GO TO 400	0161
IF(.NOT.ADSURF) GO TO 300	0161
JP1 = INDXJT + 1	0161
DO 290 J=JP1, JTOTAL	0161
WRITE(1,1610) J, IDENT(J), (NUSECT(I,J), I=1,6), (NUMATL(I,J), I=1,6),	0161
+ AREA(J)	0162

1610	FORMAT(I5,I5,4X,6A1,4X,6A1,F10.1)	0162
290	CONTINUE	0162
300	IF(NNEWPL.EQ.0) GO TO 390	0162
	KP1 = INDXKT + 1	0162
	KNEXT = NORBPL	0162
	DO 350 K=KP1,KTOTAL	0162
	KNEXT = KNEXT + 1	0162
	WRITE(1,1620)KNEXT,PNTSC(KNEXT),(NULOC(I,K),I=1,6),	0162
	+ (NUTYPE(I,K),I=1,6),CXLOC(K),CYLOC(K),CZLOC(K),	0163
	+ CTHETA(K),CPHI(K)	0163
1620	FORMAT(I5,I5,4X,6A1,4X,6A1,5F10.1)	0163
350	CONTINUE	0163
390	WRITE(1,1630)	0163
1630	FORMAT(5H99999)	0163
	BUILD NAMELIST \$INPUTB	0163
400	CONTINUE	0163
	WRITE(1,1700)	0163
1700	FORMAT(8H \$INPUTB)	0164
	WRITE(1,1500)	0164
	HANDLE OUTPUT FOR NAMELIST \$MPDB (ONLY MODS OR SPECIFIC INPUTS)	0164
	WRITE(1,1800)	0164
1800	FORMAT(6H \$MPDB)	0164
	MISSION DATA BANK MODIFICATIONS	0164
	IF (CHATT) WRITE(1,1810) PITCH,YAW,ROLL	0165
1810	FORMAT(8H PITCH=,F6.2,1H,/6H YAW=,F6.2,1H,/7H ROLL=,	0165
	+F6.2,1H,)	0165
	IF (CHALT) WRITE(1,1820) ALT	0165
1820	FORMAT(6H ALT=,F6.2,1H,)	0165
	IF (CHVEL) WRITE(1,1830) VA	0165
1830	FORMAT(5H VA=,F6.2,1H,)	0165
	IF (.NOT. CHSUN) GO TO 500	0165
	IF (SUNL) WRITE(1,1835)	0165
1835	FORMAT(14H SUNL=.TRUE.,/15H SUNM=.FALSE.,)	0166
	IF (SUNH) WRITE(1,1840)	0166
1840	FORMAT(14H SUNH=.TRUE.,/15H SUNM=.FALSE.,)	0166
	MCD ANALYSIS DATA	0166
500	DO 510 I=1,25	0166
	IF (.NOT. CHLOC(I)) GO TO 510	0166
	WRITE(1,1865)I,X0(I),I,Y0(I),I,Z0(I)	0166
1865	FORMAT(5H X0(,I2,2H)=,F6.2,4H,Y0(,I2,2H)=,F6.2,	0166
	+4H,Z0(,I2,2H)=,F6.2,1H,)	0167
510	CONTINUE	0167
	IF (.NOT. MCD) GO TO 530	0167
	DO 520 I=1,25	0167
	IF (RECEVR(I) .EQ. 0) GO TO 520	0167
	WRITE(1,1880) I,THETA(I),I,PHIL(I)	0167
1880	FORMAT(9H THETA(,I2,2H)=,F6.2,6H,PHIL(,I2,2H)=,F6.2,1H,)	0167
520	CONTINUE	0167
530	IF (.NOT. (RFAS2 .AND. RFSS)) GO TO 600	0167
	DO 560 I=1,25	0168
	IF (.NOT. CHVIEW(I)) GO TO 540	0168
	73	0168

```

WRITE(1,1885) I,ALPHA(I),I,BETA(I),I,GAMMA(I)
1885 FORMAT(8H ALPHA(,I2,2H)=,F6.2,6H,BETA(,I2,2H)=,F6.2,
+7H,GAMMA(,I2,2H)=,F6.2,1H,)
540 IF(RECEVR(I).EQ.0) GO TO 560
WRITE(1,1890) I,THETA1(I),I,THETA2(I)
1890 FORMAT(9H THETA1(,I2,2H)=,F6.2,8H,THETA2(,I2,2H)=,F6.2,1H,)
WRITE(1,1895) I,PHI1(I),I,PHI2(I)
1895 FORMAT(7H PHI1(,I2,2H)=,F6.2,6H,PHI2(,I2,2H)=,F6.2,1H,)
WRITE(1,1900) I,DTHETA(I),I,DPHI(I)
1900 FORMAT(9H DTHETA(,I2,2H)=,F6.2,6H,DPHI(,I2,2H)=,F6.2)
560 CONTINUE

```

```

600 WRITE(1,1500)
WRITE(1,2000)
2000 FORMAT(4HSTOP)

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

RETURN
END
SUBROUTINE KYBDIN(IRSPNS,IDESIR,ITYPE,IVALUE,FVALUE,IFIRST,ILAST)

```

```

*****
*
*   K E Y B O A R D   I N P U T   B U F F E R   *
*
*****

```

- J. C. PIZZICAROLI
7/12/83

THIS SUBROUTINE PROVIDES THE CAPABILITY FOR AN INTERACTIVE FORTRAN PROGRAM TO CHECK THE STRUCTURE AND VALIDITY OF A USER'S KEYBOARD INPUT BEFORE PASSING THE RESPONSE ON TO THE MAIN PROGRAM. THIS PREVENTS A PROGRAM FROM BOMBING DUE TO A FORTRAN ERROR INDUCED BY TYPOGRAPHICAL ERRORS, VARIABLE TYPE MISMATCHES OR OTHERWISE INAPPROPRIATE USER RESPONSES.

THE APPROACH TAKEN IS TO INPUT ALL USER RESPONSES AS CHARACTER STRING. EACH CHARACTER IN THE RESPONSE STRING IS THEN DECODED, ONE AT A TIME, UNTIL THE STRING CONTENTS CAN BE INTERPRETED AS ONE OF THE FOLLOWING

```

ITYPE = 0 - INTEGER
1 - FLOATING POINT
2 - FLOATING POINT, EXPONENTIAL FORMAT
*** NOT CURRENTLY IMPLEMENTED ***
3 - STRING RESPONSE (I.E., YES, NO, ETC)
4 - BLANK
5 - NULL (CARRIAGE RETURN)
6 - UNRECOGNIZABLE (I.E., 1.2D, 4.4.4, ETC)

```

THE USER'S RESPONSE IS PASSED TO THE SUBROUTINE KYBD AS THE CHARACTER ARRAY IRSPNS(80). THE SUBROUTINE RETURNS THE FOLLOWING PARAMETERS -

```

IDESIR - TYPE OF RESPONSE DESIRED, PER ABOVE LIST
ITYPE - TYPE OF RESPONSE DECODED, PER ABOVE LIST
IVALUE - IF RESPONSE DECODED AS INTEGER, CONTAINS CORRESPONDING INTEGER VALUE
FVALUE - IF RESPONSE DECODED AS FLOATING POINT, CONTAINS CORRESPONDING FLOATING POINT VALUE
IFIRST - IF RESPONSE DECODED AS CHARACTER STRING, CONTAINS LOCATION OF FIRST NON-BLANK CHARACTER

```

ILAST - IF RESPONSE DECODED AS CHARACTER STRING, CON0174
LOCATION OF LAST NON-BLANK CHARACTER 0174

TO USE THIS SUBROUTINE, FOR EXAMPLE, THE FOLLOWING FORTRAN CODE - 0174

READ(5,5001) X 0175
5001 FORMAT(F10.4) 0175

WOULD BE REPLACED BY - 0175

CALL KYBD(IRSPNS,IDESIR,ITYPE,IVALUE,FVALUE,IFIRST, 0175

THE SUBROUTINE CHECKS TO SEE THAT THE TYPE OF RESPONSE MADE BY THE 0175
USER WAS APPROPRIATE. IF NOT, THE USER WILL BE PROMPTED FOR ANOTHER 0175
RESPONSE. 0175

KEY PROGRAM VARIABLES 0176

----- 0176

IDECPT - = 1 IF A DECIMAL POINT IS FOUND IN THE RESPONSE 0176
 = 0 OTHERWISE 0176
IDIG - = 1 IF AN INITIAL DIGIT IS FOUND IN THE RESPONSE 0176
 = 0 OTHERWISE 0176
IFDIG1 - LOCATION IN ARRAY IRSPNS(80) CORRESPONDING TO THE 0176
 LEFTMOST DIGIT IN A NUMERICAL RESPONSE 0177
IFDIG2 - LOCATION IN ARRAY IRSPNS(80) CORRESPONDING TO THE 0177
 LEFTMOST DIGIT TO THE RIGHT OF A DECIMAL POINT IN 0177
 A NUMERICAL RESPONSE 0177
ILDIG1 - LOCATION IN IRSPNS(80) CORRESPONDING TO THE RIGHTMO 0177
 DIGIT FOUND TO THE LEFT OF A DECIMAL POINT (IN A 0177
 F.P. NUMBER), OR, THE ABSOLUTE RIGHTMOST DIGIT IN 0177
 AN INTEGER RESPONSE 0177
ILDIG2 - LOCATION IN IRSPNS(80) CORRESPONDING TO THE RIGHTMO 0177
 DIGIT FOUND TO THE RIGHT OF A DECIMAL POINT 0177
LDECPT - LOCATION IN IRSPNS(80) OF THE DECIMAL POINT, IF ANO 0178
NDIG1 - NUMBER OF DIGITS FOUND TO THE LEFT OF A DECIMAL P 00178
 (IF ANY), OR, NUMBER OF DIGITS IN AN INTEGER RESP 00178
NDIG2 - NUMBER OF DIGITS FOUND TO THE RIGHT OF A DECIMAL P 0178
 (IF ANY) 0178

----- 0178
DIMENSION NUM(10), IRSPNS(80) 0178
DATA NUM/1H0, 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9/ 0178
DATA IPLUS, MINUS, IBLANK, IPOINT/1H+, 1H-, 1H , 1H./ 0179

READ IN USERS KEYBOARD RESPONSE 0179

1 ITYPE = 5 0179
DO 2 J=1,80 0179
2 IRSPNS(J) = IBLANK 0179
READ(5,5001)(IRSPNS(I), I=1,80) 0179
5001 FORMAT(80A1) 0179
IF(EOF(5)) 900,5 0179
 INFORMATION READ 0180
5 WRITE(6,5002) (IRSPNS(I), I=1,80) 0180
5002 FORMAT(/1X,2H? ,80A1) 0180

INITIALIZE VARIABLES AND FLAGS 0180

IDIG = 0 0180

NDIG1 = 0	0180
NDIG2 = 0	0180
IDECPT = 0	0180
IFIRST = 0	0181
ILAST = 0	0181
ISIGN = +1	0181
IVALUE = 0	0181
FVALUE = 0.	0181
VALUE1 = 0.	0181
VALUE2 = 0.	0181
I = 1	0181
FIND FIRST NON-BLANK CHARACTER IN USER RESPONSE. IF WE FALL THROUGH THE LOOP, INPUT CONSISTED ENTIRELY OF A BLANK LINE.	0181
10 ICHAR = IRSPNS(I)	0182
IF(ICHAR.NE.IBLANK) GO TO 20	0182
I = I + 1	0182
IF(I.LE.80) GO TO 10	0182
ITYPE = 4	0182
RETURN	0182
A NON-BLANK CHARACTER WAS FOUND. CHECK AGAINST 4 POSSIBILITIES - A SIGN (+ OR -), A DECIMAL POINT, A DIGIT, OR OTHERWISE.	0182
CHECK FOR A SIGN FIRST. IF FOUND, SET THE FLAG ISIGN AND RETURN TO CONTINUE SCANNING THE REST OF THE RESPONSE FOR A NON-BLANK CHARACTER	0183
20 IF(ICHAR.NE.IPLUS) GO TO 30	0183
I = I + 1	0183
IF(I.LE.80) GO TO 10	0183
ITYPE = 6	0184
GO TO 900	0184
30 IF(ICHAR.NE.MINUS) GO TO 40	0184
ISIGN = -1	0184
I = I + 1	0184
IF(I.LE.80)GO TO 10	0184
ITYPE = 6	0184
GO TO 900	0184
CHECK FOR A DECIMAL POINT (WITH NO DIGITS TO THE LEFT OF IT). IF THIS SHOULD BE THE CASE, SET FLAGS AND BRANCH OFF TO SCAN THE REST OF THE NUMBER (FRACTIONAL PART OF FLOATING POINT NUMBER).	0184
40 IF(ICHAR.NE.IPOINT) GO TO 50	0185
LDECPT = I	0185
IDECPT = 1	0185
NDIG1 = 0	0185
GO TO 200	0185
CHECK FOR AN INITIAL DIGIT. IF THIS SHOULD BE THE CASE, BRANCH OFF TO SCAN THE REST OF THE NUMBER (COULD BE INTEGER OR FLOATING POINT)	0185
50 DO 60 J=1,10	0186
IF(ICHAR.EQ.NUM(J)) IDIG = 1	0186
60 CONTINUE	0186
IF(IDIG.NE.1) GO TO 70	0186
NDIG1 = 1	0186
IFDIG1 = I	0186
GO TO 100	0186


```

APPARENTLY, RESPONSE IS A CHARACTER STRING (FIRST NON-BLANK CHARACTER WAS NEITHER A SIGN, A DECIMAL POINT NOR A DIGIT). MARK POSITION OF THE FIRST CHARACTER AND SCAN FROM THE END TO FIND THE LAST NON-BLANK CHARACTER. PASS THESE RESULTS BACK TO THE CALLING ROUTINE.
70 IFIRST = I
    ITYPE = 3
    J = 81
80 J = J - 1
    ILAST = J
    IF(J.EQ.I) GO TO 900
    ICHAR = IRSPNS(J)
    IF(ICCHAR.EQ.IBLANK) GO TO 80
    GO TO 900
-----
AT THIS POINT, AN INITIAL DIGIT HAS BEEN FOUND, BUT THE USER RESPONSE COULD STILL BE EITHER INTEGER OR FLOATING POINT (OR UNRECOGNIZABLE). CONTINUE SCANNING THE RESPONSE UNTIL A DISCRIMINATION CAN BE MADE.
100 IDIG = 0
    I = I + 1
    IF(I.GT.80) GO TO 120
    ICHAR = IRSPNS(I)
    DO 110 J=1,10
    IF(ICCHAR.EQ.NUM(J)) IDIG = 1
110 CONTINUE
    IF(IDIG.NE.1) GO TO 120
NEXT CHARACTER WAS A DIGIT - UPDATE STATUS FLAGS AND CONTINUE SCAN
    NDIG1 = NDIG1 + 1
    ILDIG1 = I
    GO TO 100
NEXT CHARACTER WAS NOT A DIGIT. CHECK FOR A DECIMAL POINT - IF FOUND IMPLIES USER RESPONSE IS A FLOATING POINT NUMBER, IN WHICH CASE BRANCH OFF TO SCAN FOR DIGITS TO THE RIGHT OF THE DECIMAL POINT (FRACTIONAL
120 IF(ICCHAR.NE.IPOINT) GO TO 130
    IDECPT = 1
    LDECPT = I
    ILDIG1 = I - 1
    IFDIG2 = I + 1
    ITYPE = 1
    GO TO 200
CHARACTER WAS NOT A DIGIT OR A DECIMAL POINT. CHECK TO SEE IF IT WAS BLANK (INTEGER TERMINATOR) - IF SO, WE CAN NOW BRANCH OFF AND DECODE INTEGER VALUE; IF NOT, RESPONSE IS UNRECOGNIZABLE.
130 IF(ICCHAR.NE.IBLANK) GO TO 140
    ILDIG1 = I - 1
    ITYPE = 0
    GO TO 300
140 ITYPE = 6
    GO TO 900
-----

```

```

THIS SECTION OF LOGIC IS ENTERED TO FIND ANY DIGITS TO THE RIGHT OF THE DECIMAL POINT IN A FLOATING POINT NUMBER.
200 IDIG = 0
    I = I + 1
    IF(I.GT.80) GO TO 300
    ICHAR = IRSPNS(I)
    DO 210 J=1,10
    IF(ICHAR.EQ.NUM(J)) IDIG = 1
210 CONTINUE
    IF(IDIG.NE.1) GO TO 220
    NDIG2 = NDIG2 + 1
    ILDIG2 = I
    GO TO 200

CHARACTER WAS NOT A DIGIT - CHECK FOR A BLANK (FLOATING POINT TERMINAL CHARACTER)
IF FOUND, BRANCH OFF TO DECODE THE FLOATING POINT VALUE; IF NOT, RESPONSE IS UNRECOGNIZABLE
220 IF(ICHAR.NE.IBLANK) GO TO 240

TERMINATING BLANK CHARACTER WAS FOUND. CHECK TO MAKE SURE RESPONSE DOES NOT CONSIST SIMPLY OF A DECIMAL POINT (NO DIGITS ON EITHER SIDE)
    IF((NDIG1.EQ.0).AND.(NDIG2.EQ.0)) GO TO 240
    ITYPE = 1
    ILDIG2 = I - 1
    GO TO 300
240 ITYPE = 6
    GO TO 900

-----

THIS SECTION OF LOGIC DECODES THE VALUE OF A FLOATING POINT OR INTEGER NUMBER FROM THE INDIVIDUAL DIGITS AND THE ORDER IN WHICH THEY OCCUR.
300 IF(IDECP.EQ.1) GO TO 400
    ITYPE = 0
    IF(NDIG1.LE.5) GO TO 310
    WRITE(6,6001)
6001 FORMAT(5X,17HINTEGER TOO LARGE)
    ITYPE = 6
    GO TO 900

INTEGER DECODE LOGIC
310 DO 320 J=IFDIG1,ILDIG1
    ICHAR = IRSPNS(J)
    DO 320 K=1,10
    IF(ICHAR.NE.NUM(K)) GO TO 320
    IVALUE = IVALUE * 10 + K - 1
320 CONTINUE
    IVALUE = ISIGN * IVALUE
    GO TO 900

FLOATING POINT DECODE LOGIC (USES INTEGER DECODE LOGIC TO OBTAIN VALUES OF LEFT AND RIGHT-HAND "INTEGERS")
400 ITYPE = 1
    IF(NDIG1.EQ.0) GO TO 450
    DO 420 J=IFDIG1,ILDIG1
    ICHAR = IRSPNS(J)

```

```

DO 420 K=1,10                                0199
IF(ICHAR.NE.NUM(K)) GO TO 420                 0199
IVALUE = IVALUE * 10 + K - 1                 0199
420 CONTINUE                                  0199
VALUE1 = IVALUE                               0199
IVALUE = 0                                    0199

450 IF(NDIG2.EQ.0) GO TO 500                  0200
DO 460 J=IFDIG2,ILDIG2                       0200
ICHAR = IRSPNS(J)                            0200
DO 460 K=1,10                                0200
IF(ICHAR.NE.NUM(K)) GO TO 460                0200
IVALUE = IVALUE * 10 + K - 1                 0200
460 CONTINUE                                  0200
VALUE2 = IVALUE                               0200

500 FVALUE = VALUE1 + VALUE2 / (10**NDIG2)    0200
FVALUE = ISIGN * FVALUE                      0201

CHECK THAT RESPONSE IS APPROPRIATE - IF NOT, USER MUST RE-INPUT 0201

900 IF(ITYPE.EQ.IDESIR) RETURN                0201

===PUT IN DESCRIPTIVE MESSAGES TO ALERT USER=== 0201

GO TO 1                                        0201
END                                             0201
SUBROUTINE ORBTR                               0202

*****
*
* SHUTTLE ORBITER BLOCK DATA
*
*****
THIS ROUTINE SETS UP THE STS ORBITER CONFIGURATION BY DEFINING 0202
GEOMETRIC SURFACES, THEIR IDENTIFICATION NUMBERS, LOCATION, MATERIAL 0202
AND AREA.
IDENT = SURFACE IDENTIFICATION NUMBER (1-999) 0203
SECT = ORBITER/SPACELAB GEOMETRIC SUBSECTION 0203
RADOOR= RADIATOR DOOR                        0203
BAY = PAYLOAD BAY LINER, SIDE STRIPS, BULKHEADS 0203
TAIL = TAILFIN                               0203
CREW = NOSE, CREW SECTION                    0203
WING = WINGS                                 0203
FUSLAG= FUSELAGE                             0203
OMS = OMS PODS                               0203
FILTER =OVERBOARD/INBOARD FILTERS           0204
MATRL = SURFACE MATERIAL                     0204
LINER = PAYLOAD BAY LINER                   0204
BLKHED= FORE AND AFT BAY BULKHEADS          0204
TEFLON= TEFLON                              0204
LRSI = LOW TEMP RSI                         0204
HRSI = HIGH TEMP RSI                       0204
NOMEX = PAINTED FELT                       0204
RCC = CARBON                                0204
CRACKS= LEAKING SURFACE                    0204
WINDOW= CABIN WINDOWS                      0205
FILF = INBOARD FILTERS                     0205
FILO = OVERBOARD FILTERS                   0205
AREA = SURFACE AREA IN SQUARE INCHES      0205

```

```

=====0205
COMMON /CNTRL/ ORBITR, PAYLOD, OUT, ED, LEAK, PLUME, 0205
+ MCD, DIRECT, RFAS2, RFSS, REFLCT, NRFLCT, 0205
+ NEWCON, NTAPE4, NEWTCD, NEWTNL, NEWMFS, NEWMFP, 0205
+ MINTMP, MAXTMP, ATCODE, R41DEP, GO, REPORT(70) 0206
COMMON/CONFIG/ SURFSC(300), SSURFS(300), PNTSC(50), NEWPL(50), 0206
+ ONTIME(50), RECEVR(25), ICCODE, FOVANG(25), 0206
+ SERIES, NEWDAT, ADSURF, NNEWPL, 0206
+ JTOTAL, KTOTAL. NORBPL, ISURF(300), 0206
+ ISSURF(300) 0206
COMMON/CHANGE/ CHNGES, CHNGEK, CHNGEP, CHNGPL 0206
COMMON /RATES/ RATE(25,10), TAU(25,10), TSTART(3), TSTOP(3), 0206
+ AGECRB, AGEPLD, MOUT1, MOUT2, MED1, MED2, 0206
+ M1, M2, AMBWT, AMBDIA, TSTARR, MACH, 0206
+ TIMEGO 0207
COMMON /ORBIT/ ALT, VA, PITCH, YAW, ROLL, SUNL, 0207
+ SUNM, SUNH 0207
COMMON /GEOM/ XORGIN, YORGIN, ZORGIN, XO(25), YO(25), ZO(25), 0207
+ ALPHA(25), BETA(25), GAMMA(25) 0207
COMMON /INTEG/ THETA(25), PHIL(25), THETA1(25), THETA2(25), 0207
+ DTHETA(25), PHI1(25), PHI2(25), DPHI(25), 0207
+ DOMEGA(25), DS(25), RMAX, NTHETA, 0207
+ NPHI 0207
COMMON /TEMPS/ TEMP(2000) 0207
COMMON/PLUMES/ PLUMEC(10,25), SPECMF(10,25), CXLOC(50), CYLOC(50), 0208
+ CZLOC(50), CTHETA(50), CPHI(50). 0208
+ CIDENT(50) 0208
COMMON/MOLEC/ MOLWT(10), DIA(10) 0208
COMMON /SURFS/IDENT(300), AREA(300) 0208
COMMON/CHAR/ITITLE(72), IRSPNS(80), NEWNAM(6), LTYPE(25), 0208
+ KINDS(25), PLACE(30), SPECIE(10), SECT(300), 0208
+ MATRL(300), NAMEPL, CLOC(50), CTYPE(50), 0208
+ NPLUME(25) 0208
COMMON/CHFLAG/ CHATT, CHALT, CHVEL, CHSUN, CHDS, CHORIG, 0208
+ CHTIM, CHAGE, CHINDX, CHWT, CHDIA, CHLOC(25), 0209
+ CHVIEW(25), CHRATE(25,10), CHTAU(25,10), 0209
+ CHPLUM(10,25), CHMF(10,25) 0209
COMMON/INDX/ INDXSP(25), INDXK(25), INDXP(30), INDXPL(25), 0209
+ INDXJT, INDXKT 0209
COMMON/NUCON/NUSECT(6,300), NUMATL(6,300), NULOC(6,50), 0209
+ NUTYPE(6,50), NUSPEC(6,10), NUKIND(6,25), 0209
+ NUPLAC(6,30), NUNPLM(6,25) 0209
0209
REAL ONTIME, MACH, MOLWT 0209
INTEGER SURFSC, SSURFS, PNTSC, RECEVR, ITITLE, LTYPE, KINDS, 0210
+ PLACE, SPECIE, SECT, MATRL, ATCODE, SERIES, CIDENT, 0210
+ CLOC, CTYPE, CHNGES, CHNGEK, CHNGEP, CHNGPL 0210
LOGICAL ORBITR, PAYLOD, OUT, ED, LEAK, PLUME, MCD, 0210
+ DIRECT, RFAS2, RFSS, REFLCT, NEWCON, NTAPE4, NEWTCD, 0210
+ NEWTNL, NEWMFS, NEWMFP, MINTMP, MAXTMP, REPORT, GO, 0210
+ NEWPL, SUNL, SUNM, SUNH, R41DEP, NEWDAT, ADSURF, 0210
+ CHATT, CHALT, CHVEL, CHSUN, CHDS, CHORIG, CHLOC, 0210
+ CHVIEW, CHTIM, CHRATE, CHTAU, CHAGE, CHPLUM, CHMF, 0210
+ CHINDX, CHWT, CHDIA, INDXSP, INDXK, INDXP, INDXPL 0210
0211
=====0211
DIMENSION IORB1(190), IORB2(190), IORB3(190), XORB4(190) 0211
0211
***** CURRENTLY THERE ARE 190 SURFACES USED IN THE ORBITER MODEL 0211
0211
DATA (IORB1(I), I=1,100) 0211

```


8	6HFILTER, 6HFILTER, 6HFILTER, 6HFILTER, 6H BAYL /	02:7
		0218
	DATA (IORB3(I), I=1,50)	0218
1	/ 6HTEFLON, 6HTEFLON, 6HTEFLON, 6HTEFLON, 6HTEFLON,	0218
2	6HTEFLON, 6HTEFLON, 6HTEFLON, 6HTEFLON, 6HTEFLON,	0218
3	6HTEFLON, 6HTEFLON, 6HTEFLON, 6HTEFLON, 6HTEFLON,	0218
4	6HTEFLON, 6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI,	0218
5	6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI,	0218
6	6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI,	0218
7	6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI,	0218
8	6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI,	0218
9	6H LRSI, 6H NOMEX, 6H NOMEX, 6H LRSI, 6H LRSI,	0219
*	6H NOMEX, 6H NOMEX, 6H LRSI, 6H LRSI, 6H LRSI /	0219
	DATA (IORB3(I), I=51,100)	0219
1	/ 6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI,	0219
2	6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI,	0219
3	6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI,	0219
4	6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI,	0219
5	6H LRSI, 6H NOMEX, 6H NOMEX, 6H NOMEX, 6H NOMEX,	0219
6	6H NOMEX, 6H LRSI, 6H HRSI, 6H HRSI, 6H LRSI,	0219
7	6H RCC, 6H RCC, 6H NOMEX, 6H NOMEX, 6H NOMEX,	0219
8	6H NOMEX, 6H NOMEX, 6H LRSI, 6H HRSI, 6H HRSI,	0220
9	6H LRSI, 6H RCC, 6H RCC, 6H NOMEX, 6H NOMEX,	0220
*	6H NOMEX, 6H NOMEX, 6H NOMEX, 6H NOMEX, 6H NOMEX /	0220
	DATA (IORB3(I), I=101,150)	0220
1	/ 6H NOMEX, 6H NOMEX, 6H NOMEX, 6H NOMEX, 6H NOMEX,	0220
2	6H NOMEX, 6H NOMEX, 6H NOMEX, 6H NOMEX, 6H NOMEX,	0220
3	6H NOMEX, 6H NOMEX, 6H NOMEX, 6H NOMEX, 6H NOMEX,	0220
4	6H NOMEX, 6H NOMEX, 6H RCC, 6H LRSI, 6H LRSI,	0220
5	6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI, 6H HRSI,	0220
6	6H HRSI, 6H HRSI, 6H HRSI, 6H HRSI, 6H HRSI,	0220
7	6H LRSI, 6H LRSI, 6H LRSI, 6HWINDOW, 6HWINDOW,	0221
8	6HWINDOW, 6HWINDOW, 6HWINDOW, 6HWINDOW, 6H LRSI,	0221
9	6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI,	0221
*	6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI /	0221
	DATA (IORB3(I), I=151,190)	0221
1	/ 6H LRSI, 6H LRSI, 6H LRSI, 6H LRSI, 6H HRSI,	0221
2	6H LINER, 6H LINER, 6H LINER, 6H LINER, 6H LINER,	0221
3	6H LINER, 6H LINER, 6H LINER, 6HBLKHED, 6HBLKHED,	0221
4	6H LINER, 6H LINER, 6H LINER, 6H LINER, 6H LINER,	0221
5	6H LINER, 6H LINER, 6H LINER, 6H FILI, 6H FILI,	0221
6	6H FILI, 6H FILI, 6H FILI, 6H FILI, 6H FILI,	0222
7	6H FILI, 6H FILO, 6H FILO, 6H FILO, 6H FILO,	0222
8	6H FILO, 6H FILO, 6H FILO, 6H FILO, 6HCRACKS /	0222
	DATA (XORB4(I), I=1,50)	0222
1	/ 12200., 12200., 12200., 12200., 12200.,	0222
2	12200., 12200., 12200., 25580., 25580.,	0222
3	25580., 25580., 25580., 25580., 25580.,	0222
4	25580., 12200., 12200., 12200., 12200.,	0222
5	12200., 12200., 12200., 12200., 25580.,	0222
6	25580., 25580., 25580., 25580., 25580.,	0223
7	25580., 25580., 32520., 32520., 25730.,	0223
8	16340., 16340., 19580., 20240., 26600.,	0223
9	30930., 30930., 24770., 26600., 30930.,	0223
*	30930., 24770., 1312., 1312., 1145. /	0223
	DATA (XORB4(I), I=51,100)	0223
1	/ 7850., 37920., 1991., 2028., 415.,	0223
2	895., 1406., 1312., 715., 600.,	0223
3	1145., 7813., 37740., 1991., 2028.,	0223
4	415., 895., 1406., 1312., 715.,	0223
5	601., 6356., 29590., 82 9125., 23340.,	0224

ORIGINAL PHOTO COPY OF POOR QUALITY

6	19280.	19280.	5650.	2508.	3302.	0224
7	2251.	3123.	6356.	29590.	9125.	0224
8	23340.	19280.	19280.	5650.	2508.	0224
9	3302.	2251.	3123.	6499.	17210.	0224
*	6499.	9125.	138.	415.	692.	0224
DATA (XORB4(I),I=101,150)						
1	/ 960.	1246.	1523.	1800.	2076.	0224
2	2353.	2630.	138.	415.	692.	0224
3	969.	1246.	1523.	1800.	2076.	0224
4	2353.	2630.	7191.	9348.	9348.	0225
5	3380.	3380.	4253.	4253.	12590.	0225
6	12590.	9600.	9600.	3705.	3705.	0225
7	20720.	10150.	10150.	1424.	1424.	0225
8	1424.	1424.	1424.	1424.	10250.	0225
9	16920.	16920.	8833.	8833.	13940.	0225
*	13940.	6116.	6116.	2744.	2744.	0225
DATA (XORB4(I),I=151,190)						
1	/ 1160.	1160.	3081.	3081.	3823.	0225
2	26620.	26620.	26620.	26620.	26620.	0225
3	26620.	26620.	26620.	32690.	32690.	0226
4	3444.	3444.	3444.	3444.	3444.	0226
5	3444.	3444.	3444.	207.	207.	0226
6	207.	207.	207.	207.	207.	0226
7	207.	144.	144.	144.	144.	0226
8	144.	144.	144.	144.	32690.	0226

LOAD ALL OF THE ABOVE ORBITER DATA INTO THE PROPER ARRAYS

```

II = 0
DO 40 I=1,190
  II = II + 1
  IDENT(II) = IORB1(I)
  SECT(II) = IORB2(I)
  MATRL(II) = IORB3(I)
  AREA(II) = XORB4(I)
  IF(MATRL(II).NE.6HCRACKS) GO TO 30
  IF(LEAK) GO TO 30
  II = II - 1
  GO TO 35
30 CONTINUE
  IF((SECT(II).NE.6H BAY).AND.(SECT(II).NE.6H BAYL).AND.
  +(SECT(II).NE.6HFILTER)) GO TO 35
  IF(PAYLOD) II = II - 1
35 CONTINUE
40 CONTINUE

JTOTAL = II

DO 100 I=1,JTOTAL
  SSURFS(I) = IDENT(I)
100 SURFSC(I) = IDENT(I)
  RETURN
  END
  SUBROUTINE LMOPX

*****
*
*  L M O P  B L O C K  D A T A  *
*
*****

RETURN

```

ORIGINAL PAGE 18
OF POOR QUALITY

+	AGEORB, AGEPLD, MOUT1, MOUT2, MED1, MED2,	0236
+	M1, M2, AMBWT, AMBDIA, TSTARR, MACH,	0236
+	TIMEOO	0236
+	COMMON /ORBIT/ ALT, VA, PITCH, YAW, ROLL, SUNL,	0236
+	SUNM, SUNH	0236
+	COMMON /GEOM/ XORGIN, YORGIN, ZORGIN, X0(25), Y0(25), Z0(25),	0237
+	ALPHA(25), BETA(25), GAMMA(25)	0237
+	COMMON /INTEG/ THETA(25), PHIL(25), THETA1(25), THETA2(25),	0237
+	DTHETA(25), PHI1(25), PHI2(25), DPHI(25),	0237
+	DOMEGA(25), DS(25), RMAX, NTHETA,	0237
+	NPHI	0237
+	COMMON /TEMPS/ TEMP(2000)	0237
+	COMMON/PLUMES/ PLUMEC(10,25), SPECMF(10,25), CXLOC(50), CYLOC(50),	0237
+	CZLOC(50), CTHETA(50), CPHI(50),	0237
+	CIDENT(50)	0237
+	COMMON/MOLEC/ MOLWT(10), DIA(10)	0238
+	COMMON /SURFS/IDENT(300), AREA(300)	0238
+	COMMON/CHAR/ITITLE(72), IRSPNS(80), NEWNAM(6), LTYPE(25),	0238
+	KINDS(25), PLACE(30), SPECIE(10), SECT(300),	0238
+	MATRL(300), NAMEPL, CLOC(50), CTYPE(50),	0238
+	NPLUME(25)	0238
+	COMMON/CHFLAG/ CHATT, CHALT, CHVEL, CHSUN, CHDS, CHORIG,	0238
+	CHTIM, CHAGE, CHINDX, CHWT, CHDIA, CHLOC(25),	0238
+	CHVIEW(25), CHRATE(25,10), CHTAU(25,10),	0238
+	CHPLUM(10,25), CHMF(10,25)	0238
+	COMMON/INDX/ INDXSP(25), INDXK(25), INDXP(30), INDXPL(25),	0239
+	INDXJT, - INDXKT	0239
+	COMMON/NUCON/NUSECT(6,300), NUMATL(6,300), NNULOC(6,50),	0239
+	NUTYPE(6,50), NUSPEC(6,10), NUKIND(6,25),	0239
+	NUPLAC(6,30), NUNPLM(6,25)	0239
+	REAL ONTIME, MACH, MOLWT	0239
+	INTEGER SURFSC, SSURFS, PNTSC, RECEVR, ITITLE, LTYPE, KINDS,	0239
+	PLACE, SPECIE, SECT, MATRL, ATCODE, SERIES, CIDENT,	0239
+	CLOC, CTYPE, CHNGES, CHNGEK, CHNGEP, CHNGPL	0239
+	LOGICAL ORBITR, PAYLOD, OUT, ED, LEAK, PLUME, MCD,	0240
+	DIRECT, RFAS2, RFSS, REFLCT, NEWCON, NTAPE4, NEWTCD,	0240
+	NEWTNL, NEWMFS, NEWMFP, MINTMP, MAXTMP, REPORT, GO,	0240
+	NEWPL, SUNL, SUNM, SUNH, R41DEP, NEWDAT, ADSURF,	0240
+	CHATT, CHALT, CHVEL, CHSUN, CHDS, CHORIG, CHLOC,	0240
+	CHVIEW, CHTIM, CHRATE, CHTAU, CHAGE, CHPLUM, CHMF,	0240
+	CHINDX, CHWT, CHDIA, INDXSP, INDXK, INDXP, INDXPL	0240
=====		0240
ORBITR=.TRUE.		0240
PAYLOD=.FALSE.		0241
OUT=.TRUE.		0241
ED=.FALSE.		0241
LEAK=.FALSE.		0241
PLUME=.FALSE.		0241
MCD=.TRUE.		0241
DIRECT=.FALSE.		0241
RFAS2=.FALSE.		0241
RFSS=.FALSE.		0241
REFLCT=.FALSE.		0241
NRFLCT=1		0242
NEWCON=.FALSE.		0242
NTAPE4=.FALSE.		0242
NEWTCD=.FALSE.		0242
NEWTNL=.FALSE.		0242
NEWMFS=.FALSE.		0242

ORIGINAL PAGE IS
OF POOR QUALITY

NEWMFP=.FALSE.	0242
MINTMP=.FALSE.	0242
MAXTMP=.TRUE.	0242
ATCODE=0	0243
	0243
DO 100 I=1,300	0243
SURFSC(I)=0	0243
SSURFS(I)=0	0243
ISURF(I)=0	0243
ISSURF(I)=0	0243
100 CONTINUE	0243
DO 110 I=1,30	0243
PNTSC(I)=0	0243
NEWPL(I)=.FALSE.	0244
ONTIME(I)=0.0	0244
110 CONTINUE	0244
DO 120 I=1,25	0244
RECEVR(I)=0	0244
FOVANG(I)=180.	0244
120 CONTINUE	0244
ICCODE=0	0244
SERIES=1000	0244
	0244
CHNGES=0	0245
CHNGEK=0	0245
CHNGEP=0	0245
CHNGPL=0	0245
MACH=1.0	0245
	0245
ALT=400.	0245
SUNL=.FALSE.	0245
SUNM=.TRUE.	0245
SUNH=.FALSE.	0245
VA=7650.	0246
PITCH=0.	0246
YAW=0.	0246
ROLL=0.	0246
DO 150 I=1,25	0246
X0(I)=1107.	0246
Y0(I)=0.	0246
Z0(I)=507.	0246
ALPHA(I)=0.	0246
BETA(I)=0.	0246
GAMMA(I)=0.	0247
THETA1(I)=0.	0247
THETA2(I)=10.24	0247
PHI1(I)=0.	0247
PHI2(I)=360.	0247
DTHETA(I)=10.24	0247
DPHI(I)=45.	0247
150 CONTINUE	0247
THETAL(1)=0.0	0247
DO 160 I=1,8	0247
I1=I+1	0248
I2=I+9	0248
I3=I+17	0248
THETAL(I1)=30.	0248
THETAL(I2)=60.	0248
THETAL(I3)=82.5	0248
160 CONTINUE	0248
PHIL(1)=0.0	0248
DO 170 I=1,3	0248

ORIGINAL PAGE IS
OF POOR QUALITY

5- C02		.078(.036)	0255
6- O2		.001(.001)	0255
7- CO		.184(.134)	0255
8- H2		.017(.170)	0255
9- H		.001(.015)	0255
10- MMHNO3		.002(.001)	0255

ORIGINAL PAGE IS
OF POOR QUALITY

THIS ROUTINE SETS UP THE STS ORBITER CONFIGURATION BY DEFINING
THEIR IDENTIFICATION NUMBERS, LOCATION, AND ORIENTATION.

CIDENT	= ENGINE/VENT IDENTIFICATION NUMBER(5000-5999)	0256
CLOC	= ENGINE/VENT SIX LETTER CODE TO DEFINE LOCATION	0256
CTYPE	= SIX LETTER NAME TO DEFINE TYPE OF PROPULSION SYSTEM	0256
CXLCC	= X LOCATION OF ENGINE/VENT IN ORBITER OR BASE COORDS.	0256
CYLOC	= Y LOCATION	0256
CZLOC	= Z LOCATION	0256
CTHETA	= ORIENTATION OF NOZZLE CENTERLINE OFF Z AXIS	0256
CPHI	= ORIENTATION OF CENTERLINE FROM +X AXIS	0256

***** NOTE - THIS ROUTINE'S STRUCTURE IS NOT ANSI STANDARD FORTRAN IV
***** IT SHOULD BE MODIFIED: PULL ALL DATA STATEMENTS FORWARD
***** OF ANY EXECUTABLE CODE AND SEPARATE MIXED MODE ASSIGNMENT
***** INTO SEPARATE DATA STATEMENTS.

```

=====
COMMON /CNTRL/ ORBITR, PAYLOD, OUT, ED, LEAK, PLUME,
+ MCD, DIRECT, RFAS2, RFSS, REFLCT, NRFLCT,
+ NEWCON, NTAPE4, NEWTCD, NEWTNL, NEWMFS, NEWMFP,
+ MINTMP, MAXTMP, ATCODE, R41DEP, GO, REPORT(70)
COMMON/CONFIG/ SURFSC(300), SSURFS(300), PNTSC(50), NEWPL(50),
+ ONTIME(50), RECEVR(25), ICCODE, FOVANG(25),
+ SERIES, NEWDAT, ADSURF, NNEWPL,
+ JTOTAL, KTOTAL, NORBPL, ISURF(300),
+ ISSURF(300)
COMMON/CHANGE/ CHNGES, CHNGEK, CHNGEP, CHNGPL
COMMON /RATES/ RATE(25,10), TAU(25,10), TSTART(3), TSTOP(3),
+ AGEORB, AGEPLD, MOUT1, MOUT2, MED1, MED2,
+ M1, M2, AMBWT, AMBDIA, TSTARR, MACH,
+ TIMEOO
COMMON /ORBIT/ ALT, VA, PITCH, YAW, ROLL, SUNL,
+ SUNM, SUNH
COMMON /GEOM/ XORGIN, YORGIN, ZORGIN, XO(25), YO(25), ZO(25),
+ ALPHA(25), BETA(25), GAMMA(25)
COMMON /INTEG/ THETA(25), PHIL(25), THETA1(25), THETA2(25),
+ DTHETA(25), PHI1(25), PHI2(25), DPHI(25),
+ DOMEGA(25), DS(25), RMAX, NTHETA,
+ NPHI
COMMON /TEMPS/ TEMP(2000)
COMMON/PLUMES/ PLUMEC(10,25), SPECMF(10,25), CXLOC(50), CYLOC(50),
+ CZLOC(50), CTHETA(50), CPHI(50),
+ CIDENT(50)
COMMON/MOLEC/ MOLWT(10), DIA(10)
COMMON /SURFS/IDENT(300), AREA(300)
COMMON/CHAR/ITITLE(72), IRSPNS(80), NEWNAM(6), LTYPE(25),
+ KINDS(25), PLACE(30), SPECIE(10), SECT(300),
+ MATRL(300), NAMEPL, CLOC(50), CTYPE(50),
+ NPLUME(25)
COMMON/CHFLAG/ CHATT, CHALT, CHVEL, CHSUN, CHDS,CHORIG,
+ CHTIM, CHAGE, CHINDX, CHWT, CHDIA, CHLOC(25),
+ CHVIEW(25), CHRATE(25,10), CHTAU(25,10),
+ CHPLUM(10,25), CHMF(10,25)
COMMON/INDX/ INDXSP(25), INDXK(25), 88 INDXP(30), INDXPL(25),

```

```

+          INDXJT,      INDXKT
COMMON/NUCON/NUSECT(6,300), NUMATL(6,300), NULOC(6,50),      0261
+          NUTYPE(6,50), NUSPEC(6,10), NUKIND(6,25),          0261
+          NUPLAC(6,30), NUNPLM(6,25)                          0261
REAL ONTIME, MACH, MOLWT                                       0261
INTEGER SURFSC, SSURFS, PNTSC, RECEVR, ITITLE, LTYPE, KINDS,  0261
+          PLACE, SPECIE, SECT, MATRL, ATCODE, SERIES, CIDENT, 0261
+          CLOC, CTYPE, CHNGES, CHNGEK, CHNGEP, CHNGPL,        0262
LOGICAL ORBITR, PAYLOD, OUT, ED, LEAK, PLUME, MCD,            0262
+          DIRECT, RFAS2, RFSS, REFLCT, NEWCON, NTAPE4, NEWTCD, 0262
+          NEWTNL, NEWMFS, NEWMFP, MINTMP, MAXTMP, REPORT, GO,  0262
+          NEWPL, SUNL, SUNM, SUNH, R41DEP, NEWDAT, ADSURF,     0262
+          CHATT, CHALT, CHVEL, CHSUN, CHDS, CHORIG, CHLOC,    0262
+          CHVIEW, CHTIM, CHRATE, CHTAU, CHAGE, CHPLUM, CHMF,  0262
+          CHINDX, CHWT, CHDIA, INDXSP, INDXK, INDXP, INDXPL    0262

```

```

=====
DIMENSION SPDATA(250),PFDATA(250),NPLME(25), IPTS(300)      0263
*****

```

LOAD IN THE PLUME FUNCTION COEFICIENTS FOR THE FOLLOWING ENGINES

```

KKINDS = 25
DATA(NPLME(K),K=1,25)
TYPES OF ENGINES/VENTS
1 / 6H RCS,
2 6H VCS,
3 6H OMS,
4 6H EVAP1,
5 6H E05HE,
6 6H CO2XE,
7 6H XECH4,
8 6H E13HE,
9 6H UMBV1,
* 6H UMBV2,
1 6H IUSSM,
2 6H IUSLM,
3 13*6H NULL/

```

ORIGINAL PAGE IS
OF POOR QUALITY

```

DO 5 K=1, KKINDS
5 NPLUME(K) = NPLME(K)

```

LOAD IN THE PLUME FUNCTION COEFICIENTS

```

DATA(PFDATA(K),K=1,250)
C1 C2 C3 THETA1 C5 C6 THETA2 MFLUX VELOC TYPE
1 /1351.,10.00,.0126,64.0,35.0,-.0840,180.,0.,3.5E+5,6H RCS,
2 23.2,8.65,.0137,40.0,5.810,-.0467,140.,0.,3.5E+5,6H VCS,
3 9332.,10.65,.0126,64.0,235.0,-.0840,180.,0.,3.5E+5,6H OMS,
4 1.963,6.00,.0106,148.,0.,0.,148.,0.,1.0E+5,6H EVAP1,
5 .00404,1.75,.0174,90.,0.,0.,90.,0.,7.8E+4,6H E05HE,
6 .00136,1.75,.0174,90.,0.,0.,90.,0.,7.8E+4,6H CO2XE,
7 .01220,1.75,.0174,90.,0.,0.,90.,0.,7.8E+4,6H XECH4,
8 .00243,1.75,.0174,90.,0.,0.,90.,0.,7.8E+4,6H E13HE,
9 .64800,1.75,.0174,90.,0.,0.,90.,0.,7.8E+4,6H UMBV1,
* .00136,1.75,.0174,90.,0.,0.,90.,0.,7.8E+4,6H UMBV2,
1 15061.,10.65,.0126,64.,299.,-.0822,179,0.,3.5E+5,6H IUSSM,
2 17752.,10.65,.0126,64.,352.,-.0622,179.89,0.,3.5E+5,6H IUSLM,

```

3 130*0./

ORIGINAL PAGE IS
OF POOR QUALITY

```

DO 20 K=1, KKINDS
DO 10 L=1, 10
10 PLUMEC(L,K) = PFDATA( (K-1) * 10 + L )
20 CONTINUE

```

LOAD IN THE SPECIES MASS FRACTIONS TO BE USED FOR THE ENGINES

DATA(SPDATA(K), K=1, 250)										
TYPE	OUT1	OUT2	H2O	N2	CO2	O2	CO	H2	H	MMH HNO3
1	0.0,	0.0,	.290,	.420,	.078,	.001,	.184,	.017,	.001,	.002,
2	0.0,	0.0,	.290,	.420,	.078,	.001,	.184,	.017,	.001,	.002,
3	0.0,	0.0,	.290,	.420,	.078,	.001,	.184,	.017,	.001,	.002,
4	0.0,	0.0,	1.000,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,
								HE	XE	CH4
5	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	1.0,	0.0,	0.0,
6	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.1,	0.0,	0.9,
7	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.9,	0.1,
8	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	1.0,	0.0,	0.0,
9	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	1.0,	0.0,	0.0,
*	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.0,	0.1,	0.0,	0.9,
AL2O3		CO	HCL	N2	H2O	CO2	ALCL	H2	ALOCL	OTHERS
1.	3130,	.2713,	.1803,	.0822,	.0579,	.0206,	.0215,	.0267,	.0106,	.0158,
2.	3130,	.2713,	.1803,	.0822,	.0579,	.0206,	.0215,	.0267,	.0106,	.0158,
3	130*0./									

```

DO 40 K=1, KKINDS
DO 30 L=1, 10
30 SPECMF(L,K)=SPDATA( (K-1)*10+L )
40 CONTINUE

```

CURRENTLY THERE ARE 48 ENGINES/VENTS USED ON THE ORBITER

* * MASTER ARRAY FOR ENGINES * *

FORWARD RCS ENGINES

DATA(IPTS(I), I=1, 84)						
1	7112,6HFLF	-X,6H	RCS,	332,	-14,	389,
2	7122,6HFCF	-X,6H	RCS,	332,	0,	391,
3	7132,6HFRF	-X,6H	RCS,	332,	14,	389,
4	7123,6HFLL	+Y,6H	RCS,	360,	-47,	368,
5	7113,6HFLL	+Y,6H	RCS,	360,	-47,	354,
6	7115,6HFLL	+Z,6H	RCS,	350,	-13,	395,
7	7125,6HFLL	+Z,6H	RCS,	350,	0,	395,
8	7135,6HFLL	+Z,6H	RCS,	350,	13,	395,
9	7116,6HFLL	-Z,6H	RCS,	333,	-41,	381,
*	7126,6HFLL	-Z,6H	RCS,	347,	-45,	386,
1	7144,6HFLL	-Y,6H	RCS,	362,	47,	368,
2	7134,6HFLL	-Y,6H	RCS,	362,	47,	354,
3	7136,6HFLL	-Z,6H	RCS,	333,	41,	381,
4	7146,6HFLL	-Z,6H	RCS,	347,	45,	386/

AFT RCS ENGINES LEFT SIDE OF ORBITER

DATA(IPTS(I), I=85, 156)						
5	7211,6HALA	+X,6H	RCS,	1557,	-119,	473,
6	7231,6HALA	+X,6H	RCS,	1557,	-132,	473,
7	7243,6HALS	+Y,6H	RCS,	1516,	-123,	459,
8	7223,6HALS	+Y,6H	RCS,	1529,	-123,	459,
9	7233,6HALS	+Y,6H	RCS,	1542,	-122,	459,
*	7213,6HALS	+Y,6H	RCS,	1555,	-122,	459,

+41H-		-/14X,	0286
+41H-	-----	//)	0286
RETURN			0286
H E A D E R F O R S U B R O U T I N E I N P U T A			
300	WRITE(6,6030)		0286
6030	FORMAT(/9X,		0286
+51H-	-----	-/9X,	0286
+51H-		-/9X,	0287
+51H-	CONTAMINATION SOURCE	-/9X,	0287
+51H-		-/9X,	0287
+51H-	DEFINITION	-/9X,	0287
+51H-	-----	-/9X,	0287
RETURN)	0287
400	RETURN		0287
500	RETURN		0287
600	RETURN		0287
700	RETURN	ORIGINAL PAGE IS	0288
800	RETURN	OF POOR QUALITY	0288
900	RETURN		0288
	END		0288

END
DATE

DEC. 29, 1983