AUTOMATED DOCUMENTATION GENERATOR
for
ADVANCED PROTEIN CRYSTAL GROWTH
(5-32809)

Final Technical Report for Period
7 November 1991 through 31 January 1993
January 1993

Prepared by
Gary A. Maddux
Anna Provancha
David Chattam
Ronald Ford

Research Institute
The University of Alabama in Huntsville
Huntsville, Alabama 35899

Prepared for
George C. Marshall Space Flight Center
National Aeronautics and Space Administration
Marshall Space Flight Center, AL 35812
Attn: JA81 (Mr. David Jex)
# AUTOMATED DOCUMENTATION GENERATOR for ADVANCED PROTEIN CRYSTAL GROWTH

**Authors:**
- Gary A. Maddux
- Anna Provancha
- David Chattam
- Ronald Ford

**Performing Organization Name and Address**
Research Institute
The University of Alabama in Huntsville
Huntsville, Alabama 35899

**Sponsoring Agency Name and Address**
George C. Marshall Space Flight Center
National Aeronautics and Space Administration
Marshall Space Flight Center, AL 35812

**Supplementary Notes**
See Attached Report

**Abstract**
See Attached Report
PREPARATION OF THE REPORT DOCUMENTATION PAGE

The last page of a report facing the third cover is the Report Documentation Page. Information presented on this page is used in announcing and cataloging reports as well as preparing the cover and title page. Thus it is important that the information be correct. Instructions for filling in each block of the form are as follows:

Block 1. Report No. NASA report series number or reassigned.

Block 2. Government Accession No. Leave blank.

Block 3. Recipient's Catalog No. Reserved for use by each report recipient.

Block 4. Title and Subtitle. Typed in caps and lower case with dash or period separating subtitle from title.

Block 5. Report Date. Approximate month and year the report will be published.


Block 7. Author(s). Provide full names exactly as they are to appear on the title page. If applicable, the word editor should follow a name.

Block 8. Performing Organization Report No. NASA installation report control number and, if desired, the non-NASA performing organization report control number.

Block 9. Performing Organization Name and Address. Provide affiliation (NASA program office, NASA installation, or contractor name) of authors.

Block 10. Work Unit No. Provide Research and Technology Objectives and Plans (RTOP) number.

Block 11. Contract or Grant No. Provide when applicable.


Block 13. Type of Report and Period Covered. NASA formal report series; for Contractor Report also list type (interim, final) and period covered when applicable.


Block 15. Supplementary Notes. Information not included elsewhere: affiliation of authors if additional space is required for block 9 notice or work sponsored by another agency, monitor or contract, information about supplements (film, data tapes, etc.), meeting site and date for presented papers, journal to which an article has been submitted, note of a report made from a thesis, appendix by author other than shown in block 7.

Block 16. Abstract. The abstract should be informative rather than descriptive and should state the objectives of the investigation, the methods employed (e.g., simulation, experiment, or remote sensing), the results obtained, and the conclusions reached.

Block 17. Key Words. Identifying words or phrases to be used in cataloging the report.

Block 18. Distribution Statement. Indicate whether report is available to public or not. If not to be controlled, use "Unclassified-Unlimited." If controlled availability is required, list the category approved on the Document Availability Authorization Form (see NHB 2200.2, Form FF427). Also specify subject category (see "Table of Contents" in a current issue of STAR), in which report is to be distributed.


Block 21. No. of Pages. Count front matter pages beginning with i, text pages including internal blank pages, and the RDP, but not the title page or the back of the title page.

Block 22. Price Code. If block 18 shows "Unclassified-Unlimited," provide the NTIS price code (see "NTIS Price Schedules" in a current issue of STAR) and at the bottom of the form add either "For sale by the National Technical Information Service, Springfield, VA 22161-2171" or "For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402-0001" whichever is appropriate.
This technical report was prepared by the staff of the Research Institute, The University of Alabama in Huntsville. The purpose of this report is to provide documentation of the work performed and results obtained under delivery order 17 of Marshall Space Flight Center (MSFC) Contract No. NAS8-38609. Mr. Gary A. Maddux was Principal Investigator for this fourteen month level of effort. Mr. David Jex of the Microgravity Experiment Projects Office provided technical coordination.

The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official NASA position, policy, or decision unless so designated by other official documentation.

I have reviewed this report, dated 1-22-93 and the report contains no classified information.

[Signature]
Principal Investigator

Approval:

[Signature]
Research Institute
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2.0 BACKGROUND AND OBJECTIVES</td>
<td>1</td>
</tr>
<tr>
<td>3.0 CURRENT ENVIRONMENT</td>
<td>1</td>
</tr>
<tr>
<td>4.0 ACTIVITIES</td>
<td>2</td>
</tr>
<tr>
<td>4.1 Form and Documentation Evaluation</td>
<td>2</td>
</tr>
<tr>
<td>4.2 Design Considerations</td>
<td>2</td>
</tr>
<tr>
<td>4.2.1 Hardware</td>
<td>2</td>
</tr>
<tr>
<td>4.2.2 Software</td>
<td>3</td>
</tr>
<tr>
<td>4.3 Software Packages</td>
<td>3</td>
</tr>
<tr>
<td>4.3.1 Science Requirements Document</td>
<td>3</td>
</tr>
<tr>
<td>4.3.2 Experiment Requirements Document</td>
<td>4</td>
</tr>
<tr>
<td>4.3.3 Project Plan</td>
<td>5</td>
</tr>
<tr>
<td>4.3.4 Science Requirements Envelope Document</td>
<td>5</td>
</tr>
<tr>
<td>4.3.5 Safety Requirements Document</td>
<td>6</td>
</tr>
<tr>
<td>5.0 PRELIMINARY RESULTS OF SOFTWARE DISTRIBUTION</td>
<td>6</td>
</tr>
<tr>
<td>6.0 LIMITATIONS OF DELIVERED SOFTWARE</td>
<td>7</td>
</tr>
<tr>
<td>7.0 CONCLUSIONS AND RECOMMENDATIONS</td>
<td>7</td>
</tr>
</tbody>
</table>

## APPENDICES

A SRD Logical Flow Chart
B SRD/ERD User Guide
C APET Hierarchy of Menu Choices for SRD/ERD Package
D Project Plan User Guide
E Science Requirements Envelope Document User Guide
F Project Plan Software Listing
G Science Requirements Envelope Document Software Listing
1.0 INTRODUCTION

The System Management and Production Laboratory at the Research Institute, the University of Alabama in Huntsville (UAH), was tasked by the Microgravity Experiment Projects (MEP) Office of the Payload Projects Office (PPO) at Marshall Space Flight Center (MSFC) to conduct research in the current methods of written documentation control and retrieval. The goals of this research were to determine the logical interrelationships within selected NASA documentation, and to expand on a previously developed prototype system to deliver a distributable, electronic knowledge-based system. This computer application would then be used to provide a "paperless" interface between the appropriate parties for the required NASA document.

2.0 BACKGROUND AND OBJECTIVES

The Microgravity Experiment Projects (MEP) Office of the Payload Projects Office (PPO) at MSFC is currently responsible for collecting and coordinating experiment/facility specifications and requirements between NASA and various colleges, universities, research centers, and other public- and private-sector organizations that are selected or are requesting to fly their respective microgravity experiments on designated flights. This coordination involves the communication of flight hardware requirements and the preparation and review of all documentation between NASA and the research groups. To reduce difficulties encountered by these customers of NASA, an effort was undertaken to research, analyze, and evaluate the current procedures involved in the information gathering activities.

The MEP Office identified a need to develop an Automated Payload Experiment Tool (APET) which would lead experiment developers through the development planning process, obtain necessary information, establish an electronic data exchange avenue and allow easy manipulation/reformatting of the collected information. In order to fulfill this need, the University of Alabama in Huntsville (UAH) was tasked to design and develop the APET software package to meet the increasing demands to lighten the burden of documentation preparation and maintenance for NASA and its customers.

The objective of this task was to expand on the results of the Automated Payload Experiment Tool (APET) Feasibility Study (previously performed by UAH) and provide procedures and software for the generation of experiment requirements and flight hardware requirements. The software would assist the scientist or engineer in generating the appropriate documentation to develop and perform flight qualified experiments for the manned microgravity environment.

3.0 CURRENT ENVIRONMENT

The current environment of manual data gathering and information dissemination is excessively reliant on paper as the primary medium of transfer. This reliance on a static media adds exponentially to the complexity of a process that by its nature is elaborate. Changes to a document stored on an information media that requires physical manipulation are costly and burden-
some. With no method in place to ensure that changes are incorporated throughout follow-on documents, (other than manual verification), modifications to science, engineering, safety, and other documents are more susceptible to human error than necessary.

The design, development and preparation of an experiment to fly in space are time consuming tasks demanding a great deal of technical and disciplinary knowledge. Reducing the time required to prepare an experiment and its supporting documentation is of vital interest to the Microgravity Science Applications Division (MSAD). Methods of developing and utilizing state of the art information technologies are of prime concern in simplifying the critical Principal Investigator (PI)/Payload Element Developer (PED) interface.

4.0 ACTIVITIES

4.1 Form and Documentation Evaluation

UAH collected, analyzed, organized, and evaluated a number of forms and documents used in the current flight hardware development process. Documents were analyzed as to their content, and also evaluated regarding their relationships both within the same document and within other documents. The findings of this research were incorporated in the design of the computer software and its accompanying knowledge base.

4.2 Design Considerations

4.2.1 Hardware

The objective of APET is to provide an easy to use tool to the Principal Investigator team. To ensure ease of use, few computer hardware requirements are necessary to operate the APET software package.

APET is designed to run on any IBM-PC compatible personal computer. There have been four modules developed and distributed, and each module's requirements depend on its level of complexity. These four modules are the SRD, the SRD/ERD, the Project Plan, and the Science Requirements Envelope Document. All four packages will run on a 386 machine. However, while it is possible to use the SRD/ERD system on a 386 PC, it is recommended that the APET user install the software on a 486 PC or higher. The graphical displays, multiple screen windows, and the complexity of the system cause noticeable slowdowns on any machine less than the 486.

The software requires that the PC be equipped with a hard disk drive. For proper execution, the hard drive (or some partition of it) must be named C:. The SRD/ERD version of APET will require approximately 14M (megabytes) of space on the hard drive for the system, plus another 1M on the hard drive for the data files created by the user. However, for optimal performance, the hard drive should have a total of at least 17M free upon installation of the software. The Project Plan and the Science Requirements Envelope Document will each require from 1M - 3M of hard disk space.
For ease of use, the PC should be equipped with a mouse. This, however, is not mandatory. APET utilizes hypertext technology as the user interface. Hypertext software systems allow for the retrieval of related information at the point and click of a mouse or, if a mouse is not used, at the touch of one or two keystrokes. For information on a highlighted topic, the user should move the mouse to that word and click. A window will be opened, overlaying the current screen. Once the information is reviewed, the user can press the space bar and close the window, returning the user to the previous screen.

4.2.2 Software

One of the primary objectives in developing the APET was to reduce the confusion of the documentation process. This guiding principle was instrumental in the design of the software as well. The four APET software packages use a standardized format for the user interface. Screen design, menu selection, method of data entry, and user messages take the same form throughout the APET packages.

In developing the questions and knowledge base for the APET, it was assumed that all necessary instructions for successful completion of PI requirements were contained in the hardcopy documents. Therefore, to formulate user prompts, questions, explanations, etc., the questions and background information were taken directly from the applicable document. Definitions were taken from pertinent entries within the applicable document. When conflicting definitions were found to exist, the most logical definition was used in that software section.

4.3 Software Packages

4.3.1 Science Requirements Document

According to the Microgravity Science Applications Division (MSAD) Management Plan, "The Science Requirements Document (SRD) is the basic document which levies the science requirements on the hardware. As such, the document must first provide adequate justification for conducting the experiment in space and then delineate and justify the individual science requirements. The science requirements include the observational and environmental data requirements necessary to meet the science objectives".

The SRD is the first documentation requirement to be met by the Principal Investigator. It was also the logical beginning for the APET software. The SRD section of APET consists of a query of 52 questions concerning the description of the experiment, the limitations of non-space testing, and the potential benefits from the space environment. The answers to these questions are narrative in form (unlike other parts of APET, which are more fill-in-the-blank or choose from a list). User prompts for these questions were taken directly from the MSAD Management Plan. The user has the option of answering these questions sequentially or randomly (See Appendix A). An option also exists to answer only the unaddressed questions, so that at any time the user can see how many questions remain. There are a number of options available to the user to make the documentation process more efficient. For example, during the question/answer session, the user has the option of viewing/editing
related answers on selected questions. This adds to the consistency of the material, and provides an easier data retrieval method for the user. For a full description of the SRD software package, see the SRD/ERD User Guide (Appendix B) and the APET Hierarchy of Menu Choices for SRD/ERD Package (Appendix C).

4.3.2 Experiment Requirements Document

The Experiment Requirements Document (ERD) is used by the payload element developer and/or the principal investigator to define experiment requirements to be accommodated by the Space Transportation System (STS) for a given mission. The ERD is the logical follow-on document to the SRD. While the SRD justifies the need for a space environment and generally describes experiment requirements, the ERD expands on these generalities and requires specific experiment specifications.

Because of the more exacting nature of the ERD, the user faces more demands to respond to questions with exact numbers rather than narrative descriptions. Therefore, the ERD user prompts will often ask for a number or word to be selected from a limited list of appropriate answers, or supply a short (one or two word) answer to the software query. Because of the more demanding requirements of the ERD, the software has a much deeper level of complexity. Questions with a limited number of answers or questions that require logical (YES/NO) responses can be checked against previous answers to ensure that conflicting or mutually exclusive responses are not accepted. This built-in "expertise" adds much to the integrity of the user supplied data, and thus makes the information contained in the ERD more consistent and useful.

The ERD section of APET is a great deal larger than the SRD. There are twelve sections of the ERD, several of which taken separately would be as large as the SRD in its entirety. However, based on the requirements of the experiment, complete sections of the ERD can be eliminated. The ERD is also more technically complex than the SRD, containing far more terminology, acronyms, etc. than the other APET modules. Therefore, the use of hypertext definitions, examples, graphics and hypertext reference sections are more widely used in the ERD.

The most critical of the ERD sections is the first, which deals with the experiment's functional objectives. Each experiment contains one or more functional objectives, which in turn are composed of one or more steps. Follow-on sections in the ERD refer back to and are based on the answers given in ERD Section One. The APET software helps the user by requiring that Section One be completed before these follow-on sections, and also aids by ensuring that if a functional objective is deleted, that the follow-on sections that refer to that deleted functional objective will also be deleted. Again, this adds to the consistency of the material, and provides an easier documentation method for the user. A full description of the ERD software package is contained in the SRD/ERD User Guide (Appendix B) and the APET Hierarchy of Menu Choices for SRD/ERD Package (Appendix C).
4.3.3 Project Plan

The Project Plan is the basic planning document that describes the overall plan for proceeding with the project. Project Plans are unique to each project and the format and level of detail vary with the size, complexity, sensitivity and other characteristics of the project. The Project Plan will cover the project to completion, including operational and data analysis periods. The Microgravity Science and Applications Division (MSAD) requires that a MSAD Project Plan be submitted and approved prior to making a major commitment of resources to an MSAD project. MSAD Project Plans are to be prepared in final draft form for the Requirements Definition Review.

Plans will be prepared and submitted for all flight experiments. Project plans will be reissued, modified, or amended for reflights depending on the complexity of the task. A plan's preparation is the responsibility of the designated Project Manager at the responsible NASA center. The Project Manager will sign the MSAD Project Plan as the preparer; the Project Scientist and the Principal Investigator will sign as concurring. The MSAD Project Plan will be signed off at the NASA center prior to submission to Headquarters by the appropriate center's authorities. When the Program Scientist and Program Manager sign to register their concurrence, the MSAD Project Plan will be submitted to the MSAD Director for approval.

The Project Manager is responsible for updating a MSAD Project Plan when significant changes occur (such as changes in scope, organization, or roles and responsibilities). This does not apply to resources, schedules or manpower, which are updated through normal budgeting and project monitoring activities. The Project Manager will establish a change control process for maintaining the MSAD Project Plan and other project documentation.

The Project Plan component of APET is similar to the SRD component, in that it is comprised primarily of text responses to a series of predefined questions. There are a number of options available to the user to make the documentation process more efficient. For a full description of the Project Plan software package, see the Project Plan User's Guide (Appendix D).

4.3.4 Science Requirements Envelope Document

The Science Requirements Envelope Document provides an envelope or volume of science requirements for a type of experimentation which is intended to encompass the science requirements generated by individual experiments of that type. The primary purpose of the document is to provide science requirements against which hardware can be conceptualized such that later, when specific PIs are chosen, their individual requirements will fall within the requirements originally stated in the Science Requirements Envelope Document.

The Science Requirements Envelope Document is very similar to the SRD. The primary difference is not the questions, but in the user responses, where a range of value is given for a capacity rather than a discreet measurement for an experiment. To complete the Science Requirements Envelope Document, questions were taken directly from the MSAD Management Plan. For a detailed description of the Science Requirements Envelope Document, see Appendix E.
4.3.5 Safety Requirements Document

The Safety Requirements Document designates the safety-related activities and documentation required of individual Payload Element Developers (PEDs). This document is applicable to all MSFC Payload Project Office managed STS attached payload missions and to all of the PEDs for those missions. STS attached payloads include Spacelab dedicated missions, middeck payloads, and partial-payload missions. A partial-payload mission is a flight that is not a Spacelab-dedicated (unique) mission and is shared with other payloads. Such missions are also referred to as mixed cargo missions. Partial payloads are defined as those payloads that do not require a Spacelab module or the Space-lab igloo.

After analysis of the Safety Requirements Document, it was determined that the UAH effort would concentrate on the preparation of the Materials Usage Agreement (MUA). The document addresses the materials to be used with the experiment, and specifies which are considered hazardous and require special safety concerns. Hazardous materials reside in a NASA-central database which is accessible via modem. At the time of this writing, software is being developed to incorporate this database into an APET module to be used in the generation of the MUA section of the Safety Requirements Document.

5.0 PRELIMINARY RESULTS OF SOFTWARE DISTRIBUTION

To solicit inputs about the APET software package, presentations and demonstrations were conducted to recently selected PIs at Marshall Space Flight Center, Huntsville, AL; Lewis Research Center in Cleveland, OH, and at NASA headquarters in Washington, D.C., where instruction manuals and system software were distributed. The primary emphasis was on the validation of the SRD package, and limited emphasis was on the ERD package. Emphasis was placed on the SRD because it is the first document to be completed by the PI and is due 12 - 36 months before the ERD. In addition, two packages for the generation of the Project Plan were distributed. No packages addressing the Science Requirements Envelope Document have been dispensed.

The preliminary results of these distributions have been favorable. The first SRD software package was distributed on diskette only, with no supporting documentation. Even so, the user was able to generate an acceptable SRD with minimal instruction from the UAH APET development staff. Comments from this and other early users of the system have found it user friendly, and an aid in meeting the documentation requirements. Users have been complimentary about the ease of data retrieval.

Although the ERD component of APET has been distributed, none of the early PI participants have completed any ERD sections. In addition, two packages for the generation of the Project Plan were distributed. Likewise, the software distribution addressing the Project Plan has not yielded any conclusive results. No packages addressing the Science Requirements Envelope Document have been dispensed. Recipients of the software have been impressed with the work NASA and MSFC have put into this effort, and all have agreed
that the research completed is a valuable and needed first step in automating the documentation process.

6.0 LIMITATIONS OF DELIVERED SOFTWARE

The most valuable comments made about the APET software are not the compliments, but the criticisms. Without the customer inputs of what is still required in the system, it would be difficult to determine the improvements necessary to make it a valuable tool for the PI. The following paragraphs represent the most common suggestions on improving the APET tool.

The APET software was designed to run on any IBM-compatible personal computer (PC) using the DOS operating system. This requirement thus restricts users of the Macintosh line of computers and their associated operating system. The determination to build for the PC and not the Macintosh was made primarily because little application software existed for the Macintosh that would offer the same support as Knowledge Pro offered for the PC. However, because the NASA PI community's alliances seem to be evenly split between PC and Macintosh, it is a reoccurring suggestion that the Macintosh be supported. Efforts to convert the existing PC code to a form that can be executed on the Macintosh are currently being investigated, and will be implemented as appropriate.

The second common suggestion is that the APET editor be improved to include a spell checker function. This was realized to be a shortcoming of Knowledge Pro from the outset of the software development project. The exclusion of a spell checker function is primarily attributable to the lack of random access memory (RAM) of the current generation of personal computers. RAM is required for both APET, Knowledge Pro, and the Knowledge Pro editor. The addition of an internal spell checker would increase the requirements of RAM to the point of system failure. The proposed solution for the lack of a spell checker is to include an external spell checker that can be called by APET. In order to add this feature, a spell checker software package that is inexpensive and free to distribute must be found. Attempts are being made to find such a package at this time.

Most other suggestions about the APET software are not necessarily criticisms of the package but of the documentation process. For instance, PIs preparing to fill out the SRD commented that many of the questions asked did not pertain to the objective of the SRD. In those cases, it was explained that all questions came from the MSAD Management Plan. However, there has been enough commentary generated to justify that this is a valid concern with the PI community. The SRD should be examined for what information is needed to meet its objective, and eliminate any irrelevant questions that may exist.

7.0 CONCLUSIONS AND RECOMMENDATIONS

The APET package has been developed and distributed to PIs required to submit SRDs and ERDs to justify/define their experiments. Additionally, software has been developed and is currently being evaluated to fulfill the
documentation requirements for Project Plans and Science Requirements Envelope Documents. Initial software efforts have begun on the Safety Requirements Document, with the prototype design being developed. Based on the preliminary comments of the PIs who have taken part in the distribution of the APET software, APET fills a need to automate the documentation process. However, more work needs to be done to enhance the APET system.

There should be a Macintosh version of APET available. There are a number of PIs who wish to use the software, but are unhappy with the PC-only restriction. The editor of the APET software should also be enhanced to include additional features, such as a spell checker. This recommendation may be delayed, however, until machines with a greater RAM are available on the market.

It is also recommended that the findings of this research be used to examine the documentation requirements placed on the PI. There are instructions in the NASA-supplied hardcopy documentation that are vague and have little or no relationship to the true objective of the master document. These instructions also offer little information as to the amount of detail required to adequately answer the question. These questions cause problems for the PI, and add unneeded complexity to the overall task.

Once a number of SRDs have been created using APET, it is suggested that the software be modified to include examples of what is expected from the PI. This could further be developed to provide the PI a model that could be copied into his answer, then customized to his individual response.

Further work should be conducted to complete the Safety Requirements Document's Material Usage Listing. This will be of great benefit to the PI, who is unlikely to be aware of the dynamic nature of the hazardous materials database. This work, along with the completed validation/distribution of the Project Plan and the Science Requirements Envelope Document, will provide a solid baseline from which NASA can move from a paper environment to an electronic environment.
SRD Logical Flow Chart

1. Ask Description of Experiment

   2. Ask Do you wish to continue with the next question?
      NO
      YES

   3. Ask Scientific Knowledge to be Gained

   4. Ask Do you wish to continue with the next question?
      NO
      YES

   5. Ask Value of Knowledge to Scientific Field

   6. Ask Do you wish to continue with the next question?
      NO
      YES

Page 1
Ask
Limitations of Ground-Based Testing

Ask
Do you wish to continue with the next question?

YES

Ask
Limitations of Drop Towers

Ask
Do you wish to continue with the next question?

NO

YES

Ask
Limitations of Testing in Aircraft

Ask
Do you wish to continue with the next question?

NO

YES

Ask
Need for Accommodations in the Shuttle
Ask Experiment Sample Requirements

Ask Do you wish to continue with the next question?

YES

Ask Atmospheric Requirements

Ask Do you wish to continue with the next question?

YES

Ask Do you have Atmospheric Requirements?

YES

Ask Atmospheric Requirements (Pressure)

Ask Do you wish to continue with the next question?

YES

Page 8
Ask
Atmospheric Requirements (Gas Composition)

Ask
Do you wish to continue with the next question?

YES

Ask
Atmospheric Requirements (Humidity)

Ask
Do you wish to continue with the next question?

YES

Ask
Atmospheric Requirements (Vacuum)

Ask
Do you wish to continue with the next question?

YES

Ask
Temperature Control and Measurement

NO
Ask Telepresence

Ask Do you wish to continue with the next question?

NO

YES

Ask Telerobotics

Ask Do you wish to continue with the next question?

NO

YES

Ask Research Equipment (Preflight)

Ask Do you wish to continue with the next question?

NO

YES

Ask Research Equipment (Postflight)
Ask Services (Film Development)

Ask Do you wish to continue with the next question?

If NO, return to previous menu.

If YES, ask Services (Software Development)

Ask Do you wish to continue with the next question?

If NO, return to previous menu.

If YES, ask Other Requirements
APPENDIX B

SRD/ERD User Guide
# TABLE OF CONTENTS

## 1.0 INTRODUCTION ........................................... 1

## 2.0 DISCUSSION ............................................... 1
2.1 Background ................................................. 1
2.2 System Requirements ....................................... 1
2.3 Installation ............................................... 2
2.4 Getting Started .......................................... 3

## 3.0 USING THE APET SYSTEM ................................. 6
3.1 How to Use the System .................................... 6
3.2 Project Selection ......................................... 6
3.3 SRD Overview and Explanation ......................... 8
3.4 ERD Overview and Explanation .......................... 8
3.5 SRD Documentation Cross-Reference .................. 8
3.6 ERD Documentation Cross-Reference .................. 8
3.7 Glossary/Acronyms ........................................ 8
3.8 Print Glossary/Acronyms .................................. 9
3.9 Filling Out the SRD ....................................... 9
3.10 Entering Project Initialization Information ........ 9
3.11 Complete Science Requirements Document .......... 13
3.12 Printing the SRD ......................................... 15
3.13 Displaying the SRD ...................................... 15
3.14 Create an ASCII File of SRD ........................... 16
3.15 Filling Out the ERD ..................................... 16
3.16 Complete Engineering Requirements Document ....... 17
3.17 Printing the ERD ......................................... 17
3.18 Displaying the ERD ...................................... 18
3.19 Baselining a Document .................................. 24
3.20 Comparing a Baseline to the Current Revision ...... 24
3.21 Copying Answers to Disk ............................... 24

## 4.0 HELPFUL HINTS ........................................... 26
LIST OF FIGURES

Figure 1 ........ Sample Screen Layout Using Hypertext

Figure 2 ........ Opening APET Main Menu

Figure 3 ........ Project Selection/Identification Menu

Figure 4 ........ SRD/ERD Activity Menu

Figure 5 ........ Fill Out SRD/ERD Documentation Menu

Figure 6 ........ Sample SRD Question Screen

Figure 7 ........ ERD List of Sections

Figure 8 ........ ERD Section Selection

Figure 9 ........ ERD Subtopic Selection

Figure 10 ...... ERD Topic Narrative

Figure 11 ...... Sample ERD Question

Figure 12 ...... Baseline Menu
1.0 INTRODUCTION

So you want to fly an experiment on the Shuttle.

Well, to begin the process, we must get a little information about your experiment and its requirements.

If you have flown with us in the past, you may remember a substantial amount of paper documentation was required. This application, the Automated Payload Experiment Tool, is designed to alleviate much of the burden of the document preparation and maintenance process. This system can currently be used to prepare two support documents, the Science Requirements Document (SRD), which defines the science objectives, and the Engineering Requirements Document (ERD), which defines the engineering design/build requirements. The version that you have is for the creation of both documents.

2.0 DISCUSSION

2.1 Background

The Microgravity Experiment Projects (MEP) Office of the Payload Projects Office (PPO) at the Marshall Space Flight Center (MSFC) is currently responsible for collecting and coordinating experiment/facility specifications and requirements between NASA and various colleges, universities, research centers, and other public- and private-sector organizations that are selected or are requesting to fly their respective microgravity experiments on designated flights. This coordination involves the communication of flight hardware requirements and the preparation and review of all documentation between NASA and the research groups. To reduce difficulties encountered by these customers of NASA, an effort was undertaken to research, analyze, and evaluate the current procedures involved in the information gathering activities.

The MEP Office identified a need to develop an Automated Payload Experiment Tool (APET) which would lead experiment developers through the development planning process, obtain necessary information, establish an electronic data exchange avenue and allow easy manipulation/reformatting of the collected information. In order to fulfill this need, the University of Alabama in Huntsville (UAH) was tasked to design and develop the APET software package to meet the increasing demands to lighten the burden of documentation preparation and maintenance for NASA and its customers.

2.2 System Requirements

The objective of APET is to provide an easy to use tool to the Principal Investigator (PI) team. To ensure ease of use, few computer hardware requirements are necessary to operate the APET software package.

APET is designed to run on any IBM-PC compatible personal computer. While it is possible to use the system on a 386 PC, it is recommended that the APET user install the software on a 486 PC or higher. The graphical displays,
multiple screen windows, and the complexity of the system cause noticeable slowdowns on any machine less than the 486.

The software requires that the PC be equipped with a hard disk drive. For proper execution, the hard drive (or some partition of it) must be named C:. The SRD/ERD version of APET will require approximately 14M (megabytes) of space on the hard drive for the system, plus another 1M on the hard drive for the data files created by the user. However, for optimal performance, the hard drive should have a total of at least 17M free upon installation of the software.

For ease of use, the PC should be equipped with a mouse. This, however, is not mandatory. APET utilizes the hypertext technology, which offers a point-and-click user interface. Instead of a mouse, the user does have the option of pressing selected function keys to achieve the same effect.

2.3 Installation

The APET software package is provided on four high density diskettes. The files stored on these diskettes have been compressed; therefore, it is required that the user follow several simple steps to ensure correct installation.

1) Insert the diskette marked as "APET SRD/ERD DISK 1" in the drive designated as A:. If the A: drive on your system is not the correct size, then use the DOS ASSIGN command to redesignate the drives appropriately. (For example, if you have 3 1/2" disks but your 3 1/2" drive is B:, then at the DOS prompt type ASSIGN A: B:.)

2) From this drive (A:) type:

   INSTALL.

   This will activate the installation routine. A series of instructions and informational text will be presented. Each screen will advise what is transpiring in the installation procedure. The installation routine will create a subdirectory on the C: drive called GARDEN. Once created, the files contained on the installation disks will be copied to the directory C:\GARDEN. Most of these files have been compressed to conserve disk space. An uncompress routine will be invoked to return these files to their normal (and usable) condition. As the installation routine is completed for each disk, the user will be advised to insert the next diskette. To cancel the installation at any time, press the CTRL (control) C keys.

3) Upon successful installation of the APET program files, the message INSTALLATION ROUTINE COMPLETE will be displayed. The APET application, running under the direction of Knowledge Pro software, will be entered and you will be presented the opening menu. All subsequent sessions using the APET software may be initiated by going to the C:\GARDEN subdirectory and typing FLY.
2.4 **Getting Started**

This application uses hypertext technology. Hypertext software systems allow for the retrieval of related information at the point and click of a mouse or, if a mouse is not used, at the touch of one or two keystrokes. For information on a highlighted topic, just move the mouse to that word and click. A window will be opened, overlaying the current window. In the new window, the information will immediately be displayed. Once this support information has been reviewed, press SPACE (or ESC) once to close the window and return to your original screen. If you are not using a mouse, please use the F3 and F4 function keys (marked Select and View) as indicated at the bottom of the screen. (See Figure 1). The F3 key allows you to select the different hypertext topics. Once the desired topic is selected (i.e. highlighted), the F4 key calls the background information for view.

Multiple page displays are indicated by the **Page 1 of 2** message at the lower right of the screen. To navigate through multiple screen displays, please use the **Page Up** and **Page Down** keys to scroll either forward or backward through the pages.

For help at anytime throughout the APET application, press the F1 key. This will retrieve location sensitive help information, and may be called from the system or system-called edit screens. This will be the method by which assistance information will be retrieved throughout this application.

APET has been designed as a menu-driven software package. This means that any function required of the user can be activated via a menu option. This includes exiting the system. It is strongly recommended that the user always "back out" of the application by using the appropriate menu options, i.e. "Return to Previous Menu". An option does exist to exit from any point in the application by selecting F10. It is not recommended that this be used from inside a question/answer section of the application. The F10 command causes an immediate exit from the program, without checking to ensure that open files have been properly saved. Therefore, the user may experience data loss if the application is exited in this manner.

After the installation and initial use of the APET software, future systems will be initiated by going to the C:\GARDEN subdirectory and typing **FLY**. This will activate the software and present the opening menu. (See Figure 2).

Due to the hypertext capabilities of the APET software, a large amount of RAM (random access memory) is required. Because of the heavy RAM demand, proper execution of the software requires no other software package be running simultaneously with the APET software. Whenever the available RAM becomes too little for the application, an "Insufficient Memory" message will be shown at the bottom right of the screen. To alleviate this situation, simply get out of APET and reboot the system. This will usually free up all available RAM and ensure proper execution. (See Helpful Hints for further instruction.)
Automated Payload Experiment Tool

The primary purposes of the Science Requirements Document are:
(1) to provide adequate justification for conducting the experiment in space.
(2) to delineate and justify the science requirements that the experiment places on the hardware.

The Outline for the Science Requirements Document:

1.0 Introduction/Summary
2.0 Background
3.0 Justification for Conducting the Experiment in Space
4.0 Experiment Details

Figure 1
Sample Screen Layout Using Hypertext
Automated Payload Experiment Tool

Please select the activity of your choice, or choose Exit to leave the system.

How to use the System
Project Selection
SRD Overview and Explanation
SRD Documentation Cross-Reference
ERD Overview and Explanation
ERD Documentation Cross-Reference
Glossary/Acronyms
Print Glossary/Acronyms
Exit System

F1 Help

F8 DOS F10 Quit

Figure 2
Opening APET Main Menu
3.0 USING THE APET SYSTEM

3.1 How to Use the System

Because the use of a hypertext tool may be a new experience, a brief on line tutorial is provided with APET. To use this tool, please choose option one on the opening menu entitled "How to Use the System". To select this option, point with the mouse to the phrase and click. If not using a mouse, use the arrow keys to highlight the option and press RETURN. You can tell when an option has been selected because it (the phrase or word) will be highlighted differently from all other options. As the mouse is moved to other options, each in turn will be highlighted.

Once the "How to Use the System" option has been selected, a different screen will be presented with a brief overview of hypertext and the methods of selecting topics. (This overview is much the same as appears in Section 2.4 of this user's guide.) Practice selecting topics and moving from one screen to another using either the mouse or the appropriate keyboard function keys.

3.2 Project Selection

The APET software package will accommodate one or more experiments for the user. However, each experiment must be identified by a short (8 characters or less) name, which must conform to the naming convention used by the DOS computer operating system. Briefly, these rules state that a DOS name cannot be over 8 characters in length, and must contain a combination of either letters, numbers, or the underscore (_) character. Any other special keys, including the SPACE, are prohibited. The rationale behind this naming convention is to allow storage of data files for each experiment in a subdirectory for that specific experiment. For example, if a user is working on two experiments, identified as THINFILM and HIPROTEK, then there would be a subdirectory for each. The configuration of these files would be as follows:

<table>
<thead>
<tr>
<th>Root Directory</th>
<th>Application Directory</th>
<th>Experiment Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:...............</td>
<td>GARDEN...............</td>
<td>THINFILM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HIPROTEK</td>
</tr>
</tbody>
</table>

Therefore, all data files containing answers for the APET questions for the experiment THINFILM would appear in the subdirectory THINFILM. If additional experiments are required, the user would identify the new experiment and an additional subdirectory would be added.

Figure 3 shows the menu for selecting, adding, or deleting an experiment project. In the example, the experiment AADSFL has been previously defined by the user. If the user wants to work on this experiment, he simply points and clicks on this selection. (This would be the case in a majority of the cases, since most Principal Investigators will have only one active experiment at any given point). However, if another experiment is required, the user would click on "ENTER A NEW PROJECT". The user would be prompted to identify the new experiment, and would immediately be passed into the APET system, where information regarding that experiment would be addressed.
Figure 3
Project Selection/Identification Menu
If a user wishes to delete an experiment, along with all its associated data files, he may do so by selecting the "DELETE AN OLD PROJECT" option from the menu. However, there is no recoverable procedure to undelete a project. Therefore, the user is strongly advised to use this procedure with caution.

3.3 SRD Overview and Explanation

The third selection from the APET Main Menu is the "SRD Overview and Explanation". This option should be selected when the user wishes to see an overview of the SRD document, along with brief explanations of the information to be covered in each section/subsection of the document. For an additional overview of the topics to be addressed in the SRD, see Appendix A of this document.

3.4 ERD Overview and Explanation

The fourth selection from the APET Main Menu is the "ERD Overview and Explanation". This option should be selected when the user wishes to see an overview of the ERD, along with brief explanations of the information to be covered in each section/subsection of the document. For an overview of the topics to be addressed in the ERD, see Appendix B of this document.

3.5 SRD Documentation Cross-Reference

The fifth selection from the APET Main Menu is the "SRD Documentation Cross-Reference". Selection of this option allows the user to more fully understand the interrelationships between the Science Requirements Document (SRD) and the other documentation required by NASA. The SRD has sections that reference information contained in other sections of the SRD as well as other documents. Only the highlighted topics are further referenced.

3.6 ERD Documentation Cross-Reference

The sixth selection from the APET Main Menu is the "ERD Documentation Cross-Reference". Selection of this option allows the user to more fully understand the interrelationships between the Engineering Requirements Document (ERD) and the other documentation required by NASA. The ERD has sections that reference information contained in other sections of the ERD as well as other documents. Only the highlighted topics are further referenced.

3.7 Glossary/Acronyms

A number of NASA specific terms, definitions, and acronyms will appear as support material throughout the documentation process. One of the primary advantages of using a hypertext-based tool is to allow for easy and immediate retrieval of these terms.

Option number seven from the APET Main Menu allows the user to retrieve a listing of these terms, and presents them in a form analogous to a glossary in
a book. To view a definition, highlight the desired term and click. A term can be highlighted by using the mouse to move the cursor to that word, or by using the F3 key for selection. To view the definition, the user should either click the mouse or press the F4 key. The definition of that word/term will be presented. Should the definition contain a term that requires further description, highlight that word and click. The new definition will overlay the previous definition. This method can be repeated as long as further definitions exist and the memory capacity of the machine is not exceeded.

Please note that the glossary consists of multiple pages. Remember to navigate through the multi-page displays by using either the Page Up/Page Down function keys.

3.8 Print Glossary/Acronyms

Option number eight from the APET Main Menu activates a routine for the printing of the glossary/acronym list, as discussed in 3.7. Because the output of this selection will be a multi-page document, the use of this option will be rare.

3.9 Filling Out the SRD

To fill out the SRD, the user must first select "Project Selection" from the APET Main Menu. Then the Project Selection/Identification Menu (Figure 3) will be presented for the user to identify either a new project or select an existing project. Once the selection/identification has been made, the SRD/ERD Activity Menu (Figure 4) will be presented. Please note that the selected project is shown on the upper right corner of the screen. Please be sure that the project shown is the one you wish to work.

The SRD/ERD Activity Menu presents eleven options for the user. The most significant of these is the first: "FILL OUT DOCUMENTATION". The selection of this option will present the Fill Out SRD/ERD Documentation Menu (Figure 5). This menu offers the user three options: "ENTER PROJECT INITIALIZATION INFORMATION", "COMPLETE SCIENCE REQUIREMENTS DOCUMENT (SRD)", and "COMPLETE ENGINEERING REQUIREMENTS DOCUMENT (ERD)".

3.10 Entering Project Initialization Information

Under most circumstances, the first information entered by the user into the APET system is the project initialization information. To enter this information, selection one from the Fill Out SRD/ERD Documentation Menu is chosen. This information is used to identify certain aspects of the experiment, and will be used throughout the documentation process. Entries include the PI name, his organization, address, city, state, zip, phone, and experiment title. This information will be provided in the form of type written responses to user prompts. For example, the user will be asked:

Please enter your first and last names, i.e. Dr. John Doe.
The user should respond with a one line response. (If there is a limitation on the length of this one line response, the screen display will provide an instruction, such as "Please limit your response to 16 characters.") This will be the format of user entries throughout the APET application. When the user is prompted to make an entry, the response should be on one line. When the user responds with a RETURN, the answer is stored and the next question, if one exists, is asked. Once all questions for that segment have been answered, the answers are written to a data file.

In the "Project Initialization Information" subsection, the only variation in the user prompt/one line response routine is with the experiment title. Because experiment titles can be several lines, the user is given a prompt and immediately sent to the APET editor. This editor is a small version of a word processor, with many of the functions of a common word processing package. The commands used in the APET editor are similar to those in the software package WORDSTAR. To see the commands available, press the F1 function key from inside the editor. A separate window will be opened and will overlay the current screen. From there, page down until the function you wish to perform is shown. The one-to-two keystroke command to accomplish the task will be shown. (Note: The symbol (') represents the CONTROL key, i.e. KW means to hold the CONTROL key down while pressing the letter K, then press the letter W.) When the user is finished entering the answer into the editor, he may exit by pressing the ESC (escape) key. A message will then appear on the screen that tells the user what file name is being saved. Respond with a RETURN to accept this name and save the answer, or an ESC to cancel the answer.

Important: Do not change the file name when the APET software asks if the name is acceptable. Future sessions that allow the user to change, delete, print, or display the SRD answers require that these file names be used. Changing the file name will make the file either inaccessible or inappropriate. Therefore, always accept the file name as given.
The project you have selected is: AADSF_L
Please enter your choice of activities from the list.

FILL OUT DOCUMENTATION
PRINT SRD
DISPLAY SRD
CREATE AN ASCII FILE OF SRD
PRINT ERD
DISPLAY ERD
BASELINE DOCUMENT
COMPARE BASELINE TO CURRENT REVISION
COPY ANSWERS TO DISK
RETURN TO PREVIOUS MENU
RETURN TO MAIN MENU
QUIT

F1 Help
F8 DOS  F10 Quit

Figure 4
SRD/ERD Activity Menu
Automated Payload Experiment Tool

Please select the activity you wish to perform on the AADSF_L project.

ENTER PROJECT INITIALIZATION INFORMATION
COMPLETE SCIENCE REQUIREMENTS DOCUMENT (SRD)
COMPLETE ENGINEERING REQUIREMENTS DOCUMENT (ERD)
RETURN TO PREVIOUS MENU
RETURN TO MAIN MENU
EXIT SYSTEM

Figure 5
Fill Out SRD/ERD Documentation Menu
3.11 Complete Science Requirements Document

To complete the Science Requirements Document, there are usually between 50 - 60 questions that must be answered. These answers will be in the form of short narratives, consisting of one or more paragraphs of text. Each question or user prompt will invoke the APET editor and give the user sufficient space to write as much (or little) as required. (For an outline of the topics to be addressed, see Appendix A.) A sample question from the SRD is shown in Figure 6.

The Complete SRD session begins by asking the user if he has begun to fill out the SRD previously. A 'NO' response causes the questions to be asked in sequence. A 'YES' response results in the question topics to be displayed in a list. If questions are to be answered from a list, a list will appear as a window that overlays the question screen. The user is expected to point-and-click on the appropriate topic. (If not using a mouse, use the arrow keys to select and press RETURN.) The user should click on the appropriate answer with the left-side mouse button.

For the initial SRD session, the user would respond with a NO and proceed to the questions. These questions will be asked in the same sequence as is shown in the outline. After each response, the user will be asked if he wants to continue to the next question. This gives the user a chance to end the session when he desires, rather than advancing through all the remaining questions. The title of each question screen will include the number of the question (i.e., Number 1 of 52). This allows the user to see where he is in the process and act accordingly.

If the user responds with a 'YES', which means there has been a previous session, the following question will appear:

Do you wish to change only one item, resume at a point and continue sequentially through the remainder of the SRD, or complete all topics previously unanswered?

This allows the user one of three options. 1) He may select the one answer that needs changing, go directly to that answer and change it, then record that answer to disk. 2) He can select the topic where he would like to resume his activities, answer that question, record the answer, and go to the next question in sequence. This gives the user the capability of selecting the 20th question, and proceed sequentially through the remaining 32 questions. 3) The user can complete all questions that have not yet been answered. This option will invoke a command to look at what answers (files) do not exist, and build a list of these topics. The user then selects the topic to answer, answers the question, records the answer, and goes to the next question of his choice. With each recorded answer, that topic is removed from the list.
Description of experiment (Question 1 of 52)

Please enter a narrative description of the experiment. This topic is also addressed under the heading "Experiment Procedures to be Used."

Press the RETURN KEY to enter the editor, ESC to leave editor, and RETURN to confirm save

C:\GARDEN\FULLTEST\EXPDESC.DAT

The body of the text goes in this area. This is the description of the experiment.

Figure 6
Sample SRD Question Screen
3.12 Printing the SRD

The user has three methods available to generate output from the APET software. These include printing the document, displaying the document, and creating an ASCII file of the document. The APET application was designed to be flexible enough to go to a variety of printers. As with most output, the best results will be with the use of a laser printer. If a laser printer is not available, the use of a dot-matrix printer will also be acceptable. A variety, although not nearly exhaustive, of dot-matrix printers have been tested with the APET software, and all have performed well.

If the document has previously been baselined (discussed later in section 3.19), then a menu will appear giving the user the option of printing the document from the baselined version, the current revision, or neither version. If the neither option is chosen, then it is assumed that the user does not want the document printed, and the program will automatically return to the previous menu. If the baseline option is chosen, then the document will be printed from the file which is in the project baseline subdirectory. If the current revision option is chosen, then the document will be printed from the version of the document which the user is currently revising. If the document has not been baselined, then the document will be printed from the current version.

The printing of the SRD will generate the document in its entirety. An initial page eject will normally (depending on printer type) advance a blank sheet of paper before the cover sheet is printed. This will be followed by a second page advance, then page one of the document will be printed, followed by two, three, etc. through the end of the document. Because there are often graphics, tables, etc. that must be inserted within the textual document, no table of contents is printed. Because of the limitation of graphics support, it is suggested that all externally generated graphic illustrations, tables, etc. be provided in an appendix, with appropriate references throughout the document.

While the print procedure is active, a message will appear in the lower left of the screen. No other activities may take place while the document is printing. In case the printer runs out of paper, an error message will appear. Reload paper in the printer and press the SPACE key to continue.

3.13 Displaying the SRD

The second method of generating output using APET is to display sections of the SRD to the screen. The SRD is divided into seven major sections, with each divided into one or more subsections (see Appendix A). The user has the capability of selecting a section and seeing the identical output as would appear if the document was printed. Displaying the SRD is recommended to quickly review answers, especially during the development phase of document preparation.

If the document has previously been baselined (discussed later in section 3.19), then a menu will appear giving the user the option of displaying
the document from the baselined version, the current revision, or neither version. If the neither option is chosen, then it is assumed that the user does not want the document displayed, and the program will automatically return to the previous menu. If the baseline option is chosen, then the document will be displayed from the file which is in the project baseline subdirectory. If the current revision option is chosen, then the document will be displayed from the version of the document which the user is currently revising. If the document has not been baselined, then the document will be displayed from the current version.

Most SRD sections will require multiple page displays. Please note that to view the equivalent of an entire printed page, there will be at least three and usually four screen displays. Use the Page Up/Page Down method to move up or down in the document. Once a page is adequately reviewed, press the SPACE key to retrieve the next page in sequence. To abandon a display at any time, press the F10 key.

3.14 Create an ASCII File of SRD

The APET software does not have the ability to generate or insert graphics, charts, etc. that were created in some other application. This is primarily due to the memory size limitations of the computer. However, to alleviate this limitation, APET does have the ability to generate an ASCII file of its SRD output. After choosing this option, the user need only type in the full file name (includes drive, file name, and extension). The file will then be created as a replica of the printed output.

The benefit of creating an ASCII text file of the SRD is in providing the user with the capability of enhancing the final printing by inserting graphics, photos, tables, equations, or other difficult to create figures. In addition, different fonts, font sizes, and special effects can be used to dress up the final printed output.

3.15 Filling Out the ERD

To fill out the ERD, the user must first select "Project Selection" from the APET Main Menu. Upon this action, the Project Selection/Identification Menu (Figure 3) will be presented for the user to identify either a new project or select an existing project. Once the selection/identification has been made, the SRD/ERD Activity Menu (Figure 4) will be presented. Please note that the selected project is shown on the upper right corner of the screen. Please be sure that the project shown is the one you wish to work.

The SRD/ERD Activity Menu presents eleven options for the user. The most significant of these is the first: "FILL OUT DOCUMENTATION". The selection of this option will present the Fill Out SRD/ERD Documentation Menu (Figure 5). This menu offers the user three options: "ENTER PROJECT INITIALIZATION INFORMATION", "COMPLETE SCIENCE REQUIREMENTS DOCUMENT (SRD)" and "COMPLETE ENGINEERING REQUIREMENTS DOCUMENT (ERD)". Select "COMPLETE ENGINEERING REQUIREMENTS DOCUMENT (ERD)" if the project initialization information has already been entered. If this information has not already been entered, select it before generating the ERD.
3.16 Complete Engineering Requirements Document

To complete the Engineering Requirements Document, the user must answer a series of questions about the experiment. The questions are grouped into topics, a list of which is presented to the user. The user is asked to select the topic which is to be displayed. The chosen topic will be displayed along with the subtopics covered within that section. The user is prompted as to whether or not he wishes to begin or continue filling out the questions in that section. (For an outline of the topics to be addressed, see Appendix B.) If the user chooses to fill out the section, he will be prompted to select one of the subheadings. Upon subheading selection, a brief description of the topic will be given, along with any necessary instructions for answering the question.

If the question invokes the editor, the answers should be given in the form of short narratives, consisting of one or more paragraphs of text. Sufficient space will be given to write as much (or little) as required. If an answer requires a numeric response, enter the number just as you wish it to appear. A sample of this procedure is shown in Figures 7 thru 11.

Upon selection of a topic, if the questions within that chosen topic have already been answered, the user will be prompted with a message saying that the section has already been completed. The user then has the options of ADDING, EDITING, or DELETING answers, or RETURNING to the previous menu. The user should select an option and follow the instructions accordingly. If the chosen topic does not have any related subtopics, and the user wishes to complete that section, the program will give instructions for answering that question, and the APET editor will be invoked as needed.

Some of the topics have accompanying illustration(s) in order to give the user a better understanding of what information is needed. If you wish to view the illustration, simply click on, or select, the appropriate phrase. This will cause the screen to momentarily go blank, and the illustration will then be presented. After you have viewed the illustration, simply press SPACE and the program will return you to your original screen.

3.17 Printing the ERD

The user has two methods available to generate output from the APET software in regard to the ERD. These include printing the document, and displaying the document. The APET application was designed to be flexible enough to go to a variety of printers. As with most output, the best results will be with the use of a laser printer. If a laser printer is not available, the use of a dot-matrix printer will also be acceptable. A variety, although not nearly exhaustive, of dot-matrix printers have been tested with the APET software, and all have performed well.

If the document has previously been baselined (discussed later in section 3.19), then a menu will appear giving the user the option of printing the document from the baselined version, the current revision, or neither version. If the neither option is chosen, then it is assumed that the user does not
want the document printed, and the program will automatically return to the previous menu. If the baseline option is chosen, then the document will be printed from the file which is in the project baseline subdirectory. If the current revision option is chosen, then the document will be printed from the version of the document which the user is currently revising. If the document has not been baselined, then the document will be printed from the current version.

The printing of the ERD must be accomplished by printing either selected sections or tables of the document. An initial page eject will normally (depending on printer type) advance a blank sheet of paper before the first sheet is printed. This will be followed by a second page advance, then page one of the document will be printed, followed by two, three, etc. through the end of the document. Because there are often graphics, tables, etc. that must be inserted within the textual document, no table of contents is printed. Because of the limitation of graphics support, it is suggested that all externally generated graphic illustrations, tables, etc. be provided in an appendix, with appropriate references throughout the document.

While the print procedure is active, a message will appear in the lower left of the screen. No other activities may take place while the document is printing. In case the printer runs out of paper, an error message will appear. Reload paper in the printer and press the SPACE key to continue.

3.18 Displaying the ERD

The second method of generating ERD output using APET is to display sections of the ERD to the screen. The ERD is divided into twelve major sections, some of which may be divided into more subsections (see Appendix B). The user has the capability of selecting a section and seeing the identical output as would appear if the document was printed. Displaying the ERD is recommended to quickly review answers, especially during the development phase of document preparation.

If the document has previously been baselined (discussed later in section 3.19), then a menu will appear giving the user the option of displaying the document from the baselined version, the current revision, or neither version. If the neither option is chosen, then it is assumed that the user does not want the document displayed, and the program will automatically return to the previous menu. If the baseline option is chosen, then the document will be displayed from the file which is in the project baseline subdirectory. If the current revision option is chosen, then the document will be displayed from the version of the document which the user is currently revising. If the document has not been baselined, then the document will be displayed from the current version.

Most ERD sections will require multiple page displays. Please note that to view the equivalent of an entire printed page, there will be at least three and usually four screen displays. Use the point-and-click (or Page Up/Page Down) method to move up or down in the document. Once a page is adequately reviewed, press the SPACE key to retrieve the next page in sequence. Displays will continue until all output has been presented.
The suggested outline for the Engineering Requirements Document (ERD) is as follows. Please choose the Section with which you would like to begin/resume:

<table>
<thead>
<tr>
<th></th>
<th>Functional Objectives &amp; Equipment Identification</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Structural/Mechanical</td>
</tr>
<tr>
<td>3</td>
<td>Pointing/Stabilization and Alignment</td>
</tr>
<tr>
<td>4</td>
<td>Orbital Requirements and Constraints</td>
</tr>
<tr>
<td>5</td>
<td>Electrical Requirements</td>
</tr>
<tr>
<td>6</td>
<td>Thermal Control/Fluid Requirements</td>
</tr>
<tr>
<td>7</td>
<td>Data System Requirements</td>
</tr>
<tr>
<td>8</td>
<td>Flight Software Requirements</td>
</tr>
<tr>
<td>9</td>
<td>Physical Integration</td>
</tr>
<tr>
<td>10</td>
<td>Mission Operations Support</td>
</tr>
<tr>
<td>11</td>
<td>Training Objectives</td>
</tr>
<tr>
<td>12</td>
<td>Environmental Contamination Data Requirements</td>
</tr>
</tbody>
</table>

Return to Previous Menu

F1 Help  F3 Select

F4 View  F8 DOS  F10 Quit

Figure 7
ERD List of Sections
Automated Payload Experiment Tool

1.0 Functional Objectives And Equipment Identification

1.1 Functional Objectives
1.2 Equipment Identification
1.3 Operational Function Flow

Do you wish to begin/continue filling out this section.

YES  NO

F1 Help   F3 Select
F4 View   F8 DOS   F10 Quit

Figure 8
ERD Section Selection
With which subheading do you wish to begin?

1.1 Functional Objectives
1.2 Equipment Identification
1.3 Operational Function Flow
Quit

Figure 9
ERD Subtopic Selection
1.0 Functional Objectives and Equipment Identification

A definition of the experiment objectives and identification of the payload element equipment items needed to accomplish these objectives are necessary to define the support required from the STS or PMM by the instrument and the Principle Investigator.

Press SPACE to continue.
Automated Payload Experiment Tool

---

Functional Objectives

Enter the number of Functional Objectives required for this experiment.

=> 0

---

F1 Help
Enter Accept

F8 DOS
F10 Quit

---

Figure 11
Sample ERD Question
3.19 **Baselining a Document**

At some point in the documentation procedure, the SRD/ERD will be considered complete and released to external offices, agencies, organizations, etc. When this occurs, that version of the document is considered the baseline, and should be easily identified as such.

To aid in the process of maintaining separate versions of the SRD and ERD, an option exists to baseline the current version of the document. (See Figure 12) The selection of this option will cause a replica of the current version's answers (or data files) to be copied to a new subdirectory for that experiment. This new subdirectory will be called `BASELINE`. From that point, all additional editing will transpire on a new version of the answers, while the baselined version of the answers will remain intact. The generation of output will require the user to identify which version (baseline or current revision) he wishes to access.

3.20 **Comparing a Baseline to the Current Revision**

Once the document has been baselined (See Section 3.19), the user may wish to compare this baseline with the current revision. APET provides a mechanism to accomplish this task. By selecting the option "Compare Baseline to Current Revision", a DOS routine will be invoked to compare all identical data files from the current revision to the baseline document. This comparison generates a file that can then be displayed or printed, so that a quick review will show which answers have been modified since the original baseline date.

3.21 **Copying Answers to Disk**

The final output option provided by APET is the creation of files that contain all data generated by the software. This can be used as either a backup mechanism during the creation of the files, or as a means of submission of the final document instead of a hard copy/printed document. By submitting the answers on diskette, the receiving party can have direct access to the answers in the same manner as would the sender. These files are not formatted as an ASCII file, and should not be confused with the final report output, which can be created using the "Create an ASCII File of SRD" (discussed in Section 3.12).

The user will have the option of selecting either the baseline document or the current revision. After this selection, the user is asked to select the drive to receive the backup (either A:, B:, C:, or D:). A DOS copy command will then be invoked to copy all files to the selected drive.
At some point in the documentation process, it is necessary to declare that all documents are complete, and that any changes to be made will be treated as revisions to the baseline document.

Do you want to baseline your answers at this time?

YES NO

Figure 12
Baseline Menu
4.0 HELPFUL HINTS

1) Avoid the use of the F10 key to exit from within the APET application. It is a better practice to back out of the APET system through the use of the menus. By doing so, the user ensures that all answers are properly recorded to the disk drive. Use of the F10 key from within the APET application will allow the user to exit but will not automatically save information generated during the session.

2) APET does not support the insertion of externally generated graphics, tables, equations, or other non-text material. To alleviate this problem without the added labor of using a secondary word processor, it is suggested that any such material be included in an Appendix, and referenced in the text generated in APET.

3) To insert an externally generated text file into the text area in the APET editor, use the command "KR from within the editor. This is one of a variety of commands that can be used from the APET word processor. To see all available commands, press the F1 key from inside the editor and page through the instruction set.

4) The APET editor uses a word wrap routine that automatically wraps the line to the next line (a common word processing feature). It also maintains vertical alignment along the left margin. If you use indented paragraphs, please be sure that the line after the indented line begins in the column you desire. To do this, use the backspace key to move the first word in the line to the column desired. The recommended solution to this problem is not to indent paragraphs, but instead insert a blank line between each paragraph.

5) If your computer system is configured to automatically load WINDOWS or some other application package, it may be necessary to alter the AUTOEXEC.BAT file (located in the boot drive). Instructions for changing the automatic load of an application will vary by computer. One of the easier methods is to edit the AUTOEXEC.BAT file and remove the line that calls the package. For example, WINDOWS is called by the command WIN. By preventing these packages from loading, a significant amount of RAM is freed and allowed for use by APET.
APPENDIX A

SRD Topic Outline
1.0 INTRODUCTION/SUMMARY

1.1 Description of Experiment (Question 1)

1.2 Scientific Knowledge to be Gained (Question 2)

1.3 Value of Knowledge to Scientific Field (Question 3)

1.4 Justification of the Need for Space Environment (Question 4)

2.0 BACKGROUND

2.1 Description of Scientific Field (Question 5)

2.2 Current Application for Research (Question 6)

2.3 Brief Historical Account of Prior Research (Question 7)

2.4 Current Research (Question 8)

2.5 Relationship of Proposed Experiment (Question 9)

2.6 Anticipated Advance in State of the Art (Question 10)

3.0 JUSTIFICATION FOR CONDUCTING THE EXPERIMENT IN SPACE

3.1 Limitations of Ground-Based Testing (Question 11)

3.2 Limitations of Drop Towers (Question 12)

3.3 Limitations of Testing in Aircraft (Question 13)

3.4 Need for Accommodations in the Shuttle (Question 14)
3.5 Limitations of Mathematical Modeling  
(Question 15)

3.6 Limitations of Other Modeling Approaches  
(Question 16)

4.0 EXPERIMENT DETAILS

4.1 Experiment Procedures to be Used  
(Question 17)

4.2 Measurements Required  
(Question 18)

4.3 Test Plan Including Ground Characterization of Flight Hardwa  
(Question 19)

4.4 Specific Analysis Required  
(Question 20)

4.5 Preflight Experiment Planned  
(Question 21)

4.6 Post Flight Data Handling and Analysis  
(Question 22)

4.7 Mathematical Models Used  
(Question 23)

4.8 Application of Results  
(Question 24)

5.0 EXPERIMENT REQUIREMENTS

5.1 Experiment Sample Requirements  
(Question 25)

5.2 Atmospheric Requirements

5.2.1 Pressure  
(Question 26)

5.2.2 Gas Composition  
(Question 27)

5.2.3 Humidity  
(Question 28)

5.2.4 Vacuum  
(Question 29)
5.3 Temperature Control and Measurement (Question 30)

5.4 Vibration Control and Measurement (Question 31)

5.5 Test Matrix (Question 32)

5.6 Imaging Requirements

5.6.1 Photography (Question 33)

5.6.2 Radiography (Question 34)

5.6.3 Television (Question 35)

5.6.4 Resolution (Question 36)

5.6.5 Frame Rate (Question 37)

5.7 Electromagnetic Limitations (Question 38)

5.8 Astronaut Involvement

5.8.1 Extravehicular Activity (Question 39)

5.8.2 Activation of Experiment (Question 40)

5.9 Data Requirements (Question 41)

5.10 Telepresence and Telerobotics

5.10.1 Telepresence (Question 42)

5.10.2 Telerobotics (Question 43)
6.0 PRINCIPAL INVESTIGATOR’S REQUIREMENTS

6.1 Research Equipment

6.1.1 Preflight
(Question 44)

6.1.2 Postflight
(Question 45)

6.2 Apparatus Design Assistance
(Question 46)

6.3 Consultation
(Question 47)

6.4 Grants and Contracts

6.4.1 Grants
(Question 48)

6.4.2 Contracts
(Question 49)

6.5 Services

6.5.1 Film Developing
(Question 50)

6.5.2 Software Development
(Question 51)

7.0 OTHER REQUIREMENTS
(Question 52)
APPENDIX B

ERD Topic Outline
For assistance in using this software, or to offer suggestions or comments, please contact the following:

Mr. Gary Maddux
Ms. Anna Provancha
Mr. David Chattam

at (205) 895-6343,
or write

Systems Management and Production Laboratory
Research Institute
RI E-47
The University of Alabama in Huntsville
Huntsville, AL 35899
1.0 FUNCTIONAL OBJECTIVES AND EQUIPMENT IDENTIFICATION

1.1 Functional Objective Requirements Sheet

Table 1-1. Experiment Functional Objectives

2.0 STRUCTURAL/MECHANICAL

3.0 POINTING/STABILIZATION AND ALIGNMENT

3.1 Pointing Requirements
3.2 Stabilization Requirements
3.3 Viewing Requirements
3.4 IPS Pointing Requirements
3.5 Experiment Pointing Capabilities
3.6 On-Orbit Acceleration and Vibration Limits
3.7 Alignment Requirements
3.8 Coalignment Requirements

4.0 ORBITAL REQUIREMENTS AND CONSTRAINTS

4.1 Desired Orbit Characteristics
4.2 Earth and Celestial Target List and Viewing Time Requirements
4.3 Viewing Requirements and Constraints (Earth and Solar Viewing)
4.4 Viewing Requirements and Constraints (Celestial Viewing)
4.5 Vehicle Motion and g-Level Limits

5.0 ELECTRICAL REQUIREMENTS

6.0 THERMAL CONTROL/FLUID REQUIREMENTS

7.0 DATA SYSTEM REQUIREMENTS
8.0 FLIGHT SOFTWARE REQUIREMENTS
  8.1 Experiment Computer Software Requirements Summary
  8.2 Functional Description of Software Package(s)

9.0 PHYSICAL INTEGRATION

10.0 MISSION OPERATIONS SUPPORT

11.0 TRAINING OBJECTIVES
  11.1 Training Participation
  11.2 Training Objectives

12.0 ENVIRONMENTAL CONTAMINATION DATA REQUIREMENTS
  12.1 Flight Environmental Limits
  12.2 On-Orbit External Contamination Control Sensitivity
  12.3 External Contamination Sources
APPENDIX C

APET Hierarchy of Menu Choices for SRD/ERD Package
APET Hierarchy of Menu Choices for SRD/ERD Package

NASA

How to use the System  Project Selection  SRD Overview and Explanation  ERD Overview and Explanation  SRD Documentation Cross-Reference  ERD Documentation Cross-Reference  Glossary/Acronyms  Print Glossary/Acronyms  Exit System

Project Name*  ENTER A NEW PROJECT  DELETE AN OLD PROJECT  RETURN TO MAIN MENU

Fill Out Documentation  Print SRD  Display SRD  Print ERD  Display ERD  Create An ASCII File of SRD  Baseline Document  Compare Baseline to Current Revision  Copy Answers to Disk  Return to Previous menu  Return to Main Menu  Quit

These menus are only used once a baseline has been made.

Baseline  Current Revision  Neither  Baseline  Current Revision  Neither

Enter Project Initialization Information  Complete SRD  Complete ERD  Return to Previous menu  Return to Main Menu  Exit System  Run Comparison Program  Display Comparisons  Print Comparisons  Return to Previous menu

*The Project Name is entered by the user. The number of project names depends on the number of projects entered by the user.
APPENDIX D

Project Plan User Guide
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2.0 DISCUSSION</td>
<td>1</td>
</tr>
<tr>
<td>2.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>2.2 System Requirements</td>
<td>1</td>
</tr>
<tr>
<td>2.3 Installation</td>
<td>2</td>
</tr>
<tr>
<td>2.4 Getting Started</td>
<td>3</td>
</tr>
<tr>
<td>3.0 USING THE APET SYSTEM</td>
<td>6</td>
</tr>
<tr>
<td>3.1 How to Use the System</td>
<td>6</td>
</tr>
<tr>
<td>3.2 Project Selection</td>
<td>6</td>
</tr>
<tr>
<td>3.3 Project Plan Overview</td>
<td>8</td>
</tr>
<tr>
<td>3.4 Glossary/Acronyms</td>
<td>8</td>
</tr>
<tr>
<td>3.5 Print Glossary/Acronyms</td>
<td>8</td>
</tr>
<tr>
<td>3.6 Filling Out the Project Plan</td>
<td>8</td>
</tr>
<tr>
<td>3.7 Entering Project Initialization</td>
<td>9</td>
</tr>
<tr>
<td>3.8 Complete Project Plan</td>
<td>12</td>
</tr>
<tr>
<td>3.9 Printing the Project Plan</td>
<td>12</td>
</tr>
<tr>
<td>3.10 Displaying the Project Plan</td>
<td>14</td>
</tr>
<tr>
<td>3.11 Determining if Project Plan is</td>
<td>14</td>
</tr>
<tr>
<td>Complete</td>
<td></td>
</tr>
<tr>
<td>3.12 Baselineing a Document</td>
<td>15</td>
</tr>
<tr>
<td>3.13 Comparing a Baseline to the Current Revision</td>
<td>15</td>
</tr>
<tr>
<td>3.14 Copying Answers to Disk</td>
<td>15</td>
</tr>
<tr>
<td>4.0 HELPFUL HINTS</td>
<td>21</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1 ...... Sample Screen Layout Using Hypertext
Figure 2 ...... Opening APET Main Menu
Figure 3 ...... Project Selection/Identification Menu
Figure 4 ...... Project Plan Activity Menu
Figure 5 ...... Fill Out Project Plan Documentation Menu
Figure 6 ...... Sample Project Plan Question Screen
Figure 7 ...... Project Plan List of Sections
Figure 8 ...... Project Plan Section Selection
Figure 9 ...... Project Plan Subtopic Selection
Figure 10 ...... Project Plan Topic Narrative
Figure 11 ...... Baseline Menu
1.0 INTRODUCTION

So you want to fly an experiment on the Shuttle.

Well, to begin the process, we must get a little information about your experiment and its requirements.

If you have flown with us in the past, you may remember a substantial amount of paper documentation was required. This application, the Automated Payload Experiment Tool, is designed to alleviate much of the burden of the document preparation and maintenance process. This system can currently be used to prepare three support documents: the Science Requirements Document (SRD), which defines the science objectives, the Engineering Requirements Document (ERD), which defines the engineering design/build requirements, and the Project Plan. The version you have is for the creation of the Project Plan only.

2.0 DISCUSSION

2.1 Background

The Microgravity Experiment Projects (MEP) Office of the Payload Projects Office (PPO) at the Marshall Space Flight Center (MSFC) is currently responsible for collecting and coordinating experiment/facility specifications and requirements between NASA and various colleges, universities, research centers, and other public- and private-sector organizations that are selected or are requesting to fly their respective microgravity experiments on designated flights. This coordination involves the communication of flight hardware requirements and the preparation and review of all documentation between NASA and the research groups. To reduce difficulties encountered by these customers of NASA, an effort was undertaken to research, analyze, and evaluate the current procedures involved in the information gathering activities.

The MEP Office identified a need to develop an Automated Payload Experiment Tool (APET) which would lead experiment developers through the development planning process, obtain necessary information, establish an electronic data exchange avenue and allow easy manipulation/reformatting of the collected information. In order to fulfill this need, the University of Alabama in Huntsville (UAH) was tasked to design and develop the APET software package to meet the increasing demands to lighten the burden of documentation preparation and maintenance for NASA and its customers.

2.2 System Requirements

The objective of APET is to provide an easy to use tool to the Principal Investigator (PI) team. To ensure ease of use, few computer hardware requirements are necessary to operate the APET software package.

APET is designed to run on any IBM-PC compatible personal computer. While it is possible to use the system on a 386 PC, it is recommended that the
APET user install the software on a 486 PC or higher. The graphical displays, multiple screen windows, and the complexity of the system cause noticeable slowdowns on any machine less than the 486.

The software requires that the PC be equipped with a hard disk drive. For proper execution, the hard drive (or some partition of it) must be named C:. The Project Plan version of APET will require approximately 2M (mega-bytes) of space on the hard drive for the system, plus another 1M on the hard drive for the data files created by the user. However, for optimal performance, the hard drive should have a total of at least 4M free upon installation of the software.

For ease of use, the PC should be equipped with a mouse. This, however, is not mandatory. APET utilizes the hypertext technology, which offers a point-and-click user interface. Instead of a mouse, the user does have the option of pressing selected function keys to achieve the same effect.

2.3 Installation

The APET software package is provided on one high density diskette. The files stored on this diskette have been compressed; therefore, it is required that the user follow several simple steps to ensure correct installation.

1) Insert the diskette marked as "APET PROJECT PLAN" in the drive designated as A:. If the A: drive on your system is not the correct size, then use the DOS ASSIGN command to redesignate the drives appropriately. (For example, if you have a 3 1/2" disk but your 3 1/2" drive is B:, then at the DOS prompt type ASSIGN A: B:.)

2) From this drive (A:) type:

PINSTALL.

This will activate the installation routine. A series of instructions and informational text will be presented. Each screen will advise what is transpiring in the installation procedure. The installation routine will create a subdirectory on the C: drive called GARDEN. Once created, the files contained on the installation disk will be copied to the directory C:\GARDEN. Most of these files have been compressed to conserve disk space. An uncompress routine will be invoked to return these files to their normal (and usable) condition. To cancel the installation at any time, press the CNTL (control) C keys.

3) Upon successful installation of the APET program files, the message INSTALLATION ROUTINE COMPLETE will be displayed. The APET application, running under the direction of Knowledge Pro software, will be entered and you will be presented the opening menu. All subsequent sessions using the APET software may be initiated by going to the C:\GARDEN subdirectory and typing PLAN.
2.4 Getting Started

This application uses hypertext technology. Hypertext software systems allow for the retrieval of related information at the point and click of a mouse or, if a mouse is not used, at the touch of one or two keystrokes. For information on a highlighted topic, just move the mouse to that word and click. A window will be opened, overlaying the current window. In the new window, the information will immediately be displayed. Once this support information has been reviewed, press SPACE (or ESC) once to close the window and return to your original screen. If you are not using a mouse, please use the F3 and F4 function keys (marked Select and View) as indicated at the bottom of the screen. (See Figure 1). The F3 key allows you to select the different hypertext topics. Once the desired topic is selected (i.e. highlighted), the F4 key calls the background information for view.

Multiple page displays are indicated by the Page 1 of 2 message at the lower right of the screen. To navigate through multiple screen displays, please use the Page Up and Page Down keys to scroll either forward or backward through the pages.

For help at anytime throughout the APET application, press the F1 key. This will retrieve location sensitive help information, and may be called from the system or system-called edit screens. This will be the method by which assistance information will be retrieved throughout this application.

APET has been designed as a menu-driven software package. This means that any function required of the user can be activated via a menu option. This includes exiting the system. It is strongly recommended that the user always "back out" of the application by using the appropriate menu options, i.e. "Return to Previous Menu". An option does exist to exit from any point in the application by selecting F10. It is not recommended that this be used from inside a question/answer section of the application. The F10 command causes an immediate exit from the program, without checking to ensure that open files have been properly saved. Therefore, the user may experience data loss if the application is exited in this manner.

After the installation and initial use of the APET software, future systems will be initiated by going to the C:\GARDEN subdirectory and typing PLAN. This will activate the software and present the opening menu. (See Figure 2).

Due to the hypertext capabilities of the APET software, a large amount of RAM (random access memory) is required. Because of the heavy RAM demand, proper execution of the software requires no other software package be running simultaneously with the APET software. Whenever the available RAM becomes too little for the application, an "Insufficient Memory" message will be shown at the bottom right of the screen. To alleviate this situation, simply get out of APET and reboot the system. This will usually free up all available RAM and ensure proper execution. (See Helpful Hints for further instruction.)
So you want to fly on the Shuttle.

Well before you can, we must get a little information about your experiment and its objectives.

If you have flown with us in the past, you may remember a substantial amount of paper documentation was required. This application, the [Automated Payload Experiment Tool], is designed to alleviate much of the burden on experiment preparation by utilizing a [hypertext] knowledge-based system. This system can be used to prepare one of our support documents, the Project Plan, which describes the overall plan for proceeding with a project.

Press SPACE to continue.

---

**Figure 1**
Sample Screen Layout Using Hypertext
Please select the activity or your choice, or choose Exit to leave the system.

How to use the System
Project Selection
Project Plan (Overview)
Glossary/Acronyms
Print Glossary/Acronyms
Exit System

Figure 2
Opening APET Main Menu
3.0 USING THE APET SYSTEM

3.1 How to Use the System

Because the use of a hypertext tool may be a new experience, a brief on line tutorial is provided with APET. To use this tool, please choose option one on the opening menu entitled "How to Use the System". To select this option, point with the mouse to the phrase and click. If not using a mouse, use the arrow keys to highlight the option and press RETURN. You can tell when an option has been selected because it (the phrase or word) will be highlighted differently from all other options. As the mouse is moved to other options, each in turn will be highlighted.

Once the "How to Use the System" option has been selected, a different screen will be presented with a brief overview of hypertext and the methods of selecting topics. (This overview is much the same as appears in Section 2.4 of this user's guide.) Practice selecting topics and moving from one screen to another using either the mouse or the appropriate keyboard function keys.

3.2 Project Selection

The APET software package will accommodate one or more experiments for the user. However, each experiment must be identified by a short (8 characters or less) name, which must conform to the naming convention used by the DOS computer operating system. Briefly, these rules state that a DOS name cannot be over 8 characters in length, and must contain a combination of either letters, numbers, or the underscore (_) character. Any other special keys, including the SPACE, are prohibited. The rationale behind this naming convention is to allow storage of data files for each experiment in a subdirectory for that specific experiment. For example, if a user is working on two experiments, identified as THINFILM and HIPROTEIN, then there would be a subdirectory for each. The configuration of these files would be as follows:

<table>
<thead>
<tr>
<th>Root Directory</th>
<th>Application Directory</th>
<th>Experiment Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\</td>
<td>GARDEN\</td>
<td>THINFILM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HIPROTEIN</td>
</tr>
</tbody>
</table>

Therefore, all data files containing answers for the APET questions for the experiment THINFILM would appear in the subdirectory THINFILM. If additional experiments are required, the user would identify the new experiment and an additional subdirectory would be added.

Figure 3 shows the menu for selecting, adding, or deleting an experiment project. In the example, the experiment APCGF has been previously defined by the user. If the user wants to work on this experiment, he simply points and clicks on this selection. (This would be the condition in a majority of the cases, since most Principal Investigators will have only one active experiment at any given point.) However, if another experiment is required, the user would click on "ENTER A NEW PROJECT". The user would be prompted to identify the new experiment, and would immediately be passed into the APET system, where information regarding that experiment would be addressed.
Please select the project of your choice, or enter a new project.

APCGF
ENTER A NEW PROJECT
DELETE AN OLD PROJECT
RETURN TO MAIN MENU

F1 HELP
F8 DOS
F10 Quit

Figure 3
Project Selection/Identification Menu
If a user wishes to delete an experiment, along with all its associated data files, he may do so by selecting the "DELETE AN OLD PROJECT" option from the menu. However, there is no recoverable procedure to undelete a project. Therefore, the user is strongly advised to use this procedure with caution.

3.3 Project Plan Overview

The third selection from the APET Main Menu is the "Project Plan Overview". This option should be selected when the user wishes to see an overview of the Project Plan document, along with brief explanations of the information to be covered in each section/subsection of the document. For an additional overview of the topics to be addressed in the Project Plan, see Appendix A of this document.

3.4 Glossary/Acronyms

A number of NASA specific terms, definitions, and acronyms will appear as support material throughout the documentation process. One of the primary advantages of using a hypertext-based tool is to allow for easy and immediate retrieval of these terms.

Option number four from the APET Main Menu allows the user to retrieve a listing of these terms, and presents them in a form analogous to a glossary in a book. To view a definition, highlight the desired term and click. A term can be highlighted by using the mouse to move the cursor to that word, or by using the F3 key for selection. To view the definition, the user should either click the mouse or press the F4 key. The definition of that word/term will be presented. Should the definition contain a term that requires further description, highlight that word and click. The new definition will overlay the previous definition. This method can be repeated as long as further definitions exist and the memory capacity of the machine is not exceeded.

Please note that the glossary consists of multiple pages. Remember to navigate through the multi-page displays by using either the Page Up/Page Down function keys.

3.5 Print Glossary/Acronyms

Option number five from the APET Main Menu activates a routine for the printing of the glossary/acronym list, as discussed in 3.4. Because the output of this selection will be a multi-page document, the use of this option will be rare.

3.6 Filling Out the Project Plan

To fill out the Project Plan, the user must first select "Project Selection" from the APET Main Menu. Then the Project Selection/Identification Menu (Figure 3) will be presented for the user to identify either a new project or select an existing project. Once the selection/identification has been made, the Project Plan Activity Menu (Figure 4) will be presented. Please note that the selected project is shown on the upper right corner of the screen. Please be sure that the project shown is the one you wish to work.
The Project Plan Activity Menu presents eleven options for the user. The most significant of these is the first: "FILL OUT DOCUMENTATION". The selection of this option will present the Fill Out Project Plan Documentation Menu (Figure 5). This menu offers the user two primary options: "ENTER PROJECT INITIALIZATION INFORMATION" and "COMPLETE PROJECT PLAN".

3.7 Entering Project Initialization Information

Under most circumstances, the first information entered by the user into the APET system is the project initialization information. To enter this information, selection one from the Fill Out Project Plan Documentation Menu is chosen. This information is used to identify certain aspects of the experiment, and will be used throughout the documentation process. Entries include the PI name, his organization, address, city, state, zip, phone, and experiment title. This information will be provided in the form of typed written responses to user prompts. For example, the user will be asked:

Please enter your first and last names, i.e. Dr. John Doe.

The user should respond with a one line response. (If there is a limitation on the length of this one line response, the screen display will provide an instruction, such as "Please limit your response to 16 characters.") This will be the format of user entries throughout the APET application. When the user is prompted to make an entry, the response should be on one line. When the user responds with a RETURN, the answer is stored and the next question, if one exists, is asked. Once all questions for that section have been answered, the answers are written to a data file.

In the "Project Initialization Information" subsection, the only variation in the user prompt/one line response routine is with the experiment title. Because experiment titles can be several lines, the user is given a prompt and immediately sent to the APET editor. This editor is a small version of a word processor, with many of the functions of a common word processing package. The commands used in the APET editor are similar to those in the software package WORDSTAR. To see the commands available, press the F1 function key from inside the editor. A separate window will be opened and will overlay the current screen. From there, page down until the function you wish to perform is shown. The one-to-two keystroke command to accomplish the task will be shown. (Note: The symbol (') represents the CONTROL key., i.e. "KQ means to hold the CONTROL key down while pressing the letter K, then press the letter Q.") When the user is finished entering the answer into the editor, he may exit by pressing the ESC (escape) key. A message will then appear on the screen that tells the user what file name is being saved. Respond with a RETURN to accept this name and save the answer, or an ESC to cancel the answer.

**Important:** Do not change the file name when the APET software asks if the name is acceptable. Future sessions that allow the user to change, delete, print, or display the Project Plan answers require that these file names be used. Changing the file name will make the file either inaccessible or inappropriate. Therefore, always accept the file name as given.
The project you have selected is: APCGF
Please enter your choice of activities from the list.

FILL OUT DOCUMENTATION
PRINT PROJECT PLAN
DISPLAY PROJECT PLAN
DETERMINE IF PROJECT PLAN IS COMPLETE
BASELINE DOCUMENT
COMPARE BASELINE TO CURRENT REVISION
COPY ANSWERS TO DISK
RETURN TO PREVIOUS MENU
RETURN TO MAIN MENU
EXIT SYSTEM

F1 HELP

F8 DOS  F10 Quit

Figure 4
Project Plan Activity Menu
Please select the activity you wish to perform on the APCGF project.

- Enter Project Initialization Information
- Complete Project Plan
- Return to Previous Menu
- Return to Main Menu
- Exit System

Figure 5
Fill Out Project Plan Documentation Menu
3.8 Complete Project Plan

To complete the Project Plan document, the user must answer a series of questions about the experiment. (For an outline of the topics to be addressed, see Appendix A.) A sample question from the Project Plan is shown in Figure 6.

The questions are grouped into topics, a list of which is presented to the user. The user is asked to select the topic which is to be addressed. The chosen topic will be displayed, along with either its accompanying question or the subtopics covered within that section. If the user chooses to fill out one of the subsections, he will be prompted to select from a list. Upon subheading selection, a brief prompt, along with any necessary instructions for answering the question, will be displayed.

Each question in the Project Plan will invoke the editor. Answers should be given in the form of short narratives, consisting of one or more paragraphs of text. Sufficient space will be given to write as much (or little) as required. A sample of this procedure is shown in Figures 7 thru 10.

Some of the topics have accompanying illustration(s) in order to give the user a better understanding of what information is needed. If you wish to view the illustration, simply click on, or select, the appropriate phrase. This will cause the screen to momentarily go blank, and the illustration will then be presented. After you have viewed the illustration, press SPACE and the program will return you to your original screen.

3.9 Printing the Project Plan

The user has two methods available to generate output from the APET software. These include either printing the document or displaying the document to the screen. The APET application was designed to be flexible enough to go to a variety of printers. As with most output, the best results will be with the use of a laser printer. If a laser printer is not available, the use of a dot-matrix printer will also be acceptable. A variety, although not nearly exhaustive, of dot-matrix printers have been tested with the APET software, and all have performed well.

If the document has previously been baselined (discussed later in section 3.12), then a menu will appear giving the user the option of printing the document from the baselined version, the current revision, or neither version. If the neither option is chosen, then it is assumed that the user does not want the document printed, and the program will automatically return to the previous menu. If the baseline option is chosen, then the document will be printed from the version of the document which the user is currently revising. If the document has not been baselined, then the document will be printed from the current version.

The printing of the Project Plan can be accomplished in either of two methods. First, the user may choose the section to be printed. This is recommended for a document that is in process. Once the entire Project Plan
4.1 Introduction

Please describe the relevance of the investigation and provide a summary rationale as to why a flight experiment is required (limit to one printed page).

\[ \text{C:\GARDEN\APCGF\PPE4_1.DAT} \]

THIS IS WHERE YOUR ANSWER GOES.

---

**Figure 6**
Sample Project Plan Question Screen
has been completed, the user can generate the document in its entirety. An initial page eject will normally (depending on printer type) advance a blank sheet of paper before the cover sheet is printed. This will be followed by a second page advance, then page one of the document will be printed, followed by two, three, etc. through the end of the document. Because there are often graphics, tables, etc. that must be inserted within the textual document, no table of contents is printed. Because of the limitation of graphics support, it is suggested that all externally generated graphic illustrations, tables, etc. be provided in an appendix, with appropriate references throughout the document.

While the print procedure is active, a message will appear in the lower left of the screen. No other activities may take place while the document is printing. In case the printer runs out of paper, an error message will appear. Reload paper in the printer and press the SPACE key to continue.

3.10 Displaying the Project Plan

The second method of generating output using APET is to display sections of the Project Plan to the screen. The Project Plan is divided into four major sections. Content (Section 4) is subdivided into nine subsections, with five of those subsections further divided. (see Appendix A). The user has the capability of selecting a section and seeing the identical output as would appear if the document was printed. Displaying the Project Plan is recommended to quickly review answers, especially during the development phase of document preparation.

If the document has previously been baselined (discussed later in section 3.12), then a menu will appear giving the user the option of displaying the document from the baselined version, the current revision, or neither version. If the neither option is chosen, then it is assumed that the user does not want the document displayed, and the program will automatically return to the previous menu. If the baseline option is chosen, then the document will be displayed from the version of the document which the user is currently revising. If the document has not been baselined, then the document will be displayed from the current version.

Most Project Plan sections will require multiple page displays. Please note that to view the equivalent of an entire printed page, there will be at least three and usually four screen displays. Use the Page Up/Page Down method to move up or down in the document. Once a page is adequately reviewed, press the SPACE key to retrieve the next page in sequence. To abandon a display at any time, press the F10 key.

3.11 Determining if Project Plan is Complete

To aid in the logical completion of the Project Plan, APET provides the user with the ability of "Determining if Project Plan is Complete". By selecting this option, the user activates a routine that checks for the existence of the data files created as answers to the Project Plan questions. The missing data files are analyzed to determine which sections have not been answered, and a listing is displayed to the screen.
3.12 Baselining a Document

At some point in the documentation procedure, the Project Plan will be considered complete and released to external offices, agencies, organizations, etc. When this occurs, that version of the document is considered the baseline, and should be easily identified as such.

To aid in the process of maintaining separate versions of the Project Plan, an option exists to baseline the current version of the document. (See Figure 11). The selection of this option will cause a replica of the current version's answers (or data files) to be copied to a new subdirectory for that experiment. This new subdirectory will be called BASELINE. From that point, all additional editing will transpire on a new version of the answers, while the baselined version of the answers will remain intact. The generation of output will require the user to identify which version (baseline or current revision) he wishes to access.

3.13 Comparing a Baseline to the Current Revision

Once the document has been baselined (See Section 3.12), the user may wish to compare this baseline with the current revision. APET provides a mechanism to accomplish this task. By selecting the option "Compare Baseline to Current Revision", a DOS routine will be invoked to compare all identical data files from the current revision to the baseline document. This comparison generates a file that can then be displayed or printed, so that a quick review will show which answers have been modified since the original baseline date.

3.14 Copying Answers to Disk

The final output option provided by APET is the creation of files that contain all data generated by the software. This can be used as either a backup mechanism during the creation of the files, or as a means of submission of the final document instead of a hard copy/printed document. By submitting the answers on diskette, the receiving party can have direct access to the answers in the same manner as would the sender. These files are not formatted as an ASCII file, and should not be confused with the final report output.

The user will have the option of selecting either the baseline document or the current revision. After this selection, the user is asked to select the drive to receive the backup (either A:, B:, C:, or D:). A DOS copy command will then be invoked to copy all files to the selected drive.
Which section do you wish to address?

1.0 General
2.0 Preparation and Approval
3.0 Changes
4.0 Content
Return to Previous Menu

Figure 7
Project Plan List of Sections
Figure 8
Project Plan Section Selection
4.4 Technical Plan

The outline for the Technical Plan includes the following sections:

- EXPERIMENT HARDWARE DESCRIPTION
- PAYLOAD CLASSIFICATION
- DEVELOPMENT APPROACH
- TECHNOLOGY PLAN
- LOGISTICS
- MISSION OPERATIONS, TRAINING AND DATA MANAGEMENT
- ANALYSIS OF MISSION RESULTS
- FACILITIES
- SAFETY
- RETURN TO PREVIOUS MENU

Figure 9
Project Plan Subtopic Selection
The Project Manager is responsible for updating a MSAD Project Plan when significant changes occur (such as changes in scope, organization, or roles and responsibilities). This does not apply to resources, schedules or manpower, which are updated through normal budgeting and project monitoring activities. The Project Manager will establish a change control process for maintaining the MSAD Project Plan and other project documentation.

Press SPACE to continue

Figure 10
Project Plan Topic Narrative
At some point in the documentation process, it is necessary to declare that all documents are complete, and that any changes to be made will be treated as revisions to the baseline document.

Do you want to baseline your answers at this time?

YES
NO

Figure 11
Baseline Menu
4.0 HELPFUL HINTS

1) Avoid the use of the F10 key to exit from within the APET application. It is a better practice to back out of the APET system through the use of the menus. By doing so, the user ensures that all answers are properly recorded to the disk drive. Use of the F10 key from within the APET application will allow the user to exit but will not automatically save information generated during the session.

2) APET does not support the insertion of externally generated graphics, tables, equations, or other non-text material. To alleviate this problem without the added labor of using a secondary word processor, it is suggested that any such material be included in an Appendix, and referenced in the text generated in APET.

3) To insert an externally generated text file into the text area in the APET editor, use the command "KR from within the editor. This is one of a variety of commands that can be used from the APET word processor. To see all available commands, press the F1 key from inside the editor and page through the instruction set.

4) The APET editor uses a word wrap routine that automatically wraps the text to the next line (a common word processing feature). It also maintains vertical alignment along the left margin. If you use indented paragraphs, please be sure that the line after the indented line begins in the column you desire. To do this, use the backspace key to move the first word in the line to the column desired. The recommended solution to this problem is not to indent paragraphs, but instead insert a blank line between each paragraph.

5) If your computer system is configured to automatically load WINDOWS or some other application package, it may be necessary to alter the AUTOEXEC.BAT file (located in the boot drive). Instructions for changing the automatic load of an application will vary by computer. One of the easier methods is to edit the AUTOEXEC.BAT file and remove the line that calls the package. For example, WINDOWS is called by the command WIN. By preventing these packages from loading, a significant amount of RAM is freed and allowed for use by APET.
APPENDIX A

Project Plan Topic Outline
1.0 INTRODUCTION
2.0 PREPARATION AND APPROVAL
3.0 CHANGES
4.0 CONTENT

4.1 Introduction
4.2 Objective
4.3 Science Requirements
4.4 Technical Plan
  4.4.1 Experiment Hardware Description
  4.4.2 Payload Classification
  4.4.3 Development Approach
  4.4.4 Technology Plan
  4.4.5 Logistics
  4.4.6 Mission Operations, Training and Data Management
  4.4.7 Analysis of Mission Results
  4.4.8 Facilities
  4.4.9 Safety
4.5 Implementation Plan
  4.5.1 Implementation Approach
  4.5.2 Summary Work Breakdown Structure
  4.5.3 Documentation
4.6 Management Plan
  4.6.1 Project Management Responsibilities and Organization
  4.6.2 Mission Management Responsibilities and Organization
4.7 Schedule
4.8 Cost Control Plan
  4.8.1 Resources
4.8.2 Cost Control Guidelines

4.8.3 Cost Reporting Control Structure

4.8.3.1 NASA Reports

4.8.3.2 Contractor Reports

4.8.4 Cost Control Strategy

4.9 Project Reviews and Meetings

4.9.1 Internal Reviews

4.9.2 External Reviews

4.9.3 Design and Readiness Reviews
APPENDIX B

APET Editor Commands
MOVING THE CURSOR:

^D ......................... Left One Character
^S ........................ Right One Character
^A ........................ Left One Word
^F ........................ Right One Word
^I ........................ Tab
^E ........................ Up One Line
^X ........................ Down One Line
^W ........................ Scroll Up
^Z ........................ Scroll Down
^R ........................ Page Up
^C ......................... Page Down
^QS ........................ Beginning of a Line
^QD ........................ End of a Line
^QE ........................ Beginning of a Page
^QX ........................ End of a Page
^QR ........................ Beginning of the File
^QC ........................ End of the File
^QB ........................ Beginning of Marked Block
^QK ........................ End of Marked Block

DELETING AND INSERTING TEXT:

^G ........................ Delete Character Under Cursor
^H ........................ Delete Character to Left of Cursor
^T ........................ Delete Next Word
^Y ........................ Delete a Line
^QY ........................ Delete to the End of a Line
^KY ........................ Delete a Marked Block
^V ........................ Insert On/Off
^N ........................ Insert a Line

BLOCK COMMANDS:

^KS ........................ Save This File
^KB ........................ Mark Beginning of Block
^KK ........................ Mark End of Block
^KH ........................ Hide/Display Block
^KC ........................ Copy a Block
^KV ........................ Move a Block
^KY ........................ Delete a Block
^KR ........................ Read a Block from a File
^KW ........................ Write a Block to a File
^KP ......................... Print a Block or Entire File if no Block is Marked
FORMATTING COMMANDS:

\^B .................................. Reformat Paragraph
\^OR ................................ Set Right Margin
\^QI ................................ Toggle Autoindent Mode
\^QW ................................ Toggle Word Wrap

FIND AND REPLACE COMMANDS:

\^QA ................................ Find and Replace a String
\^QF ................................ Find an Occurrence of a String
\^L ................................... Find the Next Occurrence
For assistance in using this software, or to offer suggestions or comments, please contact the following:

Mr. Gary Maddux
Ms. Anna Provancha
Mr. David Chattam

at (205) 895-6343,
or write

Systems Management and Production Laboratory
Research Institute
RI E-47
The University of Alabama in Huntsville
Huntsville, AL 35899
APPENDIX E

Science Requirements Envelope Document User Guide
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2.0 DISCUSSION</td>
<td>1</td>
</tr>
<tr>
<td>2.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>2.2 System Requirements</td>
<td>1</td>
</tr>
<tr>
<td>2.3 Installation</td>
<td>2</td>
</tr>
<tr>
<td>2.4 Getting Started</td>
<td>2</td>
</tr>
<tr>
<td>3.0 USING THE APET SYSTEM</td>
<td>6</td>
</tr>
<tr>
<td>3.1 How to Use the System</td>
<td>6</td>
</tr>
<tr>
<td>3.2 Project Selection</td>
<td>6</td>
</tr>
<tr>
<td>3.3 Science Requirements Envelope</td>
<td>8</td>
</tr>
<tr>
<td>3.4 Glossary/Acronyms</td>
<td>8</td>
</tr>
<tr>
<td>3.5 Print Glossary/Acronyms</td>
<td>8</td>
</tr>
<tr>
<td>3.6 Filling Out the Science Requirements Envelope</td>
<td>9</td>
</tr>
<tr>
<td>3.7 Entering Project Initialization</td>
<td>9</td>
</tr>
<tr>
<td>3.8 Complete Science Requirements Envelope</td>
<td>12</td>
</tr>
<tr>
<td>3.9 Printing the Science Requirements Envelope</td>
<td>17</td>
</tr>
<tr>
<td>3.10 Displaying the Science Requirements Envelope</td>
<td>18</td>
</tr>
<tr>
<td>3.11 Create an ASCII File of the Science Requirements Envelope</td>
<td>18</td>
</tr>
<tr>
<td>3.12 Baselining a Document</td>
<td>18</td>
</tr>
<tr>
<td>3.13 Comparing a Baseline to the Current Revision</td>
<td>19</td>
</tr>
<tr>
<td>3.14 Copying Answers to Disk</td>
<td>19</td>
</tr>
<tr>
<td>4.0 HELPFUL HINTS</td>
<td>21</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1 ....... Sample Screen Layout Using Hypertext
Figure 2 ....... Opening APET Main Menu
Figure 3 ....... Project Selection/Identification Menu
Figure 4 ....... Science Requirements Envelope Document Activity Menu
Figure 5 ....... Fill Out Science Requirements Envelope Document Menu
Figure 6 ....... Science Requirements Envelope Document Previous Session Menu Screen
Figure 7 ....... Science Requirements Envelope Document Question Completion Menu
Figure 8 ....... Science Requirements Envelope Document Topic Selection Menu
Figure 9 ....... Sample Science Requirements Envelope Document Question
Figure 10 ....... Baseline Menu
1.0 INTRODUCTION

So you want to fly an experiment on the Shuttle.

Well, to begin the process, we must get a little information about your experiment and its requirements.

If you have flown with us in the past, you may remember a substantial amount of paper documentation was required. This application, the Automated Payload Experiment Tool, is designed to alleviate much of the burden of the document preparation and maintenance process. This system can currently be used to prepare four of our support documents: the Science Requirements Document (SRD), which defines the science objectives, the Experiment Requirement Document (ERD), which defines the experiment design/build requirements, the Project Plan, which is the basic planning document that describes the overall plan for proceeding with the project, and the Science Requirements Envelope Document. The version you have is for the completion of the Science Requirements Envelope Document only.

2.0 DISCUSSION

2.1 Background

The Microgravity Experiment Projects (MEP) Office of the Payload Projects Office (PPO) at the Marshall Space Flight Center (MSFC) is currently responsible for collecting and coordinating experiment/facility specifications and requirements between NASA and various colleges, universities, research centers, and public- and private-sector organizations that are selected or are requesting to fly their respective experiments on NASA flights. This coordination involves the communication of flight hardware requirements and the preparation and review of all documentation between NASA and the research groups. To reduce difficulties encountered by these customers of NASA, an effort was undertaken to research, analyze, and evaluate the current procedures involved in the information gathering activities.

The MEP Office identified a need to develop an Automated Payload Experiment Tool (APET) which would lead experiment developers through the development planning process, obtain necessary information, establish an electronic data exchange avenue and allow easy manipulation/reformatting of the collected information. In order to fulfill this need, the University of Alabama in Huntsville (UAH) was tasked to design and develop the APET software package to meet the increasing demands to lighten the burden of documentation preparation and maintenance.

2.2 System Requirements

The objective of APET is to provide an easy to use tool to the Principal Investigator (PI) team. To ensure ease of use, few computer hardware requirements are necessary to operate the APET software package.

APET is designed to run on any IBM-PC compatible personal computer. It is
recommended that the APET user install the software on a 386 PC or higher. The multiple screen windows, and the complexity of the system cause noticeable slowdowns on any machine less than the 386.

The software requires that the PC be equipped with a hard disk drive. For proper execution, the hard drive (or some partition of it) must be named C:. The Science Requirements Envelope version of APET will require approximately 1M of space on the hard drive for the system, plus another 1M on the hard drive for the data files created by the user. However, for optimal performance, the hard drive should have a total of at least 3M free upon installation of the software.

For ease of use, the PC should be equipped with a mouse. This, however, is not mandatory. APET utilizes the hypertext technology, which offers a point-and-click user interface. Instead of a mouse, the user does have the option of pressing selected function keys to achieve the same effect.

2.3 Installation

The APET software package is provided on one 3 1/2" diskette. The files stored on this diskette have been compressed; therefore, it is required that the user follow several simple steps to ensure correct installation.

1) Insert the 3 1/2" diskette in the drive designated as A:. If the A: drive on your system is not a 3 1/2" drive, then use the DOS ASSIGN command to redesignate the drives appropriately. (For example, if you have a 3 1/2" disk, but your 3 1/2" drive is B:, then at the DOS prompt type ASSIGN A: B:.)

2) From the A: drive type:

\textit{VINSTALL.}

This will activate the installation routine. A series of instructions and informational text will be presented. Each screen will advise what is transpiring in the installation procedure. The installation routine will create a subdirectory on the C: drive called GARDEN. Once created, the files contained on the installation disk will be copied to the directory C:\GARDEN. Most of these files have been compressed to conserve disk space. An uncompress routine will be invoked to return these files to their normal (and usable) condition. To cancel the installation at any time, press the CTRL (control) C keys.

3) Upon successful installation of the APET program files, the message \textit{INSTALLATION ROUTINE COMPLETE} will be displayed. The APET application, running under the direction of Knowledge Pro software, will be entered and you will be presented the opening menu. All subsequent sessions using the APET software may be initiated by going to the C:\GARDEN subdirectory and typing \textit{ENVELOPE}

2.4 Getting Started

This application uses hypertext technology. Hypertext software systems
allow for the retrieval of related information at the point and click of a mouse or, if a mouse is not used, at the touch of one or two keystrokes. For information on a highlighted topic, just move the mouse to that word and click. A window will be opened, overlaying the current window. In the new window, the information will immediately be displayed. Once this support information has been reviewed, press SPACE (or ESC) once to close the window and return to your original screen. If you are not using a mouse, please use the F3 and F4 function keys (marked Select and View) as indicated at the bottom of the screen. (See Figure 1). The F3 key allows you to select the different hypertext topics. Once the desired topic is selected (i.e. highlighted), the F4 key calls the background information for view.

Multiple page displays are indicated by the Page 1 of 2 message at the lower right of the screen. To navigate through multiple screen displays, please use the Page Up and Page Down keys to scroll either forward or backward through the pages.

For help at anytime throughout the APET application, select the F1 key. This will retrieve location sensitive help information, and may be called from the system or system-called edit screens. This will be the method by which assistance information will be retrieved throughout this application.

APET has been designed as a menu-driven software package. This means that any function required of the user can be activated via a menu option. This includes exiting the system. It is strongly recommended that the user always "back out" of the application by using the appropriate menu options, i.e. "Return to Previous Menu". An option does exist to exit from any point in the application by selecting F10. It is not recommended that this be used from inside a question/answer section of the application. The F10 command causes an immediate exit from the program, without checking to ensure that open files have been properly saved. Therefore, the user may experience data loss if the application is exited in this manner.

After the installation and initial use of the APET software, future systems will be initiated by going to the C: \GARDEN subdirectory and typing \ENVELOPE. This will activate the software and present the opening menu. (See Figure 2).

Due to the hypertext capabilities of the APET software, a large amount of RAM (random access memory) is required. Because of the heavy RAM demand, proper execution of the software requires no other software package be running simultaneously with the APET software. Whenever the available RAM becomes too little for the application, an "Insufficient Memory" message will be shown at the bottom right of the screen. To alleviate this situation, simply get out of APET and reboot the system. This will free up all available RAM and ensure proper execution. (See Helpful Hints for further instructions.)
So you want to fly on the Shuttle.

Well before you can, we must get a little information about your experiment and its requirements.

If you have flown with us in the past, you may remember a substantial amount of paper documentation was required. This application, the Automated Payload Experiment Tool, is designed to alleviate much of the burden of the document preparation and maintenance process by utilizing a hypertext, knowledge-based system. This system can be used to prepare one of our support documents, the Science Requirements Envelope Document, which provides an envelope or volume of science requirements for a type of experimentation.

Press SPACE to continue.

Figure 1
Sample Screen Layout Using Hypertext
Please select the activity of your choice, or choose Exit to leave the system.

HOW TO USE THE SYSTEM
PROJECT SELECTION
SCIENCE REQUIREMENTS ENVELOPE DOCUMENT (OVERVIEW)
GLOSSARY/ACRONYMS
PRINT GLOSSARY/ACRONYMS
EXIT SYSTEM

F1 HELP

F8 DOS  F10 Quit

Figure 2
Opening APET Main Menu
3.0 USING THE APET SYSTEM

3.1 How to Use the System

Because the use of a hypertext tool may be a new experience, a brief online tutorial is provided with APET. To use this tool, please choose option one on the opening menu entitled "How to Use the System". To select this option, point with the mouse to the phrase and click. If not using a mouse, use the arrow keys to highlight the option and press RETURN. You can tell when an option has been selected because it (the phrase or word) will be highlighted differently from all other options. As the mouse is moved to other options, each in turn will be highlighted.

Once the "How to Use the System" option has been selected, a different screen will be presented with a brief overview of hypertext and the methods of selecting topics. (This overview is much the same as appears in Section 2.4 of this user's guide.) Practice selecting topics and moving from one screen to another using either the mouse or the appropriate keyboard function keys.

3.2 Project Selection

The APET software package will accommodate one or more experiments for the user. However, each experiment must be identified by a short (8 characters or less) name, which must conform to the naming convention used by the DOS computer operating system. Briefly, these rules state that a DOS name cannot be over 8 characters in length, and must contain a combination of either letters, numbers, and the underscore (_) character. Any other special keys, including the SPACE, are prohibited. The rationale behind this naming convention is to allow storage of data files for each experiment in a subdirectory for that specific experiment. For example, if a user is working on two experiments, identified as THINFILM and HIPROTEIN, then there would be a subdirectory for each. The configuration of these files would be as follows:

<table>
<thead>
<tr>
<th>Root Directory</th>
<th>Application Directory</th>
<th>Experiment Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>C:\</td>
<td>GARDEN\</td>
<td>THINFILM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HIPROTEIN</td>
</tr>
</tbody>
</table>

Therefore, all data files containing answers for the APET questions for the experiment THINFILM would appear in the subdirectory THINFILM. If additional experiments are required, the user would identify the new experiment and an additional subdirectory would be added.

Figure 3 shows the menu for selecting, adding, or deleting an experiment project. In the example, the experiment APCGF has been previously defined by the user. If the user wants to work on this experiment, he simply points and clicks on this selection. (This would be the case in a majority of the cases, since most Principal Investigators will have only one active experiment.
Please select the project of your choice, or enter a new project.

APCGF
ENTER A NEW PROJECT
DELETE AN OLD PROJECT
RETURN TO MAIN MENU

Figure 3
Project Selection/Identification Menu
at any given point). However, if another experiment is required, the user
would click on "ENTER A NEW PROJECT". The user would be prompted to identify
the new experiment, and would immediately be passed into the APET system,
where information regarding that experiment would be addressed.

If a user wishes to delete an experiment, along with all its associated
data files, he may do so by selecting the "DELETE AN OLD PROJECT" option from
the menu. However, there is no recoverable procedure to undelete a project.
Therefore, the user is strongly advised to use this procedure with caution.

3.3 Science Requirements Envelope Document Overview

The third selection from the APET Main Menu is the "Science Requirements
Envelope Document (Overview)". This option should be selected when the user wishes
to see an overview of the Science Requirements Envelope Document, along with brief
explanations of the information to be covered in each section/subsection of the
document. For an additional overview of the topics to be addressed in the Science
Requirements Envelope, see Appendix A of this document.

3.4 Glossary/Acronyms

A number of NASA specific terms, definitions, and acronyms will appear as
support material throughout the documentation process. One of the primary
advantages of using a hypertext-based tool is to allow for easy and immediate
retrieval of these terms.

Option number four from the APET Main Menu allows the user to retrieve a
listing of these terms, and presents them in a form analogous to a glossary in
a book. To view a definition, highlight the desired term and click. A term
can be highlighted by using the mouse to move the cursor to that word, or by
using the F3 key for selection. To view the definition, the user should
either click the mouse or press the F4 key. The definition of that word/term
will be presented. Should the definition contain a term that requires further
description, highlight that word and click. The new definition will overlay
the previous definition. This method can be repeated as long as further
definitions exist and the memory capacity of the machine is not exceeded.

Please note that the glossary consists of multiple pages. Remember to navigate
through the multi-page displays by using the Page Up/Page Down function keys.

3.5 Print Glossary/Acronyms

Option number five from the APET Main Menu activates a routine for the
printing of the glossary/acronym list, as discussed in 3.4. Because the
output of this selection will be a multi-page document, the use of this option
will be rare.
3.6 Filling Out the Science Requirements Envelope Document

To fill out the Science Requirements Envelope, the user must first select "Project Selection" from the APET Main Menu. The Project Selection/Identification Menu (Figure 3) will then be presented for the user to identify either a new project or select an existing project. Once the selection/identification has been made, the Science Requirements Envelope Activity Menu (Figure 4) will be presented. Please note that the selected project is shown on the upper right corner of the screen. Please be sure that the project shown is the one you wish to work.

The Science Requirements Envelope Activity Menu presents seven options for the user. The most significant of these is the first: "FILL OUT DOCUMENTATION". The selection of this option will present the Fill Out Science Requirements Envelope Documentation Menu (Figure 5). This menu offers the user two options: "ENTER PROJECT INITIALIZATION INFORMATION" and "COMPLETE SCIENCE REQUIREMENTS ENVELOPE DOCUMENT".

3.7 Entering Project Initialization Information

Under most circumstances, the first information entered by the user into the APET system is the project initialization information. To enter this information, selection one from the Fill Out Envelope Documentation Menu is chosen. This information is used to identify certain aspects of the experiment, and will be used throughout the documentation process. Entries include the PI name, his organization, address, city, state, zip, phone, and experiment title. This information will be provided in the form of type written responses to user prompts. For example, the user will be asked:

*Please enter your first and last names, i.e. Dr. John Doe.*

The user should respond with a one line response. *(If there is a limitation on the length of this one line response, the screen display will provide an instruction, such as "Please limit your response to 16 characters.") This will be the format of user entries throughout the APET application. When the user is prompted to make an entry, the response should be on one line. When the user responds with a RETURN, the answer is stored and the next question, if one exists, is asked. Once all questions for that segment have been answered, the answers are written to a data file.

In the "Project Initialization Information" subsection, the only variation in the user prompt/one line response routine is with the experiment title. Because experiment titles can be several lines, the user is given a prompt and immediately sent to the APET editor. This editor is a small version of a word processor, with many of the functions of a common word processing package. The commands used in the APET editor are similar to those in the software package WORDSTAR. To see the commands available, press the F1 function key from inside the editor. A separate window will be opened and will overlay the current screen. From there, page down until the function you wish to perform is shown. The one-to-two keystroke command to accomplish the task will be shown. *(Note: The symbol (') represents the CONTROL key, i.e. "KQ*
The project you have selected is: APCGF
Please enter your choice of activities from the list.

- FILL OUT DOCUMENTATION
- PRINT SCIENCE REQUIREMENTS ENVELOPE DOCUMENT
- DISPLAY ENVELOPE DOCUMENT
- CREATE AN ASCII FILE OF ENVELOPE DOCUMENT
- BASELINE DOCUMENT
- COMPARE BASELINE TO CURRENT REVISION
- COPY ANSWERS TO DISK
- RETURN TO PREVIOUS MENU
- RETURN TO MAIN MENU
- QUIT

Figure 4
Science Requirements Envelope Document Activity Menu
Please select the activity you wish to perform on the APCGF project.

Enter Project Initialization Information
[Complete Science Requirements: Envelope Document]
Return to Previous Menu
Return to Main Menu
Exit System

F1 HELP
F8 DOS
F10 Quit

Figure 5
Fill Out Science Requirements
Envelope Document Menu
means to hold the CONTROL key down while pressing the letter K, then press the letter Q.) When the user is finished entering the answer into the editor, he may exit by pressing the ESC (escape) key. A message will then appear on the screen that tells the user what file name is being saved. Respond with a RETURN to accept this name and save the answer, or an ESC to cancel the answer.

Important: Do not change the file name when the APET software asks if the name is acceptable. Future sessions that allow the user to change, delete, print, or display the Science Requirements Envelope answers require that these file names be used. Changing the file name will make the file either inaccessible or inappropriate. Therefore, always accept the file name as given.

3.8 **Complete Science Requirements Envelope Document**

To complete the Science Requirements Envelope Document, there are usually between 30 - 39 questions that must be answered. These answers will be in the form of short narratives, consisting of one or more paragraphs of text. Each question or user prompt will invoke the APET editor and give the user sufficient space to write as much (or little) as required. (For an outline of the topics to be addressed, see Appendix A.) A sample question scenario from the Science Requirement Envelope is shown in Figures 6 - 9.

The Complete Science Requirement Envelope Document session begins by asking the user if he has been to fill out the Science Requirements Envelope Document previously. A 'NO' response causes the questions to be asked in sequence. A 'YES' response displays the question topics in a list. If questions are to be answered from a list, a list will appear as a window that overlays the question screen. The user is expected to point-and-click on the appropriate topic. (If not using a mouse, use the arrow keys to select and press RETURN.) The user should click on the appropriate answer with the left-side mouse button.

For the initial Science Requirements Envelope session, the user would respond with a 'NO' and proceed to the questions. These questions will be asked in the same sequence as is shown in the outline. After each response, the user will be asked if he wants to continue to the next question. This gives the user a chance to end the session when he desires, rather than advancing through all the remaining questions. The title of each question screen will include the number of the question (i.e., Number 1 of 39). This allows the user to see where he is in the process and act accordingly.

If the user responds with a 'YES', which means there has been a previous session, the following question will appear:

Do you wish to change only one item, resume at a point and continue sequentially through the remainder of the Envelope Document, or complete all topics previously unanswered?

This allows the user one of three options. 1) He may select the one answer that needs changing, go directly to that answer and change it, then
Have you already begun to fill out the Science Requirements Envelope Document in a previous session?

Figure 6
Science Requirements Envelope Document
Previous Session Menu
Do you wish to change only one item, resume at a point and continue sequentially through the remainder of the Envelope Document or complete all topics previously unanswered?

CHANGE ONE ITEM
CHANGE AND CONTINUE
COMPLETE UNANSWERED TOPICS
QUIT

F1 HELP
F8 DOS
F10 Quit

Figure 7
Science Requirements Envelope Document Question Completion Menu
Change and Continue

With what subtopic do you wish to resume your activity?

Figure 8
Science Requirements Envelope Document
Topic Selection Menu
"Automated_Payload_Experiment_Tool"

Description of Experiment Type or Class (Question 1 of 39)

Please enter a narrative description of the type or class of the experiment. This topic is also addressed under the heading "Experiment Procedures to be Used" and "Description of Type of Experiments".

Press the RETURN key to enter the editor,

C:\GARDEN\APCGF\ENV1_1.DAT

ENTER TEXT HERE.

F1 HELP F3 Select Page
Space Cont. F4 View F8 DOS F10 Quit

Figure 9
Sample Science Requirements Envelope Document Question
record that answer to disk. 2) He can select the topic where he would like to resume his activities, answer that question, record the answer, and go to the next question in sequence. This gives the user the capability of selecting the 20th question, and proceed sequentially through the remaining 19 questions. 3) The user can complete all questions that have not yet been answered. This option will invoke a command to look at what answers (files) do not exist, and build a list of these topics. The user then selects the topic to answer, answers the question, records the answer, and goes to the next question of his choice. With each recorded answer, that topic is removed from the list.

3.9 Printing the Science Requirements Envelope Document

The user has three methods available to generate output from the APET software. These include printing the document, displaying the document, and creating an ASCII file of the document. The APET application was designed to be flexible enough to go to a variety of printers. As with most output, the best results will be with the use of a laser printer. If a laser printer is not available, the use of a dot-matrix printer will also be acceptable. A variety, although not nearly exhaustive, of dot-matrix printers have been tested with the APET software, and all have performed well.

If the document has previously been baselined (discussed later in section 3.12), then a menu will appear giving the user the option of printing the document from the baselined version, the current revision, or neither version. If "Neither" is chosen, then it is assumed that the user does not want the document printed, and the program will automatically return to the previous menu. If "Baseline" is chosen, then the document will be printed from the file which is in the project's Baseline subdirectory. If the current revision option is chosen, then the document will be printed from the version of the document which the user is currently revising. If no baselined version of the document exists, then the document will be printed from the current revision.

The Science Requirements Envelope Document is divided into six sections. Each of these section can be printed individually if the user wishes, or the document can be printed in its entirety. If the option to print the entire document is chosen, an initial page eject will normally (depending on printer type) advance a blank sheet of paper before the cover sheet is printed. This will be followed by a second page advance, then page one of the document will be printed, followed by two, three, etc. through the end of the document. Because there are often graphics, tables, etc. that must be inserted within the textual document, no table of contents is printed. Because of the limitation of graphics support, it is suggested that all externally generated graphic illustrations, tables, etc. be provided in an appendix, with appropriate references throughout the document. If one of the document's sections is chosen to be printed individually, a blank page will be ejected, followed by the desired section.

While the print procedure is active, a message will appear in the lower left of the screen. No other activities may take place while the document is printing. In case the printer runs out of paper, an error message will appear. Reload paper in the printer and press the SPACE key to continue.
3.10 Displaying the Science Requirements Envelope Document

The second method of generating output using APET is to display sections of the Science Requirements Envelope to the screen. The Science Requirements Envelope is divided into six major sections, with each divided into one or more subsections (see Appendix A). The user has the capability of selecting a section and seeing the identical output as would appear if the document was printed. Displaying the Science Requirements Envelope is recommended to quickly review answers, especially during the development phase of document preparation.

As stated in Section 3.9, if the document has previously been baselined then a menu will appear giving the user the options for displaying the document from the baselined version, the current revision, or neither version. If "Neither" is chosen, then it is assumed that the user does not want the document displayed, and the program will automatically return to the previous menu. If "Baseline" is chosen, then the document will be displayed from the file which is in the project's Baseline subdirectory. If "Current Revision" is chosen, then the document will be displayed from the version of the document which the user is currently revising. If no baselined version of the document exists, then the document will be displayed from the current revision.

Most Science Requirements Envelope Document sections will require multiple page displays. Please note that to view the equivalent of an entire printed page, there will be at least three and usually four screen displays. Use the Page Up/Page Down method to move up or down in the document. Once a page is adequately reviewed, press the SPACE key to retrieve the next page in sequence.

3.11 Create an ASCII File of Science Requirements Envelope Document

The APET software does not have the ability to generate or insert graphics, charts, etc. that were created in some other application. This is primarily due to the memory size limitations of the computer. However, to alleviate this limitation, APET does have the ability to generate an ASCII file of the Science Requirements Envelope Document's output. After choosing this option, the user need only type in the full file name (includes drive, file name, and extension). The file will then be created as a replica of the printed output.

The benefit of creating an ASCII text file of the Science Requirements Envelope is in providing the user with the capability of enhancing the final printing by inserting graphics, photos, tables, equations, or other difficult to create figures. In addition, different fonts, font sizes, and special effects can be used to dress up the final printed output.
3.12 Baselining a Document

At some point in the documentation procedure, the Science Requirements Envelope will be considered complete and released to external offices, agencies, organizations, etc. When this occurs, that version of the document is considered the baseline, and should be easily identified as such.

To aid in the process of maintaining separate versions of the Science Requirements Envelope, an option exists to baseline the current version of the document. (See Figure 10) The selection of this option will cause a replica of the current version's answers (or data files) to be copied to a new subdirectory for that experiment. This new subdirectory will be called BASELINE. From that point, all additional editing will transpire on a new version of the answers, while the baselined version of the answers will remain intact. The generation of output will require the user to identify which version (baseline or current revision) he wishes to access.

3.13 Comparing a Baseline to the Current Revision

Once the document has been baselined (See Section 3.12), the user may wish to compare this baseline with the current revision. APET provides a mechanism to accomplish this task. By selecting the option "Compare Baseline to Current Revision", a DOS routine will be invoked to compare all identical data files from the current revision to the baseline document. This comparison generates a file that can then be displayed or printed, so that a quick review will show which answers have been modified since the original baseline date.

3.14 Copying Answers to Disk

The final output option provided by APET is the creation of files that contain all data generated by the software. This can be used as either a backup mechanism during the creation of the files, or as a means of submission of the final document instead of a hard copy/printed document. By submitting the answers on diskette, the receiving party can have direct access to the answers in the same manner as would the sender. These files are not formatted as an ASCII file, and should not be confused with the final report output, which can be created using the "Create ASCII File of the Science Requirements Document" (discussed in Section 3.11).

The user will have the option of selecting either the baseline document or the current revision. After this selection, the user is asked to select the drive to receive the backup (either A:, B:, C:, or D:). A DOS copy command will then be invoked to copy all files to the selected drive.
At some point in the documentation process, it is necessary to declare that all documents are complete, and that any changes to be made will be treated as revisions to the baseline document.

Do you want to baseline your answers at this time?

YES
NO

Figure 10
Baseline Menu
4.0 HELPFUL HINTS

1) Avoid the use of the F10 key to exit the program. This key should only be used when the user wishes to exit completely out of the application. It is better practice to back out of the APET system through the use of the menus. By doing so, the user ensures that all answers are properly recorded to the disk drive. Use of the F10 Key will allow the user to exit but will not automatically save information generated during the session.

2) APET does not support the insertion of externally generated graphics, tables, equations, or other non-text material. To alleviate this problem without the added labor of using a secondary word processor, it is suggested that any such material be included in an Appendix, and referenced in the text generated in APET.

3) To insert an externally generated text file into the text area in the APET editor, use the command `KR from within the editor. This is one of a variety of commands that can be used from the APET word processor. To see all the available commands, press the F1 Key from inside the editor. A list of all the available commands is also provided in Appendix B of this manual.

4) The APET editor uses a word wrap routine that automatically wraps the line to the next line (a common word processing feature). It also maintains vertical alignment along the left margin. If you use indented paragraphs, please be sure that the line after the indented line begins in the column you desire. To do this, use the backspace key to move the first word in the line to the column desired. The recommended solution to this problem is not to indent paragraphs, but to insert a blank line between each paragraph.

5) If your computer system is configured to automatically load WINDOWS or some other application package, it may be necessary to alter the AUTOEXEC.BAT file (located in the boot drive). Instructions for changing the automatic load of an application will vary by computer. One of the easier methods is to edit the AUTOEXEC.BAT file and remove the line that calls the package. For example, WINDOWS is called up by the command WIN. By preventing these packages from loading, a significant amount of RAM is freed and allowed for use by APET.
APPENDIX A

Science Requirements Envelope Document
Topic Outline
OUTLINE FOR THE ENVELOPE DOCUMENT

1.0 PURPOSE

2.0 FUNCTION

3.0 PREPARATION, APPROVAL AND UPDATING

4.0 SUGGESTED CONTENTS

4.1 Introduction

4.1.1 Description of Experiment Type or Class .... ENV1_1.DAT
4.1.2 Scientific Knowledge to be Gained from This Type of Experimentation ........ ENV1_2.DAT
4.1.3 Value of Knowledge of This Type of Experimentation to scientific field ........ ENV1_3.DAT
4.1.4 Necessity for Space Environment to Experiment ................................ ENV1_4.DAT

4.2 BACKGROUND

4.2.1 Scientific Field to Which the Experiment Type belongs ................................ ENV2_1.DAT
4.2.2 Current Applications for Research in the Field ..................................... ENV2_2.DAT
4.2.3 Brief Historical Account of Prior Research in the Field .............................. ENV2_3.DAT
4.2.4 Current Research ................................ ENV2_4.DAT
4.2.5 Relationship of Proposed Experiment Type to Scientific Field ..................... ENV2_5.DAT
4.2.6 Anticipated Advance in State of the Art for This Type of Experimentation .... ENV2_6.DAT

4.3 JUSTIFICATION FOR CONDUCTING THIS TYPE OF EXPERIMENT IN SPACE

4.3.1 Limitations of Ground-Based Testing .......... ENV3_1.DAT
4.3.2 Limitations of Drop-Towers ................ ENV3_2.DAT
4.3.3 Limitations of Testing in Aircraft .......... ENV3_3.DAT
4.3.4 Need for Accommodations in the Shuttle ...... ENV3_4.DAT
4.3.5 Limitations of Mathematical Modeling ...... ENV3_5.DAT
4.3.6 Limitations of Other Modeling Approaches ... ENV3_6.DAT

4.4 DESCRIPTION OF EXPERIMENT TYPES

4.4.1 General Description of Type of Experiments.. ENV4_1.DAT
4.4.2 Types of Experiment Procedures to be Used .. ENV4_2.DAT
4.4.3 Types of Measurements and Ranges of Values.. ENV4_3.DAT Required ................................ ENV4_4.DAT
### 4.5 SCIENCE REQUIREMENTS ENVELOPE

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5.1</td>
<td>General Description of Experiment Sample Requirements</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Range of Atmospheric Requirements (Pressure, Gas Composition, Humidity, Vacuum)</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Temperature Control, Measurement Range, and Accuracy Required</td>
</tr>
<tr>
<td>4.5.4</td>
<td>Vibration Control, Measurement Range, Accuracy and Frequency of Measurements Required</td>
</tr>
<tr>
<td>4.5.5</td>
<td>Typical Test Matrices (Number and Duration of Test Required)</td>
</tr>
<tr>
<td>4.5.6</td>
<td>Imaging Requirements Envelope (Photography, Radiography; Television; Resolution and Frame Rate)</td>
</tr>
<tr>
<td>4.5.7</td>
<td>Electromagnetic Limitations for Type of Experimentation</td>
</tr>
<tr>
<td>4.5.8</td>
<td>Typical Astronaut Involvement (Extravehicular Activity [EVA], Activation of Experiment)</td>
</tr>
<tr>
<td>4.5.9</td>
<td>Typical Data Requirements</td>
</tr>
<tr>
<td>4.5.10</td>
<td>Telepresence, Telerobotics Requirements Telepresence Requirements</td>
</tr>
<tr>
<td></td>
<td>Telerobotics Requirements</td>
</tr>
</tbody>
</table>

### 4.6 OTHER REQUIREMENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6.1</td>
<td>Other Applicable Material Not Addressed in These Requirements</td>
</tr>
</tbody>
</table>
APPENDIX B

Text Editor Commands
FORMATTING COMMANDS:

^B .................................. Reformat Paragraph
^OR .................................. Set Right Margin
^QI .................................. Toggle Autoindent Mode
^QW .................................. Toggle Word Wrap

FIND AND REPLACE COMMANDS:

^QA .................................. Find and Replace a String
^QF .................................. Find an Occurrence of a String
^L .................................. Find the Next Occurrence
APPENDIX F

Project Plan Software Listing
This is the Automated Paylaod Experiment Tool, a knowledge-based system to aid in the development of NASA documentation for pre-flight planning and control.

no_edit_key ()
no_debug ()
column = 3.
row = 3.

action = '

nasaloop = 1.
glossary_load = 0.
yn = [YES,NO].

while ?action <> 'Exit System'
then do (mainmenu).

topic 'mainmenu'.
window (,,red,yellow,?column,?row,76,17).
choices = ['How to use the System','Project Selection','Project Plan (Overview)','Glossary/Acronyms','Print Glossary/Acronyms','Exit System'].

set_number_of_values (action,1).

ask ('#e Please select the activity of your choice, or choose Exit to leave the system.',action,?choices).

close_window ()

if ?action = 'How to use the System'
then new_kb ('ppintro.hkb').

if ?action = 'Project Selection'
then new_kb ('ppproject.hkb').

if ?action = 'Project Plan (Overview)'
then new_kb ('ppout.hkb').

if ?action = 'Glossary/Acronyms'
then
glossary_load = (?glossary_load + 1)
and
do (glossary).

if ?action = 'Exit System'
then exit ()

if ?action = 'Print Glossary/Acronyms'
then
ask ('#e
The printing of the glossary/acronym list can require a significant amount of time (5-10 minutes depending on your system). Are you sure you want to print the glossary at this time?',printok,?yn)

and
if ?printok = NO
then new_kb ("NASAPLAN.CKB")
else
window (,white,red,yellow,1,16,27,4)
and
WRITE ('con:',
'GLOSSARY is being printed.

Please stand by.  
')
and
glossary_print is read ('PPTERMS.DAT','//KSC')
and
glossary_print is string_replace(?glossary_print, '//', '')
and
glossary_print is string_replace(?glossary_print, '/end', '')
and
print (?glossary_print)
and
glossary_print is read ('PPTERMS.DAT','//KSC')
and
glossary_print is string_replace(?glossary_print, '//', '')
and
glossary_print is string_replace(?glossary_print, '/end', '')
and
print (?glossary_print)
and
close_window ()
end.  (* glossary *)

if ?glossary_load = 1
then
window (,white,red,yellow,1,16,27,4)
and
WRITE ('con:',
'A slight delay will occur while the glossary is loaded.  
Please stand by.  
')
and
glossary_text is read ('PPINDEX.DAT')
and
close_window ()
and
close ('PPTERMS.DAT').
say (?glossary_text).
close_window ()
end.  (* glossary *)

topic mark (find_string).
text is read ('PPTERMS.DAT',concat('//',?find_string),'/end').
window (?find_string,blue,white,white,,72,).
say (?text).
close_window ()

end.

end. (* mainmenu *)
This is the introductory screen for the NASA Automated Payload Element Tool. It is used to give the novice user a brief tour of the functions of the system.

No edit key (.).
No debug ()

Yn is [YES,NO].
Column = 3.
Row = 3.

Tried = 0.

Do (so_you_want_to_fly).
New_kb ('nasaplan.ckb').
Topic so_you_want_to_fly.
Say ('

#Bmagenta So you want to fly on the Shuttle. #D

Well, before you can, we must get a little information about your experiment and its objectives.

If you have flown with us in the past, you may remember a substantial amount of paper documentation was required. This application, the Automated Payload Experiment Tool, is designed to alleviate much of the burden of experiment preparation by utilizing a hypertext, knowledge-based system. This system can be used to prepare one of our support documents, the Project Plan, which describes the overall plan for proceeding with a project.

Press #Fyellow SPACE#D to continue.').

If ?tried = 0 then
  Column = ?column + 1
  And
  Row = ?row + 1
  And
  Window (' ',white,red,white)
  And

  Say ( '#e
  For more information on a highlighted topic, just move the mouse to that word and click. The information will immediately be displayed. If you are not using a mouse, please use the function keys as indicated at the bottom of the screen.

  For multiple page definitions, please use the #Fyellow Page Up#D and #Fyellow Page Down#D keys to scroll back and forth through the pages. Multiple page displays are indicated by the #Fyellow Page x of x #D message at the lower right of the screen.

  For help at anytime throughout the application, select the #Fyellow F1#D key. This will retrieve location sensitive
help information, and may be called from the system or system-called edit screens.

This will be the method by which support documentation will be retrieved throughout this application.

Press #fyellow SPACE#D to continue.

and
close_window ()
and
tried = 1
and
column = ?column - 1
and
row = ?row - 1
and
do (so_you_want_to_fly).

topic mark (find_string).
column = ?column + 1.
row = ?row + 1.
text is read ('PPTERMS.dat',concat('/','?find_string'),'/end').
window (?find_string,blue,white,white,?column,?row,72,).
say (?text).
column = ?column - 1.
row = ?row - 1.
close_window ()
end. (* mark *)

end. (* so_you_want_to_fly *)
(* PPOUT.KB  This is the outline for appendix E of the MSAD Project Plans *)

no_edit_key ().
no_debug ().

column = 3.
row = 3.
menu_choice = ' '.
menu_option = ['GENERAL', 'PREPARATION AND APPROVAL', 'CHANGES', 'CONTENT', 'RETURN TO PREVIOUS MENU'].

while ?menu_choice <> 'RETURN TO PREVIOUS MENU'
    then do (outline).

topic 'outline'.
ask('#e

OUTLINE FOR PREPARING
MSAD PROJECT PLANS

Which section do you wish to display?', menu_choice, ?menu_option).

if ?menu_choice = 'GENERAL'
    then do ('General').

if ?menu_choice = 'PREPARATION AND APPROVAL'
    then do ('Preparation and Approval').

if ?menu_choice = 'CHANGES'
    then do ('Changes').

if ?menu_choice = 'CONTENT'
    then new_kb ('dispcont.hkb').

if ?menu_choice = 'RETURN TO PREVIOUS MENU'
    then new_kb ('nasaplan.ckb').

end. (* outline *)

(*================================ THREADED TOPICS ===================================*)

topic 'General'.
window ('1.0  General', blue, white, white, ?column, ?row, 76, 18).
say(['

The Microgravity Science and Applications Division (MSAD) requires that a MSAD Project Plan be submitted and approved prior to making a major commitment of resources to an MSAD project. MSAD Project Plans are to be prepared in final draft form for the Requirements Definition Review #40RDR#41.

Press #fyellow SPACE#d to continue']].)

close_window ().
end. (* General *)
Plans will be prepared and submitted for all flight experiments. Project plans will be reissued, modified, or amended for reflights depending on the complexity of the task. A plan's preparation is the responsibility of the designated Project Manager at the responsible NASA center. The Project Manager will sign the MSAD Project Plan as the preparer; the Project Scientist and the Principal Investigator will sign as concurring. The MSAD Project Plan will be signed off at the NASA center prior to submission to Headquarters by the appropriate center's authorities. When the Program Scientist and Program Manager sign to register their concurrence, the MSAD Project Plan will be submitted to the MSAD Director for approval.

Press #fyellow SPACE#d to continue').

The Project Manager is responsible for updating a MSAD Project Plan when significant changes occur (such as changes in scope, organization, or roles and responsibilities). This does not apply to resources, schedules or manpower, which are updated through normal budgeting and project monitoring activities. The Project Manager will establish a change control process for maintaining the MSAD Project Plan and other project documentation.

Press #fyellow SPACE#d to continue'}.
This is the project menu to allow the user to define a new project or select an existing project. It then calls the appropriate submenu.

no_edit_key ().
no_debug ().
do_gloss = 1.
yn is [YES, NO].
projlist is ''. 
do (firstpass).

if ?project want = 'RETURN TO MAIN MENU'
then new_kb ('nasaplan.ckb').

topic 'firstpass'.
eof = number_to_char (26).
projtest is read_line ('pproj.dat').
if ?projtest = ?eof
then
d0 (new_project)
else
projlist is read ('pproj.dat')
and
do (old_project).

topic 'new_project'.

window (,white,red,yellow,5,5,75,16).
read_response ('#e
Please enter an identifier for your project. This identifier should be eight (8) characters or less. #n',newproject).

newproject = string_replace(?newproject, ' ', ',',8).

then
projlist gets ?newproject
and
new_file ('pproj.dat')
and
write ('pproj.dat',#o,?projlist)
and
close ('pproj.dat')
and
project_want = ?newproject
and
cur_dir = string_replace(?project_want,' ','',10)
and
new_file ('CURDIR.DAT')
and
write ('CURDIR.DAT',?cur_dir)
and
close ('CURDIR.DAT')
and
DOSCOMMAND = CONCAT('MD ',?NEWPROJECT)
and
dos (?DOSCOMAND,restore)
else
  say ('#e
    Sorry, the identifier for your project must be a valid DOS
    name, i.e., eight (8) characters or less.

    Please press #yellow SPACE#d and begin again. ')

  and
  new_kb ('pproject.hkb').
close_window ().
end. (* new_project *)

topic 'old_project'.

window (,white, red, yellow, 5,5,75,16).
choose_project = ?projlist.
choose_project gets 'ENTER A NEW PROJECT'.
choose_project gets 'DELETE AN OLD PROJECT'.
choose_project gets 'RETURN TO MAIN MENU'.
ask ('#e
  Please select the project of your choice, or enter a new
  project.' ,project_want ,?choose_project).

if ?project_want = '' or ?project_want = [ ] or ?project_want = ''
  then do (new_project).

if ?project_want = 'DELETE AN OLD PROJECT'
  then do (kill_project).

if ?project_want = 'RETURN TO MAIN MENU'
  then new_kb ('nasaplan.ckb').

if ?project_want = 'ENTER A NEW PROJECT'
  then do (new_project)
else
  cur_dir = string_replace(?project_want, , ,
  and
  new_file ('CURDIR.DAT')
  and
  write ('CURDIR.DAT',?cur_dir)
  and
  close ('CURDIR.DAT').
close_window ().
end. (* old_project *)
end. (* firstpass *)

new_kb ('ppnasam.hkb').

topic 'kill_project'.
close ('pproj.dat').
deletename = ''. 
window (,white,red,yellow,5,5,75,16).
ask ('#e

  You have chosen to delete a project. This will erase all data
files for the project from your hard drive, plus will remove
the project from the list of available projects. This deletion
is permanent and cannot be undone; therefore, use this option
with CAUTION.

Do you wish to proceed with the project deletion? ',
deleteok,?yn).

if ?deleteok = YES
  then
    read_response ('#e
    Please enter the project identifier exactly as it appears in
    the project selection list. This identifier should be eight
    (8) characters or less. #n',killproj).

oldlist is string_to_list (?projlist).

if ?deleteok = YES
  then
    deletename is intersect(?killproj,?oldlist).

deletefn1 = NO.

if ?deleteok = YES
  then
    if ?deletename <> ' ' and ?deletename <> '' and ?deletename <> [ ]
    then
      ask ([#'e
        This is your FINAL WARNING.
        #s
        Do you want to delete the project: ',?deletename,'?',deletefn1,?yn)
      else
        say ('#e
        This project was not found. Please be sure to type
        project title as it appears on the list, i.e. using
        appropriate upper- and lower-case letters.

        Press #yellow SPACE#d to continue.').

if ?deletefn1 = YES
  then
    oldlist is remove (?oldlist,?deletename)
    and
    new_file ('PPROJ.DAT')
    and
    write ('PPROJ.DAT',?oldlist)
    and
    close ('PPROJ.DAT')
    and
    doscommand = concat ('ERASE C:\GARDEN\',?deletename,\'*.*')
    and
    dos (?doscommand,restore)
    and
    doscommand = concat ('RD ',?deletename)
    and
    dos (?doscommand,restore).
  close_window ()
  new_kb ('PPROJECT.HKB').
end. (* kill_project *)

(* ********************************************* *)

end. (* nasamenu *)
This is the activity menu to allow the user to select an activity to perform on an existing project.

no_edit_key().
no_debug().
do_gloss = 1.
today = date().
month = element(?today,1).
day = element(?today,2).
year = element(?today,3).
today = concat(month,'/',day,'/',year).
yn is [YES,NO].
curdir is read_line('CURDIR.DAT').
close(concat('C:\GARDEN\',?CURDIR,'\BASELINE.DAT')).
curdir is string_replace(?curdir,','').
eof = number_to_char(26).

menu_option is ['FILL OUT DOCUMENTATION','PRINT PROJECT PLAN','DISPLAY PROJECT PLAN','DETERMINE IF PROJECT PLAN IS COMPLETE','BASELINE DOCUMENT','COMPARE BASELINE TO CURRENT REVISION','COPY ANSWERS TO DISK','RETURN TO PREVIOUS MENU','RETURN TO MAIN MENU','EXIT SYSTEM'].
menu_choice = ' '.

while ?menu_choice <> 'EXIT SYSTEM'
    then do (nasamenu).

topic nasamenu.

    ask(['
        The project you have selected is: ',?curdir,'#d
        Please enter your choice of activities from the list.'],menu_choice,
        ?menu_option).

if ?menu_choice = 'FILL OUT DOCUMENTATION'
    then new_kb('ppfillm.hkb').

if ?menu_choice = 'RETURN TO PREVIOUS MENU'
    then new_kb('PPROJECT.HKB').

if ?menu_choice = 'RETURN TO MAIN MENU'
    then new_kb('NASAPLAN.CKB').

if ?menu_choice = 'PRINT PROJECT PLAN'
    then new_kb('PPPRINT.HKB').

if ?menu_choice = 'BASELINE DOCUMENT'
    then do (baseline_rtn).

if ?menu_choice = 'DISPLAY PROJECT PLAN'
    then new_kb('DISPPROJ.HKB').

if ?menu_choice = 'COPY ANSWERS TO DISK'
    then do (copyfiles).

if ?menu_choice = 'COMPARE BASELINE TO CURRENT REVISION'
    then do (compare_rtn).
if ?menu_choice = 'DETERMINE IF PROJECT PLAN IS COMPLETE'
    then new_kb ('ppsearch.hkb').

if ?menu_choice = 'EXIT SYSTEM'
    then stop ().

topic 'copyfiles'.

ask ('Do you want to copy your answers to a different drive?', wantcopy, ?yn).
if ?wantcopy = YES
    then drivelist is [A:, B:, C:, D:, NONE]
and
    ask
        ('Please choose the drive to which you wish to copy the files: ',
         drive_destination, ?drivelist).

if ?wantcopy = YES and ?drive_destination <> NONE
    then
        copy_command = concat ('COPY C:\GARDEN\', ?curdir, '\*.DAT ', ?DRIVE_DESTINATION
and
        say ('Please insert diskette now if you are copying to a floppy drive.

        Please press #eYELLOW SPACE#d when ready. ')

        and
        move_cursor (1, 10)

        and
        dos (?copy_command, restore)

        and
        say ('#e

        Answers have been moved to drive #s', ?drive_destination, ' #n

        Please press #eYELLOW SPACE#d to exit. ')

    do (nasamenu).

end. (* copyfiles *)

topic 'baseline_rtn'.

ask ('#e

At some point in the documentation process, it is necessary to
declare that all documents are complete, and that any changes
to be made will be treated as revisions to the baseline document.

Do you want to baseline your answers at this time?', baseline, ?yn).

curbase = ?eof.
overwrite = YES.
if ?baseline = YES
    then
        curbase is read_line (concat('C:\GARDEN\', ?CURDIR, '\BASELINE.DAT'))

and
        close (concat('C:\GARDEN\', ?CURDIR, '\BASELINE.DAT'))

and
if ?curbase <> ?eof
    then
        ask
            ('#e
            You have already baselined this experiment in the past. Do you
            want to take all revisions and overwrite your previous baseline
            to create a new baseline? ', overwrite, ?yn).

if ?curbase = ?eof and ?baseline = YES
    then
        md_command = concat ('MD C:\GARDEN\', ?curdir, '\BASELINE')
        and
            dos (?md_command, restore)
        and
            copy_command = concat
                ('COPY C:\GARDEN\', ?CURDIR, '*.DAT C:\GARDEN\', ?CURDIR, '\BASELINE\*.*')
        and
            dos (?copy_command, restore)
        and
            write (concat('C:\GARDEN\', ?CURDIR, '\BASELINE.DAT'), ?today)
        and
            say ('#e
            Baseline document has been created. All changes to this
document will be stored in the revision. A new baseline
must be created to incorporate any revisions into the
final document.

Please press #fyellow SPACE#d to exit.  ')

if ?curbase <> ?eof and ?baseline = YES and ?overwrite = YES
    then
        xcopy_command = concat
        and
            dos (?xcopy_command, restore)
        and
            new_file (concat('C:\GARDEN\', ?CURDIR, '\BASELINE.DAT'))
        and
            write (concat('C:\GARDEN\', ?CURDIR, '\BASELINE.DAT'), ?today)
        and
            say ('#e
            All revisions have been incorporated in the baseline
document. Addition changes to this document will be
stored in a new revision. A new baseline must be
created to incorporate any new revisions into the
final document.

Please press #fyellow SPACE#d to exit.  ')

do (nasamenu).
end. (* baseline_rtn *)
topic 'compare_rtn'.
curbase is read_line (concat('C:\GARDEN\',?CURDIR,'\BASELINE.DAT'))
and
close (concat('C:\GARDEN\',?CURDIR,'\BASELINE.DAT'))
and
if ?curbase = ?eof
then
  say
  ('#e
   You have not yet baselined this experiment; therefore, no comparison is necessary.
   
   Press #fyellow SPACE#d to continue')

and
new_kb ('ppnasam.HKB').

comp_choices = ['RUN COMPARISON PROGRAM','DISPLAY COMPARISONS','PRINT COMPARISONS','RETURN TO PREVIOUS MENU'].

ask ('#e

Do you wish to run the comparison program to generate a new listing of differences between the baseline and revision, print or display the results of the most recent comparison, or exit this menu?',comp_ans,?comp_choices).

if ?comp_ans = 'RETURN TO PREVIOUS MENU'
then new_kb ('NASAMENU.HKB').

if ?comp_ans = 'RUN COMPARISON PROGRAM'
then
do (comp_pgm).

topic 'comp_pgm'.
  comp_command = concat
  ('FC /a C:\GARDEN\',?CURDIR,'\*.DAT C:\GARDEN\',?CURDIR,'\BASELINE\*. > C:\GARDEN\',?CURDIR,'\DIFFER.DAT').
  dos (?comp_command,restore).
  close (concat('C:\GARDEN\',?CURDIR,'\DIFFER.DAT,'))
  
say ('#e
     Files have been compared. Please use the display or print options to view the results of the comparison.
     
     Press #fyellow SPACE#d to continue.
   )

end. (* comp_pgm *)

if ?comp_ans = 'DISPLAY COMPARISONS'
then
  comp_file = read (concat(C:\GARDEN\',?CURDIR,'\DIFFER.DAT'))
  and
  say (?comp_file).

if ?comp_ans = 'PRINT COMPARISONS'
then
  comp_file = read (concat(C:\GARDEN\',?CURDIR,'\DIFFER.DAT'))
  and
\[ \text{print (p,\text{comp_file},p).} \]
\[ \text{do (nasamenu).} \]
\[ \text{end. (\text{* compare_rtn *} *)} \]
\[ \text{end. (\text{* nasamenu *} *)} \]
This program is used to search for the necessary data files to complete the Project Plan, and notify the user of any missing sections.

yn = [YES, NO].
column = 3.
row = 3.
eof = number_to_char (26).
no_edit_key ()
no_debug ()
curdir is read_line ('CURDIR.DAT').
curdir = string_replace(curdir,' ','',8).
incomplete_sections = []
completed_sections = 0.
total_files = 25.
do (search).
close_window ()

if completed_sections = total_files
then
say ('All questions have been answered. The Project Plan is now ready to print.

Press [yellow]SPACE[red]d to continue')
else
say ('The following topics have not been answered. The Project Plan is not ready for final printing.

Press [yellow]SPACE[red]d to continue').
new_kb ('PPNASAM.HKB')
topic 'search'.
window (,white,red,yellow,1,16,27,4).
WRITE ('con:',
'Database is being searched.

Please stand by.')

readfile = concat('C:\GARDEN\', curdir, '\PPE4_1.DAT')
testfile = read_line(readfile).
do ('tally_rtn')
if testfile = eof
then incomplete_sections gets '4.1 Introduction'.

close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\PPE4_2.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
    then incomplete_sections gets '4.2 Objectives'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\PPE4_3.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
    then incomplete_sections gets '4.3 Science Requirements'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\PPE4_4_1.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
    then incomplete_sections gets '4.4.1 Experiment Hardware Description'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\PPE4_4_2.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
    then incomplete_sections gets '4.4.2 Payload Classification'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\PPE4_4_3.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
    then incomplete_sections gets '4.4.3 Development Approach'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\PPE4_4_4.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
    then incomplete_sections gets '4.4.4 Technology Plan'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\PPE4_4_5.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
    then incomplete_sections gets '4.4.5 Logistics'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\PPE4_4_6.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
    then incomplete_sections gets
    '4.4.6 Mission Operations Training & Data Management'.
close (?readfile).
readfile = concat('C:\GARDEN\',?CURDIR, '\PPE4_4_7.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
  then incomplete_sections gets
    '4.4.7 Analysis of Mission Results'.
close (?readfile).

readfile = concat('C:\GARDEN\',?CURDIR, '\PPE4_4_8.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
  then incomplete_sections gets
    '4.4.8 Facilities'.
close (?readfile).

readfile = concat('C:\GARDEN\',?CURDIR, '\PPE4_4_9.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
  then incomplete_sections gets
    '4.4.9 Safety'.
close (?readfile).

readfile = concat('C:\GARDEN\',?CURDIR, '\PPE4_5_1.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
  then incomplete_sections gets
    '4.5.1 Implementation Approach'.
close (?readfile).

readfile = concat('C:\GARDEN\',?CURDIR, '\PPE4_5_2.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
  then incomplete_sections gets
    '4.5.2 Summary Work Breakdown Structure'.
close (?readfile).

readfile = concat('C:\GARDEN\',?CURDIR, '\PPE4_5_3.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
  then incomplete_sections gets
    '4.6.2 Mission Management Responsibilities and Organization'.
close (?readfile).

readfile = concat('C:\GARDEN\',?CURDIR, '\PPE4_7.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
then incomplete_sections gets
'4.7 Schedule'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\PPE4_8_1.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
  then incomplete_sections gets
    '4.8.1 Resources'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\PPE4_8_2.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
  then incomplete_sections gets
    '4.8.2 Cost Control Guidelines'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\E4_8_3_1.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
  then incomplete_sections gets
    '4.8.3.1 NASA Reports'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\E4_8_3_2.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
  then incomplete_sections gets
    '4.8.3.2 Contractor Reports'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\PPE4_8_4.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
  then incomplete_sections gets
    '4.8.4 Cost Control Strategy'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\E4_9_1_1.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
  then incomplete_sections gets
    '4.9.1.1 Internal Reviews'.
close (?readfile).

readfile = concat('C:\GARDEN\', ?CURDIR, '\E4_9_1_2.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
if ?testfile = ?eof
  then incomplete_sections gets
    '4.9.1.2 External Reviews'.
close (?readfile).
readfile = concat('C:\GARDEN\', ?CURDIR, '\E4_9_1_3.DAT').
testfile = read_line(?readfile).
do ('tally_rtn').
   if ?testfile = ?eof
      then incomplete_sections gets
          '4.9.1.3 Design and Readiness Reviews'.
      close (?readfile).

   topic 'tally_rtn'.
      if ?testfile <> ?eof
         then completed_sections = ?completed_sections + 1.
   end. (* tally_rtn *)
end. (* search *)
(

(* DISPPROJ.HKB is used to display all sections of the PROJECT PLAN. *)

no_edit_key ()
no_debug()
choice = []
eof = number_to_char (26).
curdir is read_line ('CURDIR.DAT')
curdir is string_replace(?curdir,' ','',8)
blankline = '
'.
line_count = 0.
do (display_Routine).

new_kb ('PPNASAM.HKB').

topic 'display_Routine'.
sections = ['i.0 INTRODUCTION',
          '2.0 OBJECTIVE',
          '3.0 SCIENCE REQUIREMENTS',
          '4.0 TECHNICAL PLAN',
          '5.0 IMPLEMENTATION PLAN',
          '6.0 MANAGEMENT PLAN',
          '7.0 SCHEDULE',
          '8.0 COST CONTROL PLAN',
          '9.0 PROJECT REVIEWS AND MEETINGS',
          'QUIT'].

window ('Display Project Plan',white,blue,white,3,3,78,18).
while ?choice <> QUIT
    then do (display_PPLAN).
close_window ()
close_all ()

topic 'display_PPLAN'.
pplan_page = []
ask ('#e

Which section do you want to display?',choice,?sections).

if ?choice = '1.0 INTRODUCTION'
    then do ('1.0 INTRODUCTION').

if ?choice = '2.0 OBJECTIVE'
    then do ('2.0 OBJECTIVE').

if ?choice = '3.0 SCIENCE REQUIREMENTS'
    then do ('3.0 SCIENCE REQUIREMENTS').

if ?choice = '4.0 TECHNICAL PLAN'
    then do ('4.0 TECHNICAL PLAN').

if ?choice = '5.0 IMPLEMENTATION PLAN'
    then do ('5.0 IMPLEMENTATION PLAN').

if ?choice = '6.0 MANAGEMENT PLAN'
    then do ('6.0 MANAGEMENT PLAN').

if ?choice = '7.0 SCHEDULE'
    then do ('7.0 SCHEDULE').
if ?choice = '8.0 COST CONTROL PLAN'
    then do ('8.0 COST CONTROL PLAN').

if ?choice = '9.0 PROJECT REVIEWS AND MEETINGS'
    then do ('9.0 PROJECT REVIEWS AND MEETINGS').

if ?choice = QUIT
    then new_kb ('PPNASAM.HKB').

topic '1.0 INTRODUCTION'.
    page_count = 0.
    pplan_page gets ' 1.0 INTRODUCTION'.
    pplan_page gets ?blankline.
    line_count = 4.
    while ?line <> ?eof
        then do (read_file).
    close (?filename).
    pplan_page gets ?blankline.
    line_count = ?line_count + i.
    do (page_advance).

end. (* 1.0 INTRODUCTION *)

topic '2.0 OBJECTIVE'.
    page_count = 0.
    filename = concat(C:\GARDEN,?CURDIR,'\PPE4_2.DAT').
    line is read_line (?filename).
    if ?line_count > 54
        then do (page_advance).
    pplan_page gets ?blankline.
    pplan_page gets ?blankline.
    pplan_page gets ' 2.0 OBJECTIVE'.
    pplan_page gets ?blankline.
    line_count = ?line_count + 4.
    while ?line <> ?eof
        then do (read_file).
    close (?filename).
    pplan_page gets ?blankline.
    line_count = ?line_count + i.
    do (page_advance).

end. (* 2.0 OBJECTIVE *)

topic '3.0 SCIENCE REQUIREMENTS'.
    page_count = 0.
    filename = concat(C:\GARDEN,?CURDIR,'\PPE4_3.DAT').
    line is read_line (?filename).
    if ?line_count > 54
        then do (page_advance).
    pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 3.0 SCIENCE REQUIREMENTS'.
pplan_page gets ?blankline.
line_count = ?line_count + 4.
while ?line <> ?eof
  then do (read_file).
  close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
do (page_advance).
end. (* 3.0 SCIENCE REQUIREMENTS *)

topic '4.0 TECHNICAL PLAN'.
  page_count = 0.
  filename = concat(C:\GARDEN,?CURDIR,'\PPE4_4_1.DAT').
  line is read_line (?filename).
  if ?line_count > 54
    then do (page_advance).

  pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 4.0 TECHNICAL PLAN'.
pplan_page gets ?blankline.
pplan_page gets ' 4.1 Experiment Hardware Description'.
pplan_page gets ?blankline.
line_count = ?line_count + 6.
while ?line <> ?eof
  then do (read_file).
  close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
  then do (page_advance).

  pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 4.2 Payload Classification'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN,?CURDIR,'\PPE4_4_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
  close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

  pplan_page gets ' 4.3 Development Approach'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN,?CURDIR,'\PPE4_4_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
  close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).
then do (page_advance).

pplan_page gets ' 4.4 Technology Plan'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\PPE4_4.4.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

pplan_page gets ' 4.5 Logistics'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\PPE4_4.5.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

pplan_page gets 4.6 Mission Operations Training and Data Management'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\PPE4_4.6.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

pplan_page gets ' 4.7 Analysis of Mission Results'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\PPE4_4.7.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

pplan_page gets ' 4.8 Facilities'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\PPE4_4_8.DAT').
line is read line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
do (page_advance).

pplan_page gets ' 4.9 Safety'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\PPE4_4_9.DAT').
line is read line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
do (page_advance).

end. (* 4.0 TECHNICAL PLAN *)

topic '5.0 IMPLEMENTATION PLAN'.
page_count = 0.
filename = concat(C:\GARDEN\, ?CURDIR, '\PPE4_5_1.DAT').
line is read line (?filename).
if ?line_count > 54
    then do (page_advance).
pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 5.0 IMPLEMENTATION PLAN'.
pplan_page gets ?blankline.
pplan_page gets ' 5.1 Implementation Approach'.
pplan_page gets ?blankline.
line_count = ?line_count + 6.
while ?line <> ?eof
    then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 54
    then do (page_advance).

pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 5.2 Summary Work Breakdown Structure'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\PPE4_5_2.DAT').
line is read line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).
pplan_page gets 5.3 Documentation.

*pplan_page gets ?blankline.*

line_count = ?line_count + 2.
filename = concat(C:\GARDEN, ?CURDIR, \\
\"PPE4_5_3.DAT\")
line is read_line (?filename).
while ?line <> ?eof
then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
do (page_advance).

end. (* 5.0 IMPLEMENTATION PLAN *)

topic '6.0 MANAGEMENT PLAN'.

page_count = 0.
filename = concat(C:\GARDEN, ?CURDIR, \\
\"PPE4_6_1.DAT\")
line is read_line (?filename).
if ?line_count > 48
then do (page_advance).

pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 6.0 MANAGEMENT PLAN'.
pplan_page gets ?blankline.
pplan_page gets ' 6.1 Project Management Responsibilities and Organization'.
pplan_page gets ?blankline.
line_count = ?line_count + 6.
while ?line <> ?eof
then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
then do (page_advance).

end. (* 6.0 MANAGEMENT PLAN *)

pplan_page gets 6.2 Mission Management Responsibilities and Organization'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN, ?CURDIR, \\
\"PPE4_6_2.DAT\")
line is read_line (?filename).
while ?line <> ?eof
then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
do (page_advance).

end. (* 6.0 MANAGEMENT PLAN *)

topic '7.0 SCHEDULE'.

page_count = 0.
filename = concat(C:\GARDEN, ?CURDIR, \\
\"PPE4_7.DAT\")
line is read_line (?filename).
if ?line_count > 56
  then do (page_advance).

pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 7.0 SCHEDULE'.
pplan_page gets ?blankline.
line_count = ?line_count + 4.
while ?line <> ?eof
  then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
do (page_advance).
end. (* 7.0 SCHEDULE *)

topic '8.0 COST CONTROL PLAN'.
page_count = 0.
filename = concat(C:\GARDEN\,?CURDIR, '\PPE4_8_1.DAT').
line is read_line (?filename).
if ?line_count > 54
  then do (page_advance).

pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 8.0 IMPLEMENTATION PLAN'.
pplan_page gets ' 8.1 Resources'.
pplan_page gets ?blankline.
line_count = ?line_count + 6.
while ?line <> ?eof
  then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
  then do (page_advance).

pplan_page gets ' 8.2 Summary Work Breakdown Structure'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\PPE4_8_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
  then do (page_advance).

pplan_page gets ' 8.3 Cost Reporting and Control Structure'.
pplan_page gets ?blankline.
pplan_page gets ' 8.3.1 Nasa Reports'.

pplan_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\\GARDEN\\, ?CURDIR, '\E4_8_3_1.DAT').
pplan_page gets ?blankline.
line is read_line (?,filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

pplan_page gets ' 8.3.2 Contractor Reports'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\\GARDEN\\, ?CURDIR, '\E4_8_3_2.DAT').
pplan_page gets ?blankline.
line is read_line (?,filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

pplan_page gets ' 8.4 Cost Control Strategy'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\\GARDEN\\, ?CURDIR, '\PPE4_8_4.DAT').
pplan_page gets ?blankline.
line is read_line (?,filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
do (page_advance).

end. (* 8.0 COST CONTROL PLAN *)

topic '9.0 PROJECT REVIEWS AND MEETINGS'.
page_count = 0.
filename = concat(C:\\GARDEN\\, ?CURDIR, '\E4_9_1_1.DAT').
line is read_line (?,filename).
if ?line_count > 56
  then do (page_advance).

pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 9.0 PROJECT REVIEWS AND MEETINGS'.
pplan_page gets ?blankline.
pplan_page gets ' 9.1 Management Reviews'.
pplan_page gets ?blankline.
pplan_page gets ' 9.1.1 Internal Reviews'.
pplan_page gets ?blankline.
line_count = ?line_count + 8.
while ?line <> ?eof
then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
    then do (page_advance).

pplan_page gets ' 9.1.2 External Reviews'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,'\E4_9_1_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
    then do (page_advance).

pplan_page gets ' 9.1.3 Design and Readiness Reviews'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,'\E4_9_1_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
do (page_advance).

end. (* 9.0 PROJECT REVIEWS AND MEETINGS *)
end. (* display_PPLAN *)

topic 'page_advance'.
    page_count = ?page_count + 1.
    window_name = concat(?choice,' Page: ',?page_count).
    window('?window_name,white,blue,white,3,3,78,19).
    say (?pplan_page).
pplan_page = [].
pplan_page gets ?blankline.
pplan_page gets ?blankline.
line_count = 2.
collect ().
close_window ().
end. (* page_advance *)

topic 'read_file'.
    if ?line_count > 58
        then do (page_advance).
        line = concat(' ',?line).
pplan_page gets ?line.
line is read_line (?filename).
line_count = ?line_count + 1.
end. (* read_file *)

end. (* display_Routine *)
(* PPPRINT.KB is used to print all sections of the Project Plan. *)

no_edit_key ().
no_debug ().
choice = ' '.
eof = number_to_char (26).
curdir is read_line ('CURDIR.DAT').
curdir is string_replace(?curdir,' ','/',8).
blankline = ' '.
line_count = 0.
page_count = 0.
do (Print_Routine).

new_kb ('PPNASAM.HKB').
topic 'Print Routine'.

sections = ['1.0 INTRODUCTION',
            '2.0 OBJECTIVE',
            '3.0 SCIENCE REQUIREMENTS',
            '4.0 TECHNICAL PLAN',
            '5.0 IMPLEMENTATION PLAN',
            '6.0 MANAGEMENT PLAN',
            '7.0 SCHEDULE',
            '8.0 COST CONTROL PLAN',
            '9.0 PROJECT REVIEWS AND MEETINGS',
            'PRINT ENTIRE DOCUMENT',
            'QUIT'].

window ('Print Project Plan',white,blue,white,3,3,78,18).
while ?choice <> QUIT then do (print_sections).
close window ()
close all ()
topic 'print_sections'.
pplan_page = []
ask ('#e
Which section do you wish to print?',choice,?sections).

if ?choice = '1.0 INTRODUCTION'
then
  window (,white,red,yellow,1,16,27,4)
  and
  write ('con:','PRINT IN PROGRESS...')
  and
  do ('1.0 INTRODUCTION')
  and
  if line_count > 2
  then do (page_advance).
  close_window ()

if ?choice = '2.0 OBJECTIVE'
then
  window (,white,red,yellow,1,16,27,4)
and write ('con:','PRINT IN PROGRESS...')
and do ('2.0 OBJECTIVE')
and if line_count > 2
then do (page_advance).
close_window ()

if ?choice = '3.0 SCIENCE REQUIREMENTS'
then
window ((white,red,yellow,1,16,27,4)
and
write ('con:','PRINT IN PROGRESS...')
and
do ('3.0 SCIENCE REQUIREMENTS')
and
if line_count > 2
then do (page_advance).
close_window ()

if ?choice = '4.0 TECHNICAL PLAN'
then
window ((white,red,yellow,1,16,27,4)
and
write ('con:','PRINT IN PROGRESS...')
and
do ('4.0 TECHNICAL PLAN')
and
if line_count > 2
then do (page_advance).
close_window ()

if ?choice = '5.0 IMPLEMENTATION PLAN'
then
window ((white,red,yellow,1,16,27,4)
and
write ('con:','PRINT IN PROGRESS...')
and
do ('5.0 IMPLEMENTATION PLAN')
and
if line_count > 2
then do (page_advance).
close_window ()

if ?choice = '6.0 MANAGEMENT PLAN'
then
window ((white,red,yellow,1,16,27,4)
and
write ('con:','PRINT IN PROGRESS...')
and
do ('6.0 MANAGEMENT PLAN')
and
if line_count > 2
then do (page_advance).
close_window ()

if ?choice = '7.0 SCHEDULE'
then

window (,white,red,yellow,1,16,27,4)
and
write ('con:','PRINT IN PROGRESS...')
and
do ('7.0 SCHEDULE')
and
if line_count > 2
then
do (page_advance).
close_window ()

if ?choice = '8.0 COST CONTROL PLAN'
then
window (,white,red,yellow,1,16,27,4)
and
write ('con:','PRINT IN PROGRESS...')
and
do ('8.0 COST CONTROL PLAN')
and
if line_count > 2
then
do (page_advance).
close_window ()

if ?choice = '9.0 PROJECT REVIEWS AND MEETINGS'
then
window (,white,red,yellow,1,16,27,4)
and
write ('con:','PRINT IN PROGRESS...')
and
do ('9.0 PROJECT REVIEWS AND MEETINGS')
and
if line_count > 2
then
do (page_advance).
close_window ()

If ?choice = 'PRINT ENTIRE DOCUMENT'
then
load ('ppttitle.hkb')
and
do ('print_title_page')
and
remove_topic ('print_title_page')
and
page_count = 0
and
window (,white,red,yellow,1,16,27,4)
and
write ('con:','PRINT IN PROGRESS...')
and
do ('1.0 INTRODUCTION')
and
do ('2.0 OBJECTIVE')
and
do ('3.0 SCIENCE REQUIREMENTS')
and
do ('4.0 TECHNICAL PLAN')
and
do ('5.0 IMPLEMENTATION PLAN')
and
do ('6.0 MANAGEMENT PLAN')
and
do ('7.0 SCHEDULE')
and
do ('8.0 COST CONTROL PLAN')
and
do ('9.0 PROJECT REVIEWS AND MEETINGS')
and
print (#p,' ').

if ?choice = 'QUIT'
then new_kb ('PPNASAM.HKB').
close_window () .
end. (* print_sections *)

topic '1.0 INTRODUCTION'.
pplan_page gets ?blankline.
pplan_page gets ?blankline.
filename = concat(C:\GARDEN\,?CURDIR,\'\PPE4_1.DAT\').
line is read_line (?filename).
pplan_page gets ' 1.0 INTRODUCTION'.
pplan_page gets ?blankline.
line_count = 4.
while ?line <> ?eof
  then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 52
  then do (page_advance).
end. (* INTRODUCTION *)

topic '2.0 OBJECTIVE'.
pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 2.0 OBJECTIVE'.
pplan_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\GARDEN\,?CURDIR,\'\PPE4_2.DAT\').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 52
  then do (page_advance).
end. (* OBJECTIVE *)

topic '3.0 SCIENCE REQUIREMENTS'.
pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 3.0 SCIENCE REQUIREMENTS'.

pplan_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\GARDEN\, ?CURDIR, '\PPE4_3.DAT').
line is read_line (?filename).
while ?line <> ?eof then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 52 then do (page_advance).

end. (* SCIENCE REQUIREMENTS *)

topic '4.0 TECHNICAL PLAN'.
pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 4.0 TECHNICAL PLAN'.
pplan_page gets ?blankline.
pplan_page gets ' 4.1 Experiment Hardware Description'.
pplan_page gets ?blankline.
line_count = ?line_count + 6.
filename = concat(C:\GARDEN\, ?CURDIR, '\PPE4_4_1.DAT').
line is read_line (?filename).
while ?line <> ?eof then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 52 then do (page_advance).

pplan_page gets ' 4.2 Payload Classification'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\PPE4_4_2.DAT').
line is read_line (?filename).
while ?line <> ?eof then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 52 then do (page_advance).

pplan_page gets ' 4.3 Development Approach'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\PPE4_4_3.DAT').
line is read_line (?filename).
while ?line <> ?eof then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 52 then do (page_advance).

pplan_page gets ' 4.4 Technology Plan'.

pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\?CURDIR,'\PPE4_4_4.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 52
    then do (page_advance).

pplan_page gets ' 4.5 Logistics'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\?CURDIR,'\PPE4_4_5.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 52
    then do (page_advance).

pplan_page gets ' 4.6 Mission Operations, Training and Data Management
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\?CURDIR,'\PPE4_4_6.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 52
    then do (page_advance).

pplan_page gets ' 4.7 Analysis of Mission Results'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\?CURDIR,'\PPE4_4_7.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 52
    then do (page_advance).

pplan_page gets ' 4.8 Facilities'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\?CURDIR,'\PPE4_4_8.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 52
    then do (page_advance).

pplan_page gets ' 4.9 Safety'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,'\PPE4_4_9.DAT').
line is read line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 52
    then do (page_advance).

end. (* TECHNICAL PLAN *)

topic '5.0 IMPLEMENTATION PLAN'.
pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 5.0 IMPLEMENTATION PLAN'.
pplan_page gets ?blankline.
pplan_page gets ' 5.1 Implementation Approach'.
pplan_page gets ?blankline.
line_count = ?line_count + 6.
filename = concat(C:\GARDEN\,?CURDIR,'\PPE4_5_1.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 52
    then do (page_advance).

pplan_page gets ' 5.2 Summary Work Breakdown Structure'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,'\PPE4_5_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 52
    then do (page_advance).

pplan_page gets ' 5.3 Documentation'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,'\PPE4_5_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 52
    then do (page_advance).
end. (* IMPLEMENTATION PLAN *)

topic '6.0 MANAGEMENT PLAN'.
pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 6.0 MANAGEMENT PLAN'.
pplan_page gets ?blankline.
pplan_page gets ' 6.1 Project Management Responsibilities and Organizat
pplan_page gets ?blankline.
line_count = ?line_count + 6.
filename = concat (C:\GARDEN\, ?CURDIR, '\PPE4_6_1.DAT').
line is read line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 52
    then do (page_advance).
end. (* MANAGEMENT PLAN *)

pplan_page gets ' 6.2 Mission Management Responsibilities and Organizat
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat (C:\GARDEN\, ?CURDIR, '\PPE4_6_2.DAT').
line is read line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
pplan_page gets ?blankline.
if ?line_count > 52
    then do (page_advance).
end. (* MANAGEMENT PLAN *)

topic '7.0 SCHEDULE'.
pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 7.0 SCHEDULE'.
pplan_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat (C:\GARDEN\, ?CURDIR, '\PPE4_7.DAT').
line is read line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
pplan_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 52
    then do (page_advance).
end. (* SCHEDULE *)
topic '8.0 COST CONTROL PLAN'.
pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets 8.0 COST CONTROL PLAN'.
pplan_page gets ?blankline.
pplan_page gets 8.1 Resources'.
pplan_page gets ?blankline.
line_count = ?line_count + 6.
filename = concat (C:\GARDEN\,?CURDIR,\'\PPE4_8_1.DAT').
line is read_line (?filename).
while ?line <> ?eof
then do (read_file).
close (?filename).
pplan_page gets ?blankline.
if ?line_count > 52
then do (page_advance).
pplan_page gets 8.2 Cost Control Guidelines'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\\GARDEN\\,?CURDIR,\\'\\PPE4_8_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
then do (read_file).
close (?filename).
if ?line_count > 52
then do (page_advance).
pplan_page gets 8.3 Cost Reporting and Control Structure'.
pplan_page gets ?blankline.
pplan_page gets 8.3.1 Nasa Reports'.
pplan_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\GARDEN\,?CURDIR,\'\E4_8_3_1.DAT').
line is read_line (?filename).
while ?line <> ?eof
then do (read_file).
close (?filename).
if ?line_count > 52
then do (page_advance).
pplan_page gets 8.3.2 Contractor Reports'.
pplan_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\\GARDEN\\,?CURDIR,\\'\\E4_8_3_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
then do (read_file).
close (?filename).
if ?line_count > 52
then do (page_advance).
pplan_page gets 8.4 Cost Control Strategy'.
pplan_page gets ?blankline.

line_count = ?line_count + 2.

filename = concat(C:\GARDEN\,?CURDIR,\'\PPE4_8_4.DAT').

line is read_line (?filename).

while ?line <> ?eof
    then do (read_file)
    close (?filename).
    line_count = ?line_count + 1.
    pplan_page gets ?blankline.
    if ?line_count > 52
        then do (page_advance).

end. (* COST CONTROL PLAN *)

topic '9.0 PROJECT REVIEWS AND MEETINGS'.

pplan_page gets ?blankline.
pplan_page gets ?blankline.
pplan_page gets ' 9.0 PROJECT REVIEWS AND MEETINGS'.
pplan_page gets ?blankline.
pplan_page gets ' 9.1 Management Reviews'.
pplan_page gets ?blankline.
pplan_page gets ' 9.1.1 Internal Reviews'.
pplan_page gets ?blankline.

line_count = ?line_count + 8.

filename = concat(C:\GARDEN\,?CURDIR,\'\E4_9_1_1.DAT').

line is read_line (?filename).

while ?line <> ?eof
    then do (read_file)
    close (?filename).
pplan_page gets ?blankline.

line_count = ?line_count + 1.

if ?line_count > 52
    then do (page_advance).

pplan_page gets ' 9.1.2 External Reviews'.

pplan_page gets ?blankline.

line_count = ?line_count + 2.

filename = concat(C:\GARDEN\,?CURDIR,\'\E4_9_1_2.DAT').

line is read_line (?filename).

while ?line <> ?eof
    then do (read_file)
    close (?filename).
    line_count = ?line_count + 1.
    pplan_page gets ?blankline.
    if ?line_count > 52
        then do (page_advance).

pplan_page gets ' 9.1.3 Design and Rediness Reviews'.

pplan_page gets ?blankline.

line_count = ?line_count + 2.

filename = concat(C:\GARDEN\,?CURDIR,\'\E4_9_1_3.DAT').

line is read_line (?filename).

while ?line <> ?eof
    then do (read_file)
    close (?filename).
    line_count = ?line_count + 1.
    pplan_page gets ?blankline.
    do (page_advance).
end. (* PROJECT REVIEWS AND MEETINGS *)

end. (* Print_PPLAN *)

topic 'page_advance'.
    page_count = ?page_count + 1.
    page_line = concat('
', ?page_count).
while ?line_count < 58
    then
        pplan_page gets ?blankline
        and
        line_count = ?line_count + 1.
    if ?choice = 'PRINT ENTIRE DOCUMENT'
        then
            pplan_page gets ?page_line.
            print (#p, ?pplan_page).
            pplan_page = [].
            pplan_page gets ?blankline.
            pplan_page gets ?blankline.
            line_count = 2.
            collect ().
    end. (* page_advance *)

end. (* read_file *)

end. (* Print_Routine *)
(* PPTITLE.KB prints the Title Page for the Project Plan *)

(* no_edit_key (). *)
(* no_debug (). *)

topic 'print_title_page'.

no_edit_key ()
filler = '
.eof = number_to_char (26).
blankline = '.

expname is read(concat(C:\GARDEN\,?CURDIR,'\TITLE.DAT')).

expname is string_replace(?expname,'''',''',600).
expname is remove(?expname,'').
lines = list_length(?expname).
line_count = 1.
title_length = 0.

window (,white,red,yellow,1,16,27,4).
write ('con:','REPORT BEING PRINTED...').

while ?line_count <= ?lines
then
  cur_line = element(?expname,?line_count)
  and
  cur_length = string_length(?cur_line)
  and
  line_count = ?line_count + 1
  and
  title_length = ?title_length + ?cur_length + 1
  and
  if ?cur_line = ''
    then
      expname = string_replace(?expname,?cur_line).
      expname = concat(element(?expname,1),'~',
        element(?expname,2),'~',
        element(?expname,3),'~',
        element(?expname,4),'~',
        element(?expname,5),'~',
        element(?expname,6),'~',
        element(?expname,7),'~',
        element(?expname,8),'~',
        element(?expname,9),'~',
        element(?expname,10),'~',
        element(?expname,11),'~',
        element(?expname,12),'~',
        element(?expname,13),'~',
        element(?expname,14),'~',
        element(?expname,15)).
  title_length = ?title_length + 3.
  counter = 1.
  write_counter = 1.
  line_one = ' '.
  while ?counter < ?title_length
    then
      char = string_copy(?expname,?counter,1)
      and
      counter = ?counter + 1
and
write_counter = write_counter + 1
and
if write_counter > 40 and char = ' '
then
  linelgth = string_length(line_one)
  and
  linelgth = 80 - ?linelgth
  and
  linelgth = ?linelgth / 2
  and
  linefiller = string_copy (blankline,1,?linelgth)
  and
  line_one = concat(linefiller,line_one,'#n',char)
  and
  write_counter = 1
  and
  new_expname gets line_one
  and
  line_one = ''
else
  line_one = concat(line_one,char).

linelgth = string_length(line_one).
linelgth = 80 - ?linelgth.
linelgth = ?linelgth / 2.
linefiller = string_copy (blankline,1,?linelgth).
line_one = concat(linefiller,line_one,'#n').

new_expname gets line_one.
new_expname = string_replace(new_expname,'-',' ',600).

authorfile = concat(C:\GARDEN\,CURDIR,'\AUTHOR.DAT').
titlepage = ('#n#n#n#n#n#n#n#n#n#n#n#n#n').
titlepage gets ' Project Plan'.
titlepage gets '
'titlepage gets new_expname.
titlepage gets '#n#n'.
pagedate = ?date.
mo_num = element(pagedate,1).
if mo_num = 1
  then month = January
else
  if mo_num = 2
    then month = February
else
  if mo_num = 3
    then month = March
else
  if mo_num = 4
    then month = April
else
  if mo_num = 5
    then month = May
else
  if mo_num = 6
    then month = June
else
if ?mo_num = 7
then month = July
else
if ?mo_num = 8
then month = August
else
if ?mo_num = 9
then month = September
else
if ?mo_num = 10
then month = October
else
if ?mo_num = 11
then month = November
else
if ?mo_num = 12
then month = December
else
month = '.

year = element(?pagedate,3).
reportdate = concat(month,'',?year).

datelgth = string_length(reportdate).
datelgth = 80 - ?datelgth.
datelgth = ?datelgth / 2.
datefiller = string_copy (?blankline,1,?datelgth).
titlepage gets '.
titlepage gets '.
titlepage gets concat(datefiller,reportdate).

basefile = concat('C:\GARDEN\',?CURDIR,'\BASELINE.DAT').
baseline = read(basefile).
if ?baseline = ?eof
then
    baseline = 'N/A'.

datelgth = string_length(baseline).
datelgth = 80 - ?datelgth.
datelgth = ?datelgth / 2.
datefiller = string_copy (?blankline,1,?datelgth).
titlepage gets '.
titlepage gets '.
titlepage gets concat(datefiller,baseline).

name is read_line (?authorfile).
nname is string_replace(name,'NAME:' ,',',1).
nname is string_replace(name,' ',',',8).
namelgth = string_length(name).
namelgth = 80 - ?namelgth.
namelgth = ?namelgth / 2.
nnamefiller = string_copy (?blankline,1,?namelgth).
titlepage gets concat('',namefiller,name).

organization is read_line (?authorfile).
onorganization is string_replace(organization,'ORGANIZATION: ',',',1).
onorganization is string_replace(organization,' ',',',8).
onorganizationlgth = string_length(organization).
onorganizationlgth = 80 - ?organizationlgth.
onorganizationlgth = ?organizationlgth / 2.
onorganizationfiller = string_copy (?blankline,1,?organizationlgth).
titlepage gets concat(?organizationfiller,?organization).

mcode is read_line (?authorfile).
mcode is string_replace(?mcode,'MAIL CODE:', '',1).
mcode is string_replace(?mcode, ' ', '', 8).
mcode_length = string_length(?mcode).
mcode_length = 80 - ?mcode_length.
mcode_length = ?mcode_length / 2.
mcodefiller = string_copy (?blankline, 1, ?mcode_length).
titlepage gets concat(?mcodefiller, ?mcode).

street is read_line (?authorfile).
street is string_replace(?street, 'STREET:', '', 1).
street is string_replace(?street, ' ', '', 8).
street_length = string_length(?street).
street_length = 80 - ?street_length.
street_length = ?street_length / 2.
streetfiller = string_copy (?blankline, 1, ?street_length).
titlepage gets concat(?streetfiller, ?street).

city is read_line (?authorfile).
city is string_replace(?city, 'CITY, STATE, ZIP:', '', 1).
city is string_replace(?city, ' ', '', 8).
city_length = string_length(?city).
city_length = 80 - ?city_length.
city_length = ?city_length / 2.
cityfiller = string_copy (?blankline, 1, ?city_length).
titlepage gets concat(?cityfiller, ?city).

phone is read_line (?authorfile).
phone is string_replace(?phone, 'PHONE:', '', 1).
phone is string_replace(?phone, ' ', '', 8).
phone_length = string_length(?phone).
phone_length = 80 - ?phone_length.
phone_length = ?phone_length / 2.
phonefiller = string_copy (?blankline, 1, ?phone_length).
titlepage gets concat(?phonefiller, ?phone).

print (#p, ?titlepage, #p).
close_window ()..
end. (* print_title_page *)
This is the menu provided to the user to determine what he/she is to do on a project.

no_edit_key ().
no_debug ().
fdaction = ' '.

curdir is read_line ('CURDIR.DAT').
curdir is string_replace (?curdir,' ',' ',8).

while ?fdaction <> 'Exit System'
    then do (filldoc).

topic 'filldoc'.

fdchoices = ['Enter Project Initialization Information',
              'Complete Project Plan',
              'Return to Previous Menu',
              'Return to Main Menu',
              'Exit System'].

window (,white,red,yellow,5,5,75,16).
    set_number_of_values (fdaction,1).

    ask ([['#e #s
            Please select the activity you wish to perform on #n
             the',?curdir,'#dproject.']],fdaction,?fdchoices).

    close_window ().

if ?fdaction = 'Enter Project Initialization Information'
    then new_kb ('Pinitial.hkb').

if ?fdaction = 'Complete Project Plan'
    then new_kb ('ppquest2.hkb').

if ?fdaction = 'Return to Main Menu'
    then new_kb ('nasaplan.ckb').

if ?fdaction = 'Return to Previous Menu'
    then new_kb ('ppnasam.hkb').

if ?fdaction = 'Exit System'
    then exit ().
This program is used to allow the user to enter standard project initialization information, i.e. name, address, title, etc.

no_edit_key ().
no_debug ().
curdir is read line ('CURDIR.DAT').
curdir = string_replace(?curdir,' ',',',8).

eof = number_to_char (26).
yn is [YES,NO].
chgwant = ''. 
do (personal_info).
new_kb ('PPFILLM.hkb').
topic 'personal_info'.

blankline = '.
close_window ().
oldtext is read (concat (C:\GARDEN,?CURDIR,'\AUTHOR.DAT')).
if ?oldtext = ?eof 
then do (new_personal)
else 
  chgwant = ' '
  and 
  while ?chgwant <> QUIT 
  then do (edit_personal).

(* =========== get new personal information ===========*)

topic 'new_personal'.
WRITE ('con:','#eIn the window below, please provide some general information about yourself and your experiment.

window (,white,red,yellow,5,5,75,16).

read_response ('#e
#fyellow Please enter your first and last names, i.e. Dr. John Doe.#d 
#n ',name,?
blankline).

name = concat ('NAME: ',?name).
personal gets ?name.

read_response ('#e
#fyellow Please enter the name of your organization.#d 
#n ',organization,?
blankline).

organization = concat ('ORGANIZATION: ',?organization).
personal gets ?organization.

read_response ('#e
#fyellow Please enter the mail code, P.O Box, room number, or other needed address information of your organization.#d 
#n',mail_code,?blankline).

mail_code = concat ('MAIL CODE: ',?mail_code).
personal gets ?mail_code.
read_response ('#e
#fyellow Please enter the street address of your organization.#d
#n',street,
?blankline).

street = concat ('STREET: ',?street).
personal gets ?street.

read_response ('#e
#fyellow Please enter the city, state, and zip code of your organization.
#d
#n',city_st_zip,
?blankline).

city_st_zip = concat ('CITY, STATE, ZIP: ',?city_st_zip).
city_st_zip = string_replace (?city_st_zip,' ','/8).
personal gets city_st_zip.

read_response ('#e
#fyellow Please enter your phone number.#d
#n',phone,
?blankline).

phone = concat ('PHONE: ',?phone).
personal gets ?phone.

say ('#e

Please enter the title of your experiment.

Press the #fyellow RETURN KEY#d to enter the editor,
#fyellow ESC#d to leave the editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C:\GARDEN\,?CURDIR,'TITLE.DAT'),yellow,black,red,9,11,70

author_info is [?name,#n,?organization,#n,?mail_code,#n,?street,#n,
?city_st_zip,#n,?phone,#n].
new_file (concat (C:\GARDEN\,?CURDIR,'AUTHOR.DAT')).
write (concat (C:\GARDEN\,?CURDIR,'AUTHOR.DAT'),?author_info,#n).
close (concat (C:\GARDEN\,?CURDIR,'AUTHOR.DAT')).

if ?chgwant <> QUIT
then
ask
([?author_info, '#n #fyellow
The information listed has been written to a file.
Do you wish to change any of these entries?#d'],
change_again,?yn,60,16)
and
if ?change_again = YES
then
oldtext is read (concat (C:\GARDEN\,?CURDIR,'AUTHOR.DAT'))
and
change_again = NO
and
chgwant = '
and
while ?chgwant <> QUIT
then
    do (edit_personal).

close_window ().
WRITE ('con:','\e ').  
end. (* new_personal *)  

(* ----------------- get corrected personal information -----------------*)  

topic 'edit_personal'.  

change_info is [NAME, ORGANIZATION, 'MAIL CODE', STREET, 'CITY STATE ZIP', PHONE, TITLE, QUIT].  

ask ('Which entry in the below list do you wish to change?', chgwant, change_info).  

if ?chwant = NAME  
then  
  old_name = element(?oldtext,1)  
  and  
  old_value = string_replace (?old_name,'NAME: ','',1)  
  and  
  read_response (['\e #yellow Your original entry for name was #s ', ?old_value, '. #\n #n #n Please enter the corrected name in its entirety. #n #o'],new_name,?old_value)  

if ?chwant = NAME  
then  
  new_name = concat ('NAME: ', ?new_name)  
  and  
  oldtext is replace(?oldtext,?old_name,?new_name).  

if ?chwant = ORGANIZATION  
then  
  old_org = element(?oldtext,2)  
  and  
  old_value = string_replace (?old_org,'ORGANIZATION: ','',1)  
  and  
  read_response (['\e #yellow Your original entry for organization was #s #n ', ?old_value, ' #\n #d #s #n #n Please enter the corrected organization in its entirety. #n'],new_org,?old_value)  

if ?chwant = ORGANIZATION  
then  
  new_org = concat ('ORGANIZATION: ', ?new_org)  
  and  
  oldtext is replace(?oldtext,?old_org,?new_org).  

if ?chwant = 'MAIL CODE'  
then  
  old_mc = element(?oldtext,3)  
  and  
  old_value = string_replace (?old_mc,'MAIL CODE: ','',1)  
  and  
  read_response (['\e #yellow Your original entry for mail code was #s #n ', ?old_value, ' #\n #d #s #n #n Please enter the corrected mail code in its entirety. #n'],new_mc,?old_value).
if ?chgwant = 'MAIL CODE'
then
  new_mc = concat ('MAIL CODE: ',?new_mc)
  and
  oldtext is replace(?oldtext,?old_mc,?new_mc).

if ?chgwant = STREET
then
  old_street = element(?oldtext,4)
  and
  old_value = string_replace (?old_street,'STREET: ',',' ,1)
  and
  read_response (["#f#yellow Your original entry for street was#s #n', ?
    old_value,'.#d #s #n#n
Please enter the corrected street address in its entirety.#n'],new_street,
  ?old_value).

if ?chgwant = STREET
then
  new_street = concat ('STREET: ',?new_street)
  and
  oldtext is replace(?oldtext,?old_street•?new_street).

if ?chgwant = 'CITY STATE ZIP'
then
  old_city = element(?oldtext,5)
  and
  old_value = string_replace (?old_city,'CITY, STATE, ZIP: ',',' ,1)
  and
  read_response (["#f#yellow Your original entry for city, state and zip was;
    old_value,'.#d #s #n#n
Please enter the corrected city, state, and zip address in its entirety.#n'],
    new_city,?old_value).

if ?chgwant = 'CITY STATE ZIP'
then
  new_city = concat ('CITY, STATE•ZIP: ' ?new_city)
  and
  oldtext is replace(?oldtext,?old_city,?new_city).

if ?chgwant = PHONE
then
  old_phone = element(?oldtext,6)
  and
  old_value = string_replace (?old_phone,'PHONE: ',',' ,1)
  and
  read_response (["#f#yellow Your original entry for phone was#s #n', ?
    old_value,'.#d #s #n#n
Please enter the corrected phone number in its entirety.#n'],
    new_phone,?old_value).

if ?chgwant = PHONE
then
  new_phone = concat ('PHONE: ',?new_phone)
  and
  oldtext is replace(?oldtext,?old_phone,?new_phone).
if ?chgwant = 'TITLE'
    then
        say ('#e
            Please enter the corrected title of your experiment.
            Press the #fyellow RETURN KEY#d to enter the editor,
            #fyellow ESC#d to leave the editor, and #fyellow RETURN#d to confirm save.
            ') and
        edit_file (concat (C:\GARDEN\, ?CURDIR, '\TITLE.DAT'), yellow, black, red, 5, 9, 70,
        if ?chgwant = QUIT
            then
                new_file (concat (C:\GARDEN\, ?CURDIR, '\AUTHOR.DAT'))
                and
                write (concat (C:\GARDEN\, ?CURDIR, '\AUTHOR.DAT'), ?oldtext, #n)
                and
                close (concat (C:\GARDEN\, ?CURDIR, '\AUTHOR.DAT')).
        if ?chgwant = QUIT
            then
                say ([?oldtext, '#fyellow The information listed has been written to a file.
                    Please press #flightgreen SPACE#d #fyellow to continue.
                    '])
        end. (* edit_personal *)
    end. (* personal_info *)
(* =============== end personal information ===============*)
The Microgravity Science and Applications Division (MSAD) requires that a MSAD Project Plan be submitted and approved prior to making a major commitment of resources to an MSAD project. MSAD Project Plans are to be prepared
Plans will be prepared and submitted for all flight experiments. Project plans will be reissued, modified, or amended for reflights depending on the complexity of the task. A plan's preparation is the responsibility of the designated Project Manager at the responsible NASA center. The Project Manager will sign the MSAD Project Plan as the preparer; the Project Scientist and the Principal Investigator will sign as concurring. The MSAD Project Plan will be signed off at the NASA center prior to submission to Headquarters by the appropriate center's authorities. When the Program Scientist and Program Manager sign to register their concurrence, the MSAD Project Plan will be submitted to the MSAD Director for approval.

The Project Manager is responsible for updating a MSAD Project Plan when significant changes occur (such as changes in scope, organization, or roles and responsibilities). This does not apply to resources, schedules or manpower, which are updated through normal budgeting and project monitoring activities. The Project Manager will establish a change control process for maintaining the MSAD Project Plan and other project documentation.
topic 'Return to Previous Menu'.
new_kb ('ppfillm.hkb').
end. (* Return to Previous Menu *)
(* DISPCONT.KB *)
(* This file displays the Content Section of the Project Plan *)

column = 3.
row = 3.
no_edit_key ()
no_debug ()
do (Content).

topic 'Content'.
menu_choice2 = '
' .
menu_option2 is ['INTRODUCTION','OBJECTIVE','SCIENCE REQUIREMENTS','TECHNICAL PLAN','IMPLEMENTATION PLAN','MANAGEMENT PLAN','SCHEDULE','COST CONTROL PLAN','PROJECT REVIEWS AND MEETINGS','RETURN'].

while ?menu choice2 <> 'RETURN'
then do (outline2).

topic 'outline2'.

window ('4.0 Content',white,blue,white,?column,?row,76,18).
ask('The following outline provides guidelines on the content, organization and format of the MSAD Project Plan.',
   menu_choice2,?menu_option2).

if ?menu_choice2 = 'INTRODUCTION'
then do ('Introduction').

if ?menu_choice2 = 'OBJECTIVE'
then do ('Objective').

if ?menu_choice2 = 'SCIENCE REQUIREMENTS'
then do ('Science Requirements').

if ?menu_choice2 = 'TECHNICAL PLAN'
then new_kb ('disptech.hkb').

if ?menu_choice2 = 'IMPLEMENTATION PLAN'
then new_kb ('dispimp.hkb').

if ?menu_choice2 = 'MANAGEMENT PLAN'
then new_kb ('dispmgt.hkb').

if ?menu_choice2 = 'SCHEDULE'
then do ('Schedule').

if ?menu_choice2 = 'COST CONTROL PLAN'
then new_kb ('dispcost.hkb').

if ?menu_choice2 = 'PROJECT REVIEWS AND MEETINGS'
then new_kb ('disprrev.hkb').

if ?menu_choice2 = 'RETURN'
then new_kb ('ppout.hkb').
Describe the relevance of the investigation and provide a summary rationale as to why a flight experiment is required (limit to one page).

Press #fyellow SPACE#d to continue’

Define the overall objectives of the flight experiment(s); if more than one flight is requested, indicate the specific objectives of each flight.

Press #fyellow SPACE#d to continue’

Summarize the science requirements against which the hardware will be built and reference the applicable Science Requirements Document(s).

Press #fyellow SPACE#d to continue’}

close_window ().
column = ?column - 1.
row = ?row - 1.
end. (* Science Requirements *)

topic 'Schedule'.
column = ?column + 1.
row = ?row + 1.
window ('4.7 Schedule',blue,white,white,?column,?row,72,17).
say ('Provide an overall project master schedule that includes the flow of hardware and software into the system integration and test activity. Identify the Headquarters and mission management center controlled milestones. To the extent possible, tie the schedule to the WBS and overlay the procurement and NASA budget cycles. Provide a narrative describing the schedule and overall project flow.

Press #fyellow SPACE#d to continue’}

close_window ().
column = ?column - 1.
row = ?row - 1.
end. (* Schedule *)
column = 3.
row = 3.

no_edit_key ().
no_debug ().

do ('Technical Plan').

topic 'Technical Plan'.
menu_choice3 = ‘ ’.
menu_option3 is ['EXPERIMENT HARDWARE DESCRIPTION', 'PAYLOAD CLASSIFICATION',
'DEVELOPMENT APPROACH', 'TECHNOLOGY PLAN', 'LOGISTICS',
'MISSION OPERATIONS, TRAINING AND DATA MANAGEMENT',
'ANALYSIS OF MISSION RESULTS', 'FACILITIES', 'SAFETY',
'RETURN'].

While ?menu_choice3 <> 'RETURN'
then do (outline3).

topic 'outline3'.
window ('4.4 Technical Plan',,,,?column,?row,77,18).
ask ('#e
The outline for the Technical Plan includes the
following sections:’,menu_choice3,?menu_option3).

if ?menu_choice3 = 'EXPERIMENT HARDWARE DESCRIPTION'
then do ('Experiment Hardware Description').

if ?menu_choice3 = 'PAYLOAD CLASSIFICATION'
then do ('Payload Classification').

if ?menu_choice3 = 'DEVELOPMENT APPROACH'
then do ('Development Approach').

if ?menu_choice3 = 'TECHNOLOGY PLAN'
then do ('Technology Plan').

if ?menu_choice3 = 'LOGISTICS'
then do ('Logistics').

if ?menu_choice3 = 'MISSION OPERATIONS, TRAINING AND DATA MANAGEMENT'
then
do ('Mission Operations, Training and Data Management').

if ?menu_choice3 = 'ANALYSIS OF MISSION RESULTS'
then do ('Analysis of Mission Results').

if ?menu_choice3 = 'FACILITIES'
then do ('Facilities').

if ?menu_choice3 = 'SAFETY'
then do ('Safety').

if ?menu_choice3 = 'RETURN'
then new_kb ('dispcont.hkb').
Provide an overall description of the experiment hardware and relate the hardware to the science requirements shown in section 4.3. Relate differing hardware configurations or upgrades to the relevant objectives shown in section 4.2.

Press #fyellow SPACE#d to continue.

State and show the rationale for the payload classification.

Press #fyellow SPACE#d to continue.

Say ("
Describe the overall development approach, indicating plans for breadboard, engineering model, and/or proto-flight hardware development. Identify the numbers of flight units and test articles, and define the fidelity of simulators required. Identify any support hardware required. Define the spares philosophy and the quantity of spares required by the approach to be used.

Press #fyellow SPACE#d to continue’).

close window ().
column = ?column - 1.
row = ?row - 1.
end. (* Development Approach *)

topic 'Technology Plan'.
column = ?column + 1.
row = ?row + 1.
window ('4.4.4 Technology Plan',blue,white,white,?column,?row,72,15).
say ("Indicate what feasibility issues or proof-of-concept issues have been identified via ground-based testing during the definition phase of the project. These should have been identified at the Conceptual Design Review (CoDR) and closed by the RDR. It is not necessary to repeat the discussion of the breadboard program defined in section 4.4.3. If there was no technology development requirement, this fact should be noted.

Press #fyellow SPACE#d to continue’}).

close window ().
column = ?column - 1.
row = ?row - 1.
end. (* Technology Plan *)

topic 'Logistics'.
column = ?column + 1.
row = ?row + 1.
window ('4.4.5 Logistics',blue,white,white,?column,?row,72,15).
say ("Identify where the major project functions, such as hardware build, different levels of integration, etc., will take place and describe the special services, vehicles, systems, and major equipment necessary to satisfy the logistic requirements of the project.

Press #fyellow SPACE#d to continue’}).
Describe the operations approach, starting with a summary of the experiment operations sequence, and relate the crew involvement with the operations. Identify the location of the integration and operations activities and the organization supplying the support and define the level of support required. Identify where and how operations training will be performed and how data will be made available to the principal investigator for analysis. All assumptions should be clearly stated.

Press #yellow SPACE#d to continue’).

Describe procedures, associated efforts, and primary locations for the postmission analysis of data and other mission results. Define experiment records to be developed and how they will be archived.

Press #yellow SPACE#d to continue’).

Describe and outline major in-house and contractor facilities (existing, modified and new) for fabrication,
test, checkout, launch, flight and mission operations, and data acquisition and analysis.

Press #fyellow SPACE#d to continue']).

close_window ().
Column = ?column - 1.
row = ?row - 1.
end. (* Facilities *)

topic 'Safety'.
column = ?column + 1.
row = ?row + 1.
window ('4.4.9 Safety',blue,white,white,?column,?row,72,15).
say ('

Define the ground and mission safety requirements for the project.

Press #fyellow SPACE#d to continue').

close_window ().
column = ?column - 1.
row = ?row - 1.
end. (* Safety *)
column = 3.
row = 3.
no_edit_key ().
no_debug ().
menu_choice = ' '
menu_option is ['IMPLEMENTATION APPROACH','SUMMARY WORK BREAKDOWN STRUCTURE','DOCUMENTATION','RETURN']

while ?menu_choice <> 'RETURN'
  then do ('Implementation Plan').

topic 'Implementation Plan'.
window ('4.5 Implementation Plan',white,red,yellow,?column,?row,76,17).
ask ('Implementation Plan includes these three sections:',menu_choice,
  ?menu_option).

if ?menu_choice = 'IMPLEMENTATION APPROACH'
  then do ('Implementation Approach').

if ?menu_choice = 'SUMMARY WORK BREAKDOWN STRUCTURE'
  then do ('Summary Work Breakdown Structure').

if ?menu_choice = 'DOCUMENTATION'
  then do ('Documentation').

if ?menu_choice = 'RETURN'
  then new_kb ('dispcont.hkb').

close window ().
end. (* Implementation Plan *)

topic 'Implementation Approach'.
column = ?column + 1.
row = ?row + 1.
window ('4.5.1 Implementation Approach',blue,white,white,?column,?row,71,15).
say ('Indicate whether the project is an in-house activity
or a contracted activity. Define the principal entities on
the project team, including the contractor team and its re-
sponsibilities. If the contractor team has not yet been se-
lected, include a description of the contractor procurement
approach and schedule.

Press #fyellow SPACE#d to continue').

close_window ().
column = ?column - 1.
row = ?row - 1.
end. (* Implementation Approach * )
Provide a summary Work Breakdown Structure (WBS) down to the "appropriate level" (as agreed to between the Project Manager, and the Program Manager) and a brief description of each WBS element. The WBS structure should be related to line items on the project master schedule and to the resources section, and should serve as the baseline WBS to be used in future budget reporting.

Press yellow SPACE d to continue’}).

Provide a project documentation tree that includes all governing, hardware development, mission integration, and safety documentation.

Press yellow SPACE d to continue’}).
(* DISPMGT.KB *)
(* THIS FILES DISPLAYS THE MANAGEMENT PLAN SECTION OF THE PROJECT PLAN. *)

column = 3.
row = 3.
no_edit_key ()
no_debug ()
do ('Management Plan').
topic 'Management Plan'.

menu_choice = ' ':
menu_option is ['PROJECT MANAGEMENT RESPONSIBILITIES AND ORGANIZATION', 'MISSION MANAGEMENT RESPONSIBILITIES AND ORGANIZATION','RETURN'].

while ?menu_choice <> 'RETURN'
then do (Outline).
topic 'Outline'.

window ('4.6 Management Plan',white,red,yellow,?column,?row,76,17).
as ('Management Plan includes these two sections: ',menu_choice, ?menu_option).

if ?menu_choice = 'PROJECT MANAGEMENT RESPONSIBILITIES AND ORGANIZATION'
then do ('Project Management Responsibilities and Organization').

if ?menu_choice = 'MISSION MANAGEMENT RESPONSIBILITIES AND ORGANIZATION'
then do ('Mission Management Responsibilities and Organization').

if ?menu_choice = 'RETURN'
then new_kb ('dispcont.hkb').
close window ()
end. (* outline *)
end. (* Management Plan *)

topic 'Project Management Responsibilities and Organization'.
column = ?column + 1.
row = ?row + 1.

window ('4.6.1 Project Management Responsibilities and Organization',blue, white,white,?column,?row,71,15).
say ('Provide an organization chart of the project, including the Center Director, Project Manager, Principal Investigator, Project Scientist, and other key positions on the project, and all external project interfaces. Relate the organization chart to the WBS (section 5.2).'

Press #yellow SPACE#d to continue').
close_window ()
column = ?column - 1.
Identify the NASA center responsible for mission management and describe what process is to be used to define the management and technical interface agreement between the NASA center responsible for the project and the Mission Management Center. Indicate the mission management responsibilities and interfaces on the organization chart prepared for section 6.1. Identify all assumptions that will affect project plans, schedules and costs.

Press #fyellow SPACE#d to continue’).
cost_control_plan

Cost Control Plan

menu_choice = ' '.
menu_options is ['RESOURCES', 'COST CONTROL GUIDELINES',
'COST REPORTING AND CONTROL STRUCTURE',
'COST CONTROL STRATEGY', 'RETURN'].

while ?menu_choice <> 'RETURN'
topic 'Outline'.

while ?menu_choice <> 'RETURN'
topic 'Resources'.

Specify the funding (including contingency) and manpower requirements needed for the life of the project against the WBS. These resource requirements should be consistent with results from the Independent Cost Review held in conjunction with the RDR. Identify any assumptions made in determining the cost. Select the appropriate WBS level to be the Cost Reporting level. Identify
any use to be made of other facilities for which another entity (other than the project) will be financially responsible. (This should be consistent with facility usage described in section 4.8.)

Press #fyellow SPACE#d to continue’}).

close_window ().
column = ?column - 1.
row = ?row - 1.
end. (* Resources *)

topic ‘Cost Control Guidelines’.
column = ?column + 1.
row = ?row + 1.
window ('4.8.2 Cost Control Guidelines',blue,white,white,?column,?row, 71,15).
say ([‘
Define the cost control guidelines to be used in coping with cost variations at the different project levels.

The following are examples of guidelines that may be used and are not intended to be universally applied. Each project can develop its own guidelines according to the individual project needs.

a. MSAD Allowance for Program Adjustment (APA) is to be utilized for changes in program scope imposed by external circumstances, such as change in launch date or launch vehicle and internal changes in the science requirements.

b. Project Office Contingency is to be utilized to fix internal problems for the Pia’s or hardware contractor, such as design changes, parts procurement, alternate approaches, and change of subcontractors.

c. Contingency reserves at the lower WBS levels (if they exist) will be used to solve problems in those WBS elements. Only when problems cannot be solved within the WBS element are they to be referred to the next higher cost control point.

Press #fyellow SPACE#d to continue’}).

close_window ().
column = ?column - 1.
row = ?row - 1.
end. (* Cost Control Guidelines *)

topic ‘Cost Control Strategy’.
column = ?column + 1.
row = ?row + 1.
window ('4.8.4 Cost Control Strategy',blue,white,white,?column,?row,71,
This section should contain a discussion of courses of the event of cost growths. The entire hardware development process should be considered, including the test plans, support equipment, and facility requirements, etc., as well as the mainline flight hardware development.

This discussion should include the prioritization of science requirements science and hardware development descope options, and appropriate decision points. Program stretch-out options should also be addressed if appropriate. Obviously, this section should be developed with a high degree of participation by the Principle investigator and Project and Program Scientists. Overall discussions on cost management, such as how to deal with overruns and underruns at the lower WBS and project levels, should also be included.

Press #fyellow SPACE#d to continue').

close_window ().
column = ?column - 1.
row = ?row - 1.
end. (* Cost Control Strategy *)
(* COSTREP.KB *)
(* THIS FILE DISPLAYS THE COST REPORTING AND CONTROL STRUCTURE SECTION *)
(* OF THE PROJECT PLAN *)

column = 3.
row = 3.
no_edit_key ().
no_debug ().
do ('Cost Reporting and Control Structure').

topic 'Cost Reporting and Control Structure'.
menu_choice = '.
menu_option is ['NASA REPORTS', 'CONTRACTOR REPORTS', 'RETURN'].

while ?menu_choice <> 'RETURN'
then do (outline).

topic 'outline'.
window ('4.8.3 Cost Reporting and Control Structure', white, red, yellow, ?column, ?row, 76, 17).
ask ('Cost Reporting and Control Structure includes these two sections: ','
menu_choice, ?menu_option).

if ?menu_choice = 'NASA REPORTS'
then do ('Nasa Reports').

if ?menu_choice = 'CONTRACTOR REPORTS'
then do ('Contractor Reports').

if ?menu_choice = 'RETURN'
then new_kb ('dispcost.hkb').
close_window ().
end. (* Outline *)
end. (* Cost Reporting and Control Structure *)

topic 'NASA Reports'.
column = ?column + 1.
row = ?row + 1.
window ('4.8.3.1 NASA Reports', blue, white, white, ?column, ?row, 72, 15).
say (['Show the #mWBS#m structure to be used for cost reporting and control, and identify the monthly Management Information Control System (MIC5) (or equivalent) and Project Operating Plan (POP) reporting format to be used. These should result from negotiations between the Project and Program Managers and will vary in detail depending on project size and complexity. Press #fyellow SPACE#d to continue']).
close_window ().
column = ?column - 1.
row = ?row - 1.
end. (* NASA Reports *)
topic 'Contractor Reports'.
column = ?column + 1.
row = ?row + 1.
window ('4.8.3.2 Contractor Reports',blue,white,white,?column,?row,72,15).
say('Where applicable, delineate the contractor reports to be provided to the NASA center.

Press #fyellow SPACE#d to continue').

close window ()..
column = ?column - 1.
row = ?row - 1.
end. (* Contractor Reports *)

topic 'WBS'.
column = ?column + 1.
row = ?row + 1.
window ('WBS',blue,white,white,?row,?column,72,13).
say('#e Work Breakdown Structure

Press #fyellow SPACE#d to continue.').

close window ()..
column = ?column - 1.
row = ?row -1.
end. (* WBS *)
Identify weekly or monthly project meetings and reviews with the NASA center's management as appropriate.
Press #fyellow SPACE#d to continue').

close window ().
column = ?column - 1.
row = ?row - 1.
end. (* Internal Reviews *)

topic 'External Reviews'.
column = ?column + 1.
row = ?row + 1.
window ('4.9.1.2 External Reviews',blue,white,white,?column,?row,72,16).
say ('Identify biweekly telecons, monthly reviews, or periodic program reviews with NASA Headquarters, Mission Management, etc., as appropriate.

Press #fyellow SPACE#d to continue').

close window ().
column = ?column - 1.
row = ?row - 1.
end. (* External Reviews *)

topic 'Design and Readiness Reviews'.
window ('4.9.1.3 Design and Readiness Reviews',blue,white,white,?column,?row,72,17).
say ('Make reference to the major project reviews, including those required by the MSAD management plan as well as integrated payload reviews, safety reviews, and flight readiness reviews.

Press #fyellow SPACE#d to continue').

close window ()..
end. (* Design and Readiness Reviews *)
(* Content.kb *)
(* This is the program for filling out the Content section of the Project *)
(* Plan. *)
column = 3.
row = 3.
no_edit_key ().
no_debug ().
curdir is read_line ('CURDIR.DAT').
curdir = string_replace(?curdir,' ','',8).

menu_choice = ' '.

menu_option is ['Introduction','Objective','Science Requirements','Technical Plan','Implementation Plan','Management Plan','Schedule','Cost Control Plan','Project Reviews and Meetings','Return to Previous Menu'].

while ?menu_choice <> 'Return to Previous menu'
    then do (Coutline).

    topic 'Coutline'.
    window ('4.0 Content',white,red,yellow,?column,?row,76,17).
    ask ('#e
        Which section would you like to complete?',menu_choice,?menu_option).

    if ?menu_choice = 'Introduction'
        then do ('Introduction').

    if ?menu_choice = 'Objective'
        then do ('Objective').

    if ?menu_choice = 'Science Requirements'
        then do ('Science Requirements').

    if ?menu_choice = 'Technical Plan'
        then do ('Technical Plan').

    if ?menu_choice = 'Implementation Plan'
        then do ('Implementation Plan').

    if ?menu_choice = 'Management Plan'
        then do ('Management Plan').

    if ?menu_choice = 'Schedule'
        then do ('Schedule').

    if ?menu_choice = 'Cost Control Plan'
        then do ('Cost Control Plan').

    if ?menu_choice = 'Project Reviews and Meetings'
        then do ('Project Reviews and Meetings').

    if ?menu_choice = 'Return to Previous Menu'
        then new_kb ('ppquest2.hkb').
close_window ().
end. (* Content *)
new_kb ('ppquest2.hkb').
Please describe the relevance of the investigation and provide a summary rationale as to why a flight experiment is required (limit to one printed page).

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.'

Please define the overall objective(s) of the flight experiment(s); if more than one flight is requested, indicate the specific objectives of each flight.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.'

...
Please summarize the science requirements against which the hardware will be built and reference the applicable Science Requirements Document(s).

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C:\GARDEN\,?CURDIR,\'PPE4_3.DAT\'),yellow,black,red,6,12,70,)
close_window ()
close (concat (C:\GARDEN\,?CURDIR,\'PPE4_3.DAT\'))
column = ?column - 1.
row = ?row - 1.
end. (* Science Requirements *)

Please indicate whether the project is an in-house activity or a contracted activity. Define the principle entities on the project team, including the contractor team and its responsibilities. If the contractor team has not yet been selected, include a description of the contractor procurement approach and schedule.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C:\GARDEN\,?CURDIR,\'PPE4_5_1.DAT\'),yellow,black,red,6,12,70

close_window ()
close (concat (C:\GARDEN\,?CURDIR,\'PPE4_5_1.DAT\'))
column = ?column - 1.
row = ?row - 1.
end. (* Implementation Approach *)

topic 'Summary Work Breakdown Structure'.
column = ?column + 1.
row = ?row + 1.
window ('4.5.2 Summary Work Breakdown Structure',blue,white,white,?
column,?row,72,14).
say ("#e
Please provide a summary Work Breakdown Structure (WBS) down to the "appropriate level" (as agreed to between the Project Manager, and the Program Manager) and a brief description of each WBS element. The WBS structure should be related to line items on the project master schedule and to the resources section, and it should serve as the baseline WBS to be used in future budget reporting.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

```plaintext
edit_file (concat (C:\GARDEN, ?CURDIR,'\PPE4_5_2.DAT'), yellow, black, red, 6, 12, 70, close_window ().
```

Please provide a project documentation tree that includes all governing, hardware development, mission integration, and safety documentation. If a narrative should accompany this chart, please provide it here.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

```plaintext
documentation (concat (C:\GARDEN, ?CURDIR,'\PPE4_5_3.DAT'), yellow, black, red, 6, 12, 70, close_window ().
```

Please provide an overall project master schedule that includes the flow of hardware and software into the system integration and test activity. Identify the Headquarters and mission management center controlled milestones. To the extent possible, tie the schedule to the WBS and overlay the procurement and NASA budget.
cycles. Provide a narrative describing the schedule and overall project flow.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

```
edit_file (concat (C:\GARDEN\, ?CURDIR, '\PPE4_7.DAT'), yellow, black, red, 6, 12, 70, close_window ()).
close (concat (C:\GARDEN\, ?CURDIR, '\PPE4_7.DAT')).
column = ?column - 1.
row = ?row - 1.
end. (* Schedule *)

topic 'Cost Control Plan'.
new_kb ('costcont.hkb').
end. (* Cost Control Plan *)

topic 'Project Reviews and Meetings'.
new_kb ('projrev.hkb').
end. (* Project Reviews and Meetings *)

topic 'WBS'.
column = ?column + 1.
row = ?row + 1.
window ('WBS', blue, white, white, ?row, ?column, 72, 14).
say( '#e

Work Breakdown Structure

Press #fyellow SPACE#d to continue.' ).
```

close_window ()..
column = ?column - 1.
row = ?row - 1.
end. (* WBS *)
(* TECHPLAN.KB *)
(* This is the program for completing the Technical Plan Section of the *)
(* Project Plan *)
column = 2.
row = 2.
no_edit_key ().
nod debug ().
curdir is read line ('CURDIR.DAT').
curdir = string replace (?curdir, ', ', '', 8).

menu_choice = ' '

menu_option is ['EXPERIMENT HARDWARE DESCRIPTION', 'PAYLOAD CLASSIFICATION',
'DEVELOPMENT APPROACH', 'TECHNOLOGY PLAN', 'LOGISTICS',
'MISSION OPERATIONS, TRAINING AND DATA MANAGEMENT',
'ANALYSIS OF MISSION RESULTS', 'FACILITIES', 'SAFETY',
'RETURN TO PREVIOUS MENU'].

While ?menu_choice <> 'RETURN TO PREVIOUS MENU'
then do (Toutline).

topic 'Toutline'.
column = ?column + 1.
row = ?row + 1.
window ('4.4 Technical Plan',,,,?column,?row,77,17).
ask ('The outline for the Technical Plan includes the following sections:',
?menu_choice,?menu_option).

if ?menu_choice = 'EXPERIMENT HARDWARE DESCRIPTION'
then do ('Experiment Hardware Description').

if ?menu_choice = 'PAYLOAD CLASSIFICATION'
then do ('Payload Classification').

if ?menu_choice = 'DEVELOPMENT APPROACH'
then do ('Development Approach').

if ?menu_choice = 'TECHNOLOGY PLAN'
then do ('Technology Plan').

if ?menu_choice = 'LOGISTICS'
then do ('Logistics').

if ?menu_choice = 'MISSION OPERATIONS, TRAINING AND DATA MANAGEMENT'
then do ('Mission Operations, Training and Data Management').

if ?menu_choice = 'ANALYSIS OF MISSION RESULTS'
then do ('Analysis of Mission Results').

if ?menu_choice = 'FACILITIES'
then do ('Facilities').

if ?menu_choice = 'SAFETY'
then do ('Safety').

if ?menu_choice = 'RETURN TO PREVIOUS MENU'
then new_kb ('content.hkb').
end. (* Technical Plan *)
new_kb ('content.hkb').

******************************* RELATED TOPICS ****************************

topic 'Experiment Hardware Description'.
relatedfile1 = '\PPE4_3.dat'.
relatedfile2 = '\PPE4_2.dat'.
column = ?column + 1.
row = ?row + 1.
window ('4.4.1 Experiment Hardware Description',blue,white,white,?column,
?row,72,15).
say ('#e

Please provide an overall description of the experiment hardware and relate the hardware to the science requirements shown in #mSection 4.3#m. Relate differing hardware configurations or upgrades to the relevant objectives shown in #mSection 4.2#m.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.’).

edit_file (concat (C:\GARDEN,?CURDIR, '\PPE4_4_1.DAT'),yellow,black,red,6,12,7
close_window ()).

end. (* Experiment Hardware Description *)

topic 'Payload Classification'.
new_kb ('payclass.hkb').
end. (* payload classification *)

topic 'Development Approach'.
column = ?column + 1.
row = ?row + 1.
window ('4.4.3 Development Approach',blue,white,white,?column,?row,72,14).
say ('#e

Please describe the overall development approach, indicating plans for breadboard, engineering model, and/or protolflight hardware development. Identify the numbers of flight units and test articles, and define the fidelity of simulators required. Identify any support hardware required. Define the spares philosophy and the quantity of spares required by the project. Use a #mdetailed flow diagram#m to clarify the development and test approach to be used.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.’).

edit_file (concat (C:\GARDEN,?CURDIR, '\PPE4_4_3.DAT'),yellow,black,red,6,12,7C

close_window ().
Please indicate what feasibility issues or proof-of-concept issues have been identified via ground-based testing during the definition phase of the project. These should have been identified at the Conceptual Design Review (CDR) and closed by the #mRDR#m. It is not necessary to repeat the discussion of the breadboard program defined in #mSection 4.4.3#m. If there was no technology development requirement, this fact should be noted.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.’}.

Please identify where the major project functions, such as hardware build, different levels of integration, etc., will take place and describe the special services, vehicles, systems, and major equipment necessary to satisfy the logistic requirements of the project.

Press #fyellow RETURN KEY#d to enter editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.’

Press #fyellow RETURN KEY#d to enter editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.’}.

Please identify where the major project functions, such as hardware build, different levels of integration, etc., will take place and describe the special services, vehicles, systems, and major equipment necessary to satisfy the logistic requirements of the project.

Press #fyellow RETURN KEY#d to enter editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.’

Press #fyellow RETURN KEY#d to enter editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.’

Press #fyellow RETURN KEY#d to enter editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.’

Please identify where the major project functions, such as hardware build, different levels of integration, etc., will take place and describe the special services, vehicles, systems, and major equipment necessary to satisfy the logistic requirements of the project.

Press #fyellow RETURN KEY#d to enter editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.’
Please describe the operations approach, starting with a summary of the experiment operations sequence, and relate the crew involvement with the operations. Identify the location of the integration and operations activities and the organization supplying the support and define the level of support required. Identify where and how operations training will be performed and how data will be made available to the principal investigator for analysis. All assumptions should be clearly stated.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.'

Please describe procedures, associated efforts, and primary locations for the postmission analysis of data and other mission results. Define experiment records to be developed and how they will be archived.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.'

Please describe and outline major in-house and contractor facilities (existing, modified and new) for fabrication,
test, checkout, launch, flight and mission operations, and
data acquisition and analysis.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d
to leave editor, and #fyellow RETURN#d to confirm save.'

edit_file (concat (C:\GARDEN\,?CURDIR,\'\PPE4_4_8.DAT' ),yellow,black,red,6,12,70,
close_window ()).
close (concat (C:\GARDEN\,?CURDIR,\'\PPE4_4_8.DAT' )).
Column = ?column - 1.
row = ?row - 1.
end. (* Facilities *)

topic 'Safety'.
column = ?column + 1.
row = ?row + 1.
window ('4.4.9 Safety',blue,white,white,?column,?row,72,14).
say ('Please define the ground and mission safety requirements
for the project.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d
to leave editor, and #fyellow RETURN#d to confirm save.'

edit_file (concat (C:\GARDEN\,?CURDIR,\'\PPE4_4_9.DAT' ),yellow,black,red,6,12,70,
close_window ()).
close (concat (C:\GARDEN\,?CURDIR,\'\PPE4_4_9.DAT' )).
column = ?column - 1.
row = ?row - 1.
end. (* Safety *)

topic 'detailed flow diagram'.
collect ()
and
dos ('SCHEDULE',restore).
end. (* detailed flow diagram *)

topic 'Section 4.3'.
column = ?column + 1.
row = ?row + 1.
related_answer is read(concat(C:\GARDEN\,?curdir,?relatedfile1)).
window ('4.3 Science Requirements',blue,white,white,?column,?row,73,15).
say ('This is your answer for Section 4.3: #t #n,'
related_answer, '#n #n #n #n Press #fyellow SPACE#d to continue.').
column = ?column - 1.
row = ?row - 1.
close_window ().
end. (* Section 4.3 *)
topic 'Section 4.2'.
column = ?column + 1.
row = ?row + 1.
related_answer is read(concat(C:\GARDEN\,?curdir,?relatedfile2)).
window ('4.2 Objectives',blue,white,white,white,?column,?row,73,15).
say ('This is your answer for Section 4.2: #t #n',
      related_answer, '#n #n #n #n
      Press #fyellow SPACE#d to continue.').
column = ?column - 1.
row = ?row - 1.
close_window ().
end. (* Section 4.2 *)

topic 'Section 4.4.3'.
column = ?column + 1.
row = ?row + 1.
related_answer is read(concat(C:\GARDEN\,?curdir,?relatedfile3)).
window ('4.4.3 Development Approach',blue,white,white,white,?column,?row,73,14).
say ('This is your answer for Section 4.4.3: #t #n',
      related_answer, '#n #n #n #n
      Press #fyellow SPACE#d to continue.').
column = ?column - 1.
row = ?row - 1.
close_window ().
end. (* Section 4.4.3 *)

topic 'RDR'.
column = ?column + 1.
row = ?row + 1.
window ('RDR',blue,white,white,?row,?column,72,14).
say('e

Requirements Definition Review

Press #fyellow SPACE#d to continue.').
close_window ().
column = ?column - 1.
row = ?row - 1.
end. (* RDR *)
(* PAYCLASS.KB *)
(* THIS PROGRAM IS FOR COMPLETING THE PAYLOAD CLASSIFICATION SECTION *)
(* OF THE PROJECT PLAN *)

column = 2.
row = 2.
no_edit_key ()
no_debug ()
curdir is read_line ('CURDIR.DAT').
curdir = string_replace (?curdir, ',','/',8).
do (PCoutlne).

topic 'PCoutlne'.
column = ?column + 1.
row = ?row + 1.
window ('4.4.2 Payload Classification',blue,white,white,?column,?row,72,17).
say ('Please state and show the rationale for the payload classification as determined by #mNasa Management Instruction 8010.1A#m. The payloads are classified as either class A, B, C, or D. The guidelines for determining payload classification are as follows:

#mClass A#m
#mClass B#m
#mClass C#m
#mClass D#m

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.').

edit_file (concat (C:\GARDEN\,?CURDIR,\'\PPE4_4_2.DAT'),yellow,black,red,6,12,70
close_window ()
close (concat (C:\GARDEN\,?CURDIR,\'\PPE4_4_2.DAT')).
column = ?column - 1.
row = ?row - 1.
new_kb ('techplan.hkb').

del. (* Payload Classification *)

topic 'Class A'.
column = ?column + 1.
row = ?row + 1.
window ('Class "A" Payload Characterization',blue,white,white,?column,?row,72,17).
say ('

==== CHARACTERIZATION, MISSION SUCCESS, AND SRM&QA COST GUIDELINES =====
FOR CLASS A PAYLOADS

CLASSIFICATION: CLASS A

Characterization: High Priority, Minimum Risk

Typical Factors Used to Determine Payload

High national prestige;
Long hardware life required;
Classifications: High complexity; Highest cost;

Long program duration;
Critical launch constraints;
Retrieval/reflight or in-flight maintenance to recover from problems is not feasible.

Achievement of Mission Success Criteria:
All affordable programmatic and other measures are taken to achieve minimum risk. The highest practical product assurance standards are utilized.

Estimated Relative SRM&QA Cost Factors: 1.0

GUIDELINES FOR SRM&QA PROGRAM REQUIREMENTS FOR PAYLOADS

SRM&QA ELEMENT:

Engineering Model, Prototype, Flight And Spare Hardware

- Engineering Model Hardware For New Or Modified Designs.
- Separate Prototype And Flight Modes Hardware. Full Set Of Assembled And Tested "Flight Spare" Replacement Units.

Failure Investigation Board Requirements

- Formal Board Required - Initiated And Conducted by Headquarters.

Treatement of Single Failure Points (SFPs)

- Success Critical SFPs Are Not Permitted Except by Formal Project Waiver. Retention of Unavoidable SFPs Requires Justification Based On Risk Analysis And Implementation Of Measures To Mitigate Risk.

Qualification, Acceptance, And Protoflight Test Program

- Full Formal Qualification And Acceptance Test Programs At All Hardware Levels. Extensive Design Margin And Development Testing For New or Modified Designs.

EEE Parts

- Grade I (per Mil-Std-975).

Reviews


Safety

- Per All Applicable NASA Safety Standards.

Materials

- Verify Heritage of Previously

Reliability

Failure Mode And Effects Analysis/Critical Items List (FEMA/CIL), Worst Case Performance And Parts Stress Analyses Required For All Parts And Circuits. Mechanical Reliability And Other Reliability Analyses Required Where Appropriate

Maintainability

Formal Maintainability Program For All Appropriate Flight And Critical Ground Support Elements.

Quality Assurance

Formal Quality Assurance Program Including Closed-Loop Problem Reporting And Analysis of SRM&QA And Performance Trends.

Software #m*2#m


Press #fyellow SPACE#d to continue').

topic '*1'.
Window (' Note *1 ',yellow,red,white,6,6,68,10,).
say ('
There are wide variations in the methods for specifying and accounting for "SRM&QA costs". For Class A Programs, these costs are typically in the range of 10 t 15% of the total program cost. The relative SRM&QA cost factors specified here are intended to require substantiative differences in the SRM&QA programs (and the associated costs) for the various program classifications in order to establish a meaningful ladder of cost/risk levels.

Press #fwhite SPACE#d to continue').'.

close_window ().
end. (* *1 *)

topic '*2'.
Window (' Note *2 ',yellow,red,white,6,6,68,7).
say ('#e
Outside/Independent Review refers to reviews conducted by personnel who are not managing or directly associated with the program or the design effort (i.e., personnel who are not assigned to or in the management chain for the program and who are not performing or managing the design effort).

Press #fwhite SPACE#d to continue').'.


close_window ().
end. (* *2 *)

close_window ().
column = ?column - 1.
row = ?row - 1.
end. (* Class A *)
topic 'Class B'.
column = ?column + 1.
row = ?row + 1.
window ('Class "B" Payload Characterization',blue,white,white,?column,?row,72,17).
say ('#e

==== CHARACTERIZATION, MISSION SUCCESS, AND SRM&QA COST GUIDELINES ======
FOR CLASS B PAYLOADS

<table>
<thead>
<tr>
<th>CLASSIFICATION:</th>
<th>CLASS B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characterization:</td>
<td>High Priority, Medium Risk</td>
</tr>
</tbody>
</table>

Typical Factors Used to Determine Payload Classifications:
- High national prestige;
- Medium hardware life required;
- High to medium complexity;
- High cost;
- Medium program duration;
- Some launch constraints;
- Retrieval/reflight or in-flight maintenance to recover from problems is difficult or not feasible.

Achievement of Mission Success Criteria:
- Compromises are used to permit somewhat reduced costs while maintaining low risk to the overall mission success and a medium risk of achieving only partial success. Stringent product assurance standards are utilized.

Estimated Relative SRM&QA Cost Factors $0.7 \times \text{Class A}$

==== GUIDELINES FOR SRM&QA PROGRAM REQUIREMENTS FOR PAYLOADS ========

SRM&QA ELEMENT:

|--------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|

Failure Investigation Board Requirements
- Formal Board Required - Initiated And Conducted by Headquarters.
<table>
<thead>
<tr>
<th>Topic</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treatment of Single Failure Points (SFPs)</td>
<td>Success Critical SFPs Are Allowed W/O Formal Waiver But Are Minimized And Mitigated By Use Of High Reliability Parts And Additional Testing. Essential Spacecraft Functions And Key Instruments Are Typically Fully Redundant. Other Hardware Has Partial Redundancy And/Or Provisions For Graceful Degredation.</td>
</tr>
<tr>
<td>EEE Parts</td>
<td>Combination Grade I and Grade II (per Mil-Std-975).</td>
</tr>
<tr>
<td>Reviews #m*2#m</td>
<td>Formal Review Program; may Include Outside/Independent Reviews.</td>
</tr>
<tr>
<td>Safety</td>
<td>Per All Applicable NASA Safety Standards.</td>
</tr>
<tr>
<td>Reliability</td>
<td>FMEA/CIL Required At A Black Box (Or Circuit Block Diagram) Level. Worst Case Performance And Parts Stress Analysis Required For All Parts And Circuits.</td>
</tr>
<tr>
<td>Maintainability</td>
<td>Formal Maintainability Program For All Appropriate Flight And Critical Ground Support Elements.</td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>Formal Quality Assurance Program Including Closed-Loop Problem Reporting And Analysis Of SRM&amp;QA And Performance Trends.</td>
</tr>
<tr>
<td>Software #m*2#m</td>
<td>Formal Software Quality Assurance Program Including A Formal Review Program With Outside/Independent Reviews and Independent Verification And Validation.</td>
</tr>
</tbody>
</table>

Press #fyellow SPACE#d to continue.

*Note*: This text is a continuation of a larger document and may require further context to fully understand its implications.
There are wide variations in the methods for specifying and accounting for "SRM&QA costs". For Class A Programs, these costs are typically in the range of 10 to 15% of the total program cost. The relative SRM&QA cost factors specified here are intended to require substantive differences in the SRM&QA programs (and the associated costs) for the various program classifications in order to establish a meaningful ladder of cost/risk levels.

Press #fwhite SPACE#d to continue').

close window ().
end. (* *1 *)

topic '*2'.
Window (' Note *2 ',yellow,red,white,6,6,68,7).
say ('#e
Outside/Independent Review refers to reviews conducted by personnel who are not managing or directly associated with the program or the design effort (i.e., personnel who are not assigned to or in the management chain for the program and who are not preforming or managing the design effort).

Press #fwhite SPACE#d to continue').

close window ().
end. (* *2 *)

close_window ()
and
collect ().
column = ?column - 1.
row = ?row - 1.
end. (* Class B *)

topic 'Class C'.
column = ?column + 1.
row = ?row + 1.
window ('Class "C" Payload Characterization',blue,white,white,?column,?row,72,17).
say ('#e

===== CHARACTERIZATION, MISSION SUCCESS, AND SRM&QA COST GUIDELINES ======
FOR CLASS C PAYLOADS

CLASSIFICATION:

CLASS C

Characterization:

Medium Priority, Medium/High Risk

Typical Factors Used to Determine Payload Classifications:

Moderate national prestige;
Short hardware life required;
Medium to low complexity;
Medium cost;
Short program duration;
Few launch constraints;
Retrieval/reflight or in-flight maintenance to recover from problems may be feasible.
Achievement of Mission Success Criteria:

Moderate risks of not achieving mission success are accepted to permit significant cost savings. Reduced product assurance requirements are allowed.

Estimated Relative SRM&QA Cost Factors: $0.4 \times \text{Class A}$

GUIDELINES FOR SRM&QA PROGRAM REQUIREMENTS FOR PAYLOADS

<table>
<thead>
<tr>
<th>SRM&amp;QA ELEMENT:</th>
<th>Engineering Model, Prototype, Flight And Spare Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Engineering Model Hardware For New Designs. &quot;Protoflight&quot; Hardware (In Lieu Of Separate Prototype And Flight Model(s). Limited Flight Spare Hardware (For Long Lead Or Difficult To Replace Flight Unit(s).)</td>
</tr>
</tbody>
</table>

| Failure Investigation Board Requirements | Formal Board Required - Initiated And Conducted by Cognizant Field Center. |


| EEE Parts | Grade II (per Mil-Std-975) or Upgraded Commercial. |

| Reviews | Limited Formal Reviews And Project Level Reviews. |

| Safety | Per All Applicable NASA Safety Standards. |


| Reliability | FMEA/CIL Requirements Determined At The Project Level. Analysis Of Interfaces Required. Parts Stress Analysis Required For All Parts And Circuits. |
Maintainability
Maintainability Considered During Design, Sparing Provided Where Economical.

Quality Assurance
Formal Inspection System Including Problem Reporting.

Software #m*2#m
Software Quality Assurance Program With Product Level Reviews.

Press #fyellow SPACE#d to continue.’).

There are wide variations in the methods for specifying and accounting for "SRM&QA costs". For Class A Programs, these costs are typically in the range of 10 to 15% of the total program cost. The relative SRM&QA cost factors specified here are intended to require substantiative differences in the SRM&QA programs (and the associated costs) for the various program classifications in order to establish a meaningful ladder of cost/risk levels.

Press #fwhite SPACE#d to continue’).

Outside/Independent Review refers to reviews conducted by personnel who are not managing or directly associated with the program or the design effort (i.e., personnel who are not assigned to or in the management chain for the program and who are not performing or managing the design effort).

Press #fwhite SPACE#d to continue’).

open window (‘Class "D" Payload Characterization’,blue,white,white,?column,?row,72,17).
say (‘#e

===== CHARACTERIZATION, MISSION SUCCESS, AND SRM&QA COST GUIDELINES ======
FOR CLASS D PAYLOADS

---

---
CLASSIFICATION:

Characterization:

Typical Factors Used to Determine Payload Classifications:

Little national prestige;
Short hardware life required;
Low complexity;
Low cost;
Short program duration;
Non-critical launch time/orbit;
Reflyable or economically replaceable, in-flight maintenance may be feasible.

Achievement of Mission Success Criteria:

Significant risks of not achieving mission success are accepted to permit minimum cost. Minimal product assurance requirements are allowed.

Estimated Relative SRM&QA Cost Factors $m\times 1\times m$:

0.1 x Class A

GUIDELINES FOR SRM&QA PROGRAM REQUIREMENTS FOR PAYLOADS

SRM&QA ELEMENT:

Limited Engineering Model And Flight Spare Hardware Hardware.

Failure Investigation Board Requirements

Failure Investigation Initiated And Conducted by Cognizant Field Center - Formal Board Not Required.

Treatment of Single Failure Points (SFPs)


Qualification, Acceptance, And Protoflight Test Program


EEE Parts

Commercial

Reviews $m\times 2\times m$

Per All Applicable NASA Safety Standards.

Safety
Materials
Requirements Based On Applicable Safety Standards.

Reliability
Analysis Requirements Based On Applicable Safety Requirements. Analysis Of Interfaces Required.

Maintainability
Requirements Based On Applicable Safety Standards.

Quality Assurance
Requirements Based On Applicable Safety Standards.

Software 
Requirements Based On Applicable Safety Guidelines.

Press #fyellow SPACE#d to continue.

There are wide variations in the methods for specifying and accounting for "SRM&QA costs". For Class A Programs, these costs are typically in the range of 10 to 15% of the total program cost. The relative SRM&QA cost factors specified here are intended to require substantiative differences in the SRM&QA programs (and the associated costs) for the various program classifications in order to establish a meaningful ladder of cost/risk levels.

Press #fwhite SPACE#d to continue.

Outside/Independent Review refers to reviews conducted by personnel who are not managing or directly associated with the program or the design effort (i.e., personnel who are not assigned to or in the management chain for the program and who are not preforming or managing the design effort).

Press #fwhite SPACE#d to continue.

close_window ()
end. (* *1 *)

topic '*2'.
Window (' Note #2 ',yellow,red,white,6,6,68,7).
say ('#e

Outside/Independent Review refers to reviews conducted by personnel who are not managing or directly associated with the program or the design effort (i.e., personnel who are not assigned to or in the management chain for the program and who are not preforming or managing the design effort).

Press #fwhite SPACE#d to continue.

close_window ()
end. (* *2 *)

close_window ()
and collect ()
column = ?column - 1.
row = ?row - 1.
end. (* Class D *)

topic 'Nasa Management Instruction 8010.1A'.
new_kb ('NMI.hkb').
end. (* NMI * )
This Instruction establishes policies and responsibilities associated with classification of airborne and space payloads for which NASA has Design, Development, Test, and Evaluation responsibility.

2. APPLICABILITY AND SCOPE

a. This Instruction is applicable to NASA Headquarters and field installations.

b. This Instruction should be implemented for all NASA payload flight equipment for which DDT&E responsibility is assigned to a field installation after release of the NMI. Applicability to equipment already assigned is at the discretion of the cognizant Headquarters Program Office.

3. CLASSIFICATION PROVISIONS

Classification provides a basis for:

a. Mutual understanding among all involved organizations and individuals of the general approach that is to be taken relative to cost versus confidence trade decisions for specific equipment.

b. Pre-selection of the official responsible for initiating and conducting investigations of in-flight, nonhazardous, nonpropagating failure of specific equipment.

c. Structuring of field installation and Headquarters prepared guidelines, standards, or policies in such areas as environmental design and test; electrical, electronic, and electromechanical parts control; materials controls; configuration control; conduct of design reviews; and other elements that are amenable to tailoring for application to each class.
d. Structuring of experience data so that effectiveness of approaches associated with each class can be evaluated and adjustments made accordingly.

---

4. DEFINITIONS

For the purpose of this Instruction, the following definitions apply:

a. Payload - Any airborne or space equipment or material that is not an integral part of the carrier vehicle \( \#40 \) i.e., not part of the carrier aircraft, balloon, sounding rocket, expendable or recoverable launch vehicle\( \#41 \). Included are items such as free-flying automated spacecraft, Spacelab payloads, Space Station payloads, flight hardware designed to conduct either coherent sets of experiments \( \#40 \) e.g., "Facility Class" instruments\( \#41 \) or individual experiments, and payload support equipment. When used in this Instruction the term "payload" is intended to include only payloads for which NASA has DDT&E responsibility.

b. Payload Class - The NASA designation classifying each program/payload according to the criticality of the mission to NASA and national objectives, program cost, and the acceptable level of risk of a partial or complete failure.

5. LIMITATIONS

All payloads must conform with all applicable NASA safety requirements. In the event of any conflict with this Instruction, the safety requirements shall take precedence.

6. POLICIES

a. NASA will recognize the four classes of payloads which are defined and characterized in Attachment A to this Instruction. The characterizations for Classes A through D contained in Attachment A are intended to serve as guidelines to define a meaningful ladder of cost/risk combinations for NASA payloads.

b. Any equipment that constitutes a payload, or part of a payload, may be separately classified \( \#40 \) e.g., payload instrument, experiment, or support equipment\( \#41 \). For example, a Class A free flyer may incorporate instruments from Classes A and B. A Class A Spacelab or Space Station payload may incorporate instruments from all four classes.

c. Guidelines for program Safety, Reliability, Maintainability, and Quality Assurance \( \#40 \) SRM&QA\( \#41 \) and related requirements for Classes A through D are provided in Attachment B. Field installation and Headquarters organizations shall develop detailed policies, standards, and/or guidelines to adapt and expand upon the examples in Attachment B as necessary.
establish a consistent set of requirements appropriate for the unique needs of their various payloads. Each subset of requirements described by the examples in Attachment B and in the corresponding detailed documents which are prepared is intended to serve as an approved starting point for establishment of a complete set of DDT&E criteria and requirements tailored to the needs of a specific project.

d. In all cases, nothing in this Instruction is intended to limit or constrain the flexibility of a project to deviate from the guidelines in Attachment A or B, provided that the concurrence of cognizant field installation organizations and Headquarters Program Offices are obtained and documented.

e. Regardless of class designation, all payloads are to be developed and fabricated using sound management, engineering and manufacturing practices. Cost savings are to be achieved by taking steps to balance the use of resources to expected benefits appropriate for each class. Cost/risk tradeoffs shall be considered in determining requirements for NASA management visibility and day-to-day control, implementation of NASA-imposed "how-to" specifications and requirements, deliverable documentation, redundancy, spares, engineering and prototype model hardware, parts and material controls, inspection and audit controls and qualification/acceptance test programs.

7 RESPONSIBILITIES

a. Headquarters Program Offices are responsible for:

(1) Assigning the class designation for each program/payload or payload element. The level of assembly selected for classification will usually consist of an integrated free flyer, or, in the case of Attached Payloads, an instrument. However, the Headquarters Program Office may establish the class designation at whatever level of assembly it considers appropriate for each project.

(2) Establishing a set of mission success criteria for each program/payload or payload element which reflect the key objectives of the program. Such criteria will typically be easily measurable and expressed in terms of the science objectives to be achieved, remote sensing of a planet for 2 years or the service to be provided, 95 percent availability of a single access link for 5 years.

(3) Defining acceptable risks for each program/payload or payload element and/or mission success criteria. Acceptable risks may be defined in terms of minimum
reliability requirements, critically of achieving the various mission success criteria, acceptable failures, or other appropriate criteria.

(4) Notifying the Office of Safety and Mission Quality #40OSMQ#41 in writing of the assigned payload classifications.

(5) Initiating and conducting investigations of in-flight, nonhazardous, nonpropagating failure of specific Class A and B equipment in accordance with Attachment B. For specific Class B equipment, the Headquarters Program Office may delegate this responsibility to the cognizant NASA field center.

b. Field installations are responsible for:

(1) Recommending a class designation for their proposed or assigned payload equipment.

(2) Recommending an appropriate classification breakdown into lower levels of assembly if payloads incorporate noncritical elements #40e.g., instruments or experiment hardware#41 for which lower class designations may be appropriate and cost effective.

(3) Planning and implementing a balanced development/acquisition effort that is consistent with paragraph 6e of this Instruction and incorporates the applicable subset of requirements shown in Attachment B. Where deviations from any of the guidelines in Attachment B are necessary, or where reclassifications are considered necessary, the Headquarters Program Office shall be coordinated with, and any changes documented and approved by the Headquarters Program Office, with a copy to the OSMQ.

(4) Maintaining documentation for each payload project showing current class designation together with a description of any deviations from the guidelines in Attachment B. This information should be documented in the Summary of Technical Plan section of the Project Plan and updated such that the current status is reflected by the latest Project Plan.

(5) Informing the cognizant Headquarters Program Office and the OSMQ, in writing, of all significant in-flight failures that occur during baseline or extended missions. Significant failures are those that result in failure to achieve payload objectives or degradation of payload performance, reliability, retrievability, or repairability to a level considered undesirable by the cognizant project office.

(6) Initiating and/or conducting investigations of in-flight, nonhazardous, nonpropagating failure of specific Class B-D
equipment in accordance with Attachment B.

c. The Office of Safety and Mission Quality is responsible for:

(1) Exercising general oversight and agencywide coordination of implementation of this Instruction.

(2) Serving as an agencywide focal point for collection and correlation of payload classification and in-flight failure information, and dissemination of lessons learned therefrom.

(3) Supporting Headquarters Program Offices in the development and review of payload class designations.

8. CANCELLATION

NMI 8010.1 dated September 26, 1979.

ATTACHMENTS:
A. Characterization, Mission Success and SRM&QA Cost Guidelines for Class A-D Payloads.
B. Guidelines for SRM&QA Program Requirements for Class A-D Payloads.

DISTRIBUTION:
SDL 1

------------------------------------------------------------------------------------------------------------------------------------
ATTACHMENT A - CHARACTERIZATION, MISSION SUCCESS AND SRM&QA COST GUIDELINES FOR CLASS A-D PAYLOADS
------------------------------------------------------------------------------------------------------------------------------------

CLASSIFICATION: Class A
Characterization: High Priority, Minimum Risk

Typical Factors Used to Determine Payload Classifications:
High national prestige; Long hardware life required; High complexity; Highest cost; Long program duration; Critical launch constraints; Retrieval/reflight or in-flight maintenance to recover from problems is not feasible.

Achievement of Mission Success Criteria:
All affordable programmatic and other measures are taken to achieve minimum risk. The highest practical product assurance standards are utilized.

Estimated Relative* SRM&QA Cost Factors:
1.0

CLASSIFICATION: Class B
Characterization: High Priority, Medium Risk

Typical Factors Used to Determine Payload Classifications:
High national prestige; Medium hardware life required; High to medium complexity; High cost; Medium program duration; Retrieval/reflight or in-flight maintenance to recover from problems is difficult or not feasible.

Achievement of Compromises are used to permit somewhat reduced costs
Mission Success Criteria: while maintaining a low risk to the overall mission success and a medium risk of achieving only partial success. Stringent product assurance standards are utilized.

Estimated Relative* SRM&QA Cost Factors: 0.7 X Class A

CLASSIFICATION: Class C
Characterization: Medium Priority, Medium/High Risk

Typical Factors Used to Determine Payload Classifications: Moderate national prestige; Short hardware life required; Medium to low complexity; Medium cost; Short program duration; Few launch constraints; Retrieval/reflight or in-flight maintenance to recover from problems may be feasible.

Achievement of Mission Success Criteria: Moderate risks of not achieving mission success are accepted to permit significant cost savings. Reduced product assurance requirements are allowed.

Estimated Relative* SRM&QA Cost Factors: 0.4 X Class A

CLASSIFICATION: Class D
Characterization: High Risk, Minimum Cost

Typical Factors Used to Determine Payload Classifications: Little national prestige; Short hardware life required; Low complexity; Low cost; Short program duration; Non-critical launch time/orbit; Reflyable or economically replaceable, in-flight maintenance may be feasible.

Achievement of Mission Success Criteria: Significant risk of not achieving mission success is accepted to permit minimum costs. Minimal product assurance requirements are allowed.

Estimated Relative* SRM&QA Cost Factors: 0.1 X Class A

* There are wide variations in the methods for specifying and accounting for "SRM&QA costs". For Class A programs, these costs are typically in the range of 10 to 15% of the total program cost. The relative SRM&QA cost factors specified here are intended to require substantive differences in the SRM&QA programs and the associated costs for the various program classifications in order to establish a meaningful ladder of cost/risk levels.

ATTACHMENT B - GUIDELINES FOR SRM&QA PROGRAM REQUIREMENTS FOR CLASS A-D PAYLOADS

CLASSIFICATION: Class A

SRM&QA ELEMENTS:

Engineering Model, Prototype, Flight Engineering Model Hardware for new or modified designs. Separate Prototype and Flight Model Hardware. Full set
and Spare Hardware of assembled and tested "Flight Spare" replacement units.

Failure Investigation Board Requirements

Formal Board Required - Initiated and Conducted by Headquarters.

Treatment of Single Failure Points #49SFPs#49

Success Critical SFP’s are not permitted except by Formal Project Waiver. Retention of Unavoidable SFP’s requires Justification based on Risk Analysis and Implementation of Measures to Mitigate Risk.

Qualification Acceptance, and Protoflight Test Program

Full Formal Qualification and Acceptance Test Programs at all hardware levels. Extensive Design Margin and Development Testing for New or Modified Designs.

EEE Parts Reviews*

Grade I #40per MIL-STD-975#41.

Full Formal Review Program including Outside/Independent Reviews.

Safety Materials

Per all applicable NASA Safety Standards.

Verify heritage of previously used materials and qualify all new or changed materials. Utilize source controls on procured materials and Acceptance Test each lot/batch.

Reliability

Failure Mode and Effects Analysis/Critical Items List #40FMEA/CIL#41, worst case performance and parts stress analyses required for all parts and circuits. Mechanical reliability and other reliability analyses required where appropriate.

Maintainability

Formal Maintainability Program for all appropriate flight and critical ground support elements.

Quality Assurance

Formal Quality Assurance Program including closed-loop problem reporting and analysis of SRM&QA and performance trends.

Software*

Formal Software Quality Assurance Program including a formal review program with Outside/Independent Reviews and independent verification and validation.

CLASSIFICATION:

Class B

SRM&QA ELEMENTS:

Engineering Model, Prototype, Flight and Spare Hardware

Engineering Model Hardware for new or significantly modified designs. "Protoflight" hardware #40in lieu of separate prototype and flight models#41 except where Extensive Qualification Testing is anticipated. Spare #40or refurbishable prototype#41 hardware as needed to avoid major program impact in flight units must be replaced.

Failure Investigation

Formal Board Required - Initiated by Headquarters; may be conducted by Congnizant Field Center #40see Par.7a#405#
## Board Requirements

<p>| Treatment of Single Failure Points | Success Critical SFP's are allowed w/o formal waiver but are minimized and mitigated by use of high reliability parts and additional testing. Essential spacecraft functions and key instruments are typically fully redundant. Other hardware has partial redundancy and/or provisions for graceful degradation. |
| Qualification Acceptance, and Protoflight Test Program | Formal Qualification and Acceptance Test Programs may be combined using &quot;Protoflight&quot; hardware approach. Design margin and development testing for new designs. |
| EEE Parts Reviews* | Combination Grade I and Grade II per MIL-STD-975. |
| Safety Materials | Formal Review Program; may include Outside/Independent Reviews. |
| Reliability | Per all applicable NASA Safety Standards. |
| Maintainability | Utilize Previously Tested/Flown Materials or qualify new materials. Acceptance test each lot of procured materials. |
| Quality Assurance | FMEA/CIL required at a block box or circuit block diagram level. Worst case performance and parts stress analyses required for all parts and circuits. |
| Software* | Formal Maintainability Program for selected flight and ground support elements. |
| CLASSIFICATION: | Formal Quality Assurance Program including closed-loop problem reporting and analysis of SRM&amp;QA and some performance trends. |
| SRM&amp;QA ELEMENTS: | Formal Software Quality Assurance Program including a formal review program and software verification and validation; may include Outside/Independent Reviews. |
| Engineering Model, Prototype, Flight and Spare Hardware | Engineerig Model Hardware for New Designs. &quot;Protoflight&quot; Hardware in lieu of separate Prototype and flight models. Limited flight spare hardware for long lead or difficult to replace flight units. |
| Failure Investigation Board Requirements | Formal Board Required - Initiated and conducted by Cognizant Field Center. |
| Treatment of Single Failure Points #49SFPs#49 | Success Critical SFP's are allowed w/o formal waiver. Single string and partially single string design approaches are commonplace. |
| Qualification Acceptance, and Protoflight Test | Limited Qualification Testing for new aspects of the design plus full acceptance test program. Qualification testing required for verification of safety compliance. |</p>
<table>
<thead>
<tr>
<th>Program</th>
<th>and interface compatibility. Limited Development Testing.</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEE Parts</td>
<td>Grade II #40per MIL-STD-975#41 or upgraded commercial.</td>
</tr>
<tr>
<td>Reviews*</td>
<td>Limited Formal Reviews and Project Level Reviews.</td>
</tr>
<tr>
<td>Safety</td>
<td>Per all applicable NASA Safety Standards.</td>
</tr>
<tr>
<td>Materials</td>
<td>Utilize previously tested/flown materials or characterize new materials. Acceptance test sample lots of procured materials.</td>
</tr>
<tr>
<td>Reliability</td>
<td>FMEA/CIL Requirements determined at the project level. Analysis of interfaces required. Parts stress analysis required for all parts and circuits.</td>
</tr>
<tr>
<td>Maintainability</td>
<td>Maintainability considered during design. Sparing provided where economical.</td>
</tr>
<tr>
<td>Quality Assurance</td>
<td>Formal Inspection System including Problem Reporting.</td>
</tr>
<tr>
<td>Software*</td>
<td>Software Quality Assurance Program with Project Level Reviews.</td>
</tr>
<tr>
<td><strong>CLASSIFICATION:</strong></td>
<td><strong>Class D</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SRM&amp;QA ELEMENTS:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Prototype, Flight</td>
<td></td>
</tr>
<tr>
<td>and Spare Hardware</td>
<td></td>
</tr>
<tr>
<td>Failure Investigation</td>
<td>Failure Investigation initiated and conducted by Congnizant Field Center - Formal Board Not Required</td>
</tr>
<tr>
<td>Board Requirements</td>
<td></td>
</tr>
<tr>
<td>Treatment of Single</td>
<td>Success Critical SFP's are allowed w/o Formal Waiver. Single string and partially single string design approaches are commonplace.</td>
</tr>
<tr>
<td>Failure Points</td>
<td></td>
</tr>
<tr>
<td>#49SFPs#49</td>
<td></td>
</tr>
<tr>
<td>Qualification</td>
<td>Qualification Testing Required only for verification of safety compliance and interface compatibility. Acceptance test program for critical performance parameters.</td>
</tr>
<tr>
<td>Acceptance, and</td>
<td></td>
</tr>
<tr>
<td>Protoflight Test</td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td></td>
</tr>
<tr>
<td>EEE Parts</td>
<td>Commercial.</td>
</tr>
<tr>
<td>Reviews*</td>
<td>Project Level Reviews.</td>
</tr>
<tr>
<td>Safety</td>
<td>Per all applicable NASA Safety Standards.</td>
</tr>
<tr>
<td>Materials</td>
<td>Requirements based on applicable safety standards.</td>
</tr>
<tr>
<td>Reliability</td>
<td>Analysis Requirements based on applicable safety requirements. Analysis of interface required.</td>
</tr>
<tr>
<td>Maintainability</td>
<td>Requirements based on applicable safety standards.</td>
</tr>
</tbody>
</table>
Quality Assurance Requirements based on applicable safety standards.

Software* Requirements based on applicable safety standards.

* Outside/Independent Review refers to reviews conducted by personnel who are not managing or directly associated with the program or the design effort. i.e., personnel who are not assigned to or in the management chain for the program and who are not performing or managing the design effort.

Press "SPACE" to Continue."

close window ()
end. (* Display Document *)
The Implementation Plan includes these three sections:

- **IMPLEMENTATION APPROACH**
- **SUMMARY WORK BREAKDOWN STRUCTURE**
- **DOCUMENTATION**

Please indicate whether the project is an in-house activity or a contracted activity. Define the principle entities on the project team, including the contractor team and its responsibilities. If the contractor team has not yet been selected, include a description of the contractor procurement approach and schedule.

Press the yellow RETURN KEY to enter the editor, ESC to continue.
Please provide a summary Work Breakdown Structure (WBS) down to the "appropriate level" (as agreed to between the Project Manager, and the Program Manager) and a brief description of each WBS element. The WBS structure should be related to line items on the project master schedule and to the resources section, and it should serve as the baseline WBS to be used in future budget reporting.

Press yellow RETURN KEY to enter the editor, yellow ESC to leave editor, and yellow RETURN to confirm save.'}.

Please provide a list of documentation that includes all governing, hardware development, mission integration, and safety documentation.

Press yellow RETURN KEY to enter the editor, yellow ESC to leave editor, and yellow RETURN to confirm save.'
topic 'list of documentation'.
column = ?column + 1.
row = ? row + 1.
window('List of Documentation',yellow,red,yellow,?column,?row,71,16).
say ('#e
Documents Subject to Formal Baselining and Configuration Control
by APCGF

<table>
<thead>
<tr>
<th>Document</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Science Requirements</td>
<td>PI</td>
</tr>
<tr>
<td>Project Implementation Plan</td>
<td>Project Office</td>
</tr>
<tr>
<td>Instrument Interface Agreement</td>
<td>Integrator</td>
</tr>
<tr>
<td>Interface Control Document</td>
<td>Developer and/or Integrator</td>
</tr>
<tr>
<td>Safety Plan</td>
<td>Developer and/or Integrator</td>
</tr>
<tr>
<td>Verification Plan</td>
<td>Developer and/or Integrator</td>
</tr>
<tr>
<td>Experiment Requirement Document</td>
<td>Developer</td>
</tr>
<tr>
<td>End Item Specification</td>
<td>Project Office</td>
</tr>
<tr>
<td>Drawings</td>
<td>Developer</td>
</tr>
<tr>
<td>Test and Checkout Requirements and Specifications Document</td>
<td>Project Office</td>
</tr>
<tr>
<td>Instrument Program and Component List</td>
<td>Developer</td>
</tr>
<tr>
<td>Operations and Maintenance Requirements/Manual</td>
<td>Developer</td>
</tr>
<tr>
<td>Software Requirements/Manual</td>
<td>Developer</td>
</tr>
</tbody>
</table>

Any proposed change to items controlled above shall be approved by the Project Manager. Changes to MSAD-controlled milestones, schedules and growth module specifications shall be submitted through the same channel. All contracts and agreements are based on project requirements. Technical requirements are set forth in specifications and technical documents. Other requirements concerning management and operating functions necessary for conduct of the project are specified in the appropriate requirements documents.

For all apparatus in development or planning, contractual requirements will ensure that all applicable documentation necessary for flight qualification is produced. At each level and at significant points in the APCGF Project, baselines will be established, and changes to these baselines will be tracked until a revised baseline is formally established. The accompanying #mtable#m provides a sample listing of those documents that shall be required for acceptance of flight hardware.

Press #fwhite SPACE#d to continue.'}.
column = ?column - 1.
row = ?row - 1.
close window ( ).
close window ( ).
end. (* documentation *)
Documentation to Be Included in Acceptance Data Package

Log Book
Mass Properties Data
Waivers/Deviations
Test Procedure/Results
Operation and Maintenance Requirements/Manual
Drawing/Engineering Change Orders
Calibration Data
Materials Usage List/Maintenance Requirements/Manual
Safety Compliance Data
Verification Data
Special Handling Data
Cleanliness Certification
Pressure Vessel Log (If Applicable)
Discrepancy/Problem Reports
Acceptance/Preship Review Data (shortages, open work, red tag items, shipping document, certifications)

Press #yellow SPACE#d to continue.'

Press #yellow SPACE#d to continue.'

Press #yellow SPACE#d to continue.'
(* MGTPLAN.KB *)
(* THIS PROGRAM IS FOR COMPLETING THE MANAGEMENT PLAN SECTION OF THE *)
(* PROJECT PLAN *)

column = 2.
row = 2.
no_edit_key ().
noc debug ().
curdir is read line ('CURDIR.DAT').
curdir = string replace (?curdir,' ','',8).
menu_choice = ' '.
menu_option is ['PROJECT MANAGEMENT RESPONSIBILITIES AND ORGANIZATION',
'MISSION MANAGEMENT RESPONSIBILITIES AND ORGANIZATION',
'RETURN TO PREVIOUS MENU'].
while ?menu_choice <> 'RETURN TO PREVIOUS MENU'
then do (MPoutline).

topic 'MPoutline'.
column = ?column + i.
row = ?row + 1.
window ('4.6 Management Plan',white,red,yellow,?column,?row,76,17).
ask ('#e

The Management Plan Section includes these two parts:' ,

menu_choice,?menu_option).
if ?menu_choice = 'PROJECT MANAGEMENT RESPONSIBILITIES AND ORGANIZATION'
then do ('Project Management Responsibilities and Organization').
if ?menu_choice = 'MISSION MANAGEMENT RESPONSIBILITIES AND ORGANIZATION'
then do ('Mission Management Responsibilities and Organization').
if ?menu_choice = 'RETURN TO PREVIOUS MENU'
then new_kb ('content.hkb').

close window ().
column = ?column - 1.
row = ?row - 1.
end. (* Management Plan *)
new_kb ('content.hkb').

topic 'Project Management Responsibilities and Organization'.
relatedfile3 = '\PPE4_5_2.dat'.
column = ?column + 1.
row = ?row + 1.
window ('4.6.1 Project Management Responsibilities and Organization',blue,
white,white,?column,?row,72,12).
say ('#e

Please provide an organization chart of the project, including
the Center Director, Project Manager, Principle Investigator,
Project Scientist, and other key positions on the project, and
all external project interfaces. Relate the organization chart
to the WBS (#mSection 4.5.2#m).

Press #y yellow SPACE#d to continue.' ).
Please identify the NASA center responsible for mission management and describe what process is to be used to define the management and technical interface agreements between the NASA center responsible for the project and the Mission Management Center. Indicate the mission management responsibilities and interfaces on the organization chart prepared for Section 4.6.1. Identify all assumptions that will affect project plans, schedules and costs.

Press #f yellow RETURN KEY#d to enter the editor, #f yellow ESC#d to leave editor, and #f yellow RETURN#d to confirm save.'

This is your answer for Section 4.5.2: #t #n,
#related_answer, '#n #n #n #n #n
Press #f yellow SPACE#d to continue.'

Press #f yellow RETURN RETURN KEY#d to enter the editor, #f yellow ESC#d to leave editor, and #f yellow RETURN#d to confirm save.'
column = 2.
row = 2.
no_edit_key ()
no_debug ()
curdir is read_line ('CURDIR.DAT').
curdir = string_replace(?curdir,' ','','','8).
menu_choice = ''.
menu_options is ['RESOURCES','COST CONTROL GUIDELINES','COST REPORTING AND CONTROL STRUCTURE','COST CONTROL STRATEGY','RETURN TO PREVIOUS MENU'].
while ?menu_choice <> 'RETURN TO PREVIOUS MENU'
    then do (CCoutline).

topic 'CCoutline'.
column = ?column + 1.
row = ?row + 1.
window ('4.8 Cost Control',white,red,white,?row,?column,76,17).
ask ('#e The Cost Control Plan Includes these four sections:','menu_choice,?menu_options).

if ?menu_choice = 'RESOURCES'
    then do ('Resources').

if ?menu_choice = 'COST CONTROL GUIDELINES'
    then do ('Cost Control Guidelines').

if ?menu_choice = 'COST REPORTING AND CONTROL STRUCTURE'
    then do ('Cost Reporting and Control Structure').

if ?menu_choice = 'COST CONTROL STRATEGY'
    then do ('Cost Control Strategy').

if ?menu_choice = 'RETURN TO PREVIOUS MENU'
    then new_kb ('content.hkb').

close_window ().
column = ?column - 1.
row = ?row - 1.
end. (* Cost Control Plan *)
new_kb ('content.hkb').

topic 'Resources'.
relatedfile4 = '\PPE4_4_8.DAT'.
column = ?column + 1.
row = ?row + 1.
window ('4.8.1 Resources',blue,white,white,?column,?row,72,15).
say ('#e Please specify the funding (including contingency) and man-
power requirements needed for the life of the project against the #mWBS#m. These resource requirements should be consistent with}
results from the Independent Cost Review held in conjunction with the RDR. Identify any assumptions made in determining the cost. Select the appropriate WBS level to be the Cost Reporting level. Identify any use to be made of other facilities for which another entity (other than the project) will be financially responsible. (This should be consistent with facility usage described in Section 4.4.8.)

Press yellow RETURN KEY to enter the editor, yellow ESC to leave editor, and yellow RETURN to confirm save.'

Please define the cost control guidelines to be used in coping with cost variations at the different project levels.

The following are yellow examples of guidelines that yellow may be used and are not intended to be universally applied. Each project can develop its own guidelines according to the individual project needs.

a. MSAD Allowance for Program Adjustment (APA) is to be utilized for changes in program scope imposed by external circumstances, such as change in launch date or launch vehicle and internal changes in the science requirements.

b. Project Office Contingency is to be utilized to fix internal problems for the PI's or hardware contractor, such as design changes, parts procurement, alternate approaches, and change of subcontractors.

c. Contingency reserves at the lower WBS levels (if they exist) will be used to solve problems in those WBS elements. Only when problems cannot be solved within the WBS element are they to be referred to the next higher cost control point.

Press yellow RETURN KEY to enter the editor, yellow ESC to leave editor, and yellow RETURN to confirm save.'
row = ?row + 1.
menu_option2 is ['NASA REPORTS','CONTRACTOR REPORTS','RETURN'].
menu_choice2 = ' '.
while ?menu_choice2 <> 'RETURN'
    then do ('cost reporting').
    new_kb ('costcont.hkb').
topic 'cost reporting'.
    window ('4.8.3 Cost Reporting and Control Structure',blue,white,white, ?column,?row,74,15).
    ask ('#e
        This section includes these two parts:'),menu_choice2,? menu_option2).
    if ?menu_choice2 = 'NASA REPORTS'
        then do ('NASA REPORTS').
    if ?menu_choice2 = 'CONTRACTOR REPORTS'
        then do ('CONTRACTOR REPORTS').
    if ?menu_choice2 = 'RETURN'
        then new_kb ('costcont.hkb').
end. (* cost reporting *)
close_window ().
column = ?column - 1.
row = ?row - 1.
end. (* Cost Reporting and Control Structure *)
topic 'NASA Reports'.
column = ?column + 1.
row = ?row + 1.
window ('4.8.3.1 NASA Reports',blue,white,white, ?column,?row,72,11).
say ('Please show the #mWBS#m structure to be used for cost reporting and control, and identify the monthly Management Information Control System (MICS) (or equivalent) and Project Operating Plan (POP) reporting format to be used. These should result from negotiations between the Project and Program Managers and will vary in detail depending on project size and complexity.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.
').
edit_file (concat (C:\GARDEN,?CURDIR,'\E4_8_3_1.DAT'),yellow,black,red,6,12,70, close_window ()).
close (concat (C:\GARDEN,?CURDIR,'\E4_8_3_1.DAT')).
column = ?column - 1.
row = ?row - 1.
end. (* NASA Reports *)
topic 'Contractor Reports'.
column = ?column + 1.
row = ?row + 1.
window ('4.8.3.2 Contractor Reports',blue,white,white,?column,?row,72,11).
say ('#e
Where applicable, please delineate the contractor reports to be provided to the NASA center.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C:\GARDEN\,?CURDIR,'\E4_8_3_2.DAT'),yellow,black,red,6,12,7C
close_window ().
close (concat (C:\GARDEN\,?CURDIR,'\E4_8_3_2.DAT')).
column = ?column - 1.
row = ?row - 1.
end. (* Contractor Reports *)

topic 'Cost Control Strategy'.
column = ?column + 1.
row = ?row + 1.
window ('4.8.4 Cost Control Strategy',blue,white,white,?column,?row,72,13).
say ('Discuss the courses of action to be followed in the event of cost growths. The entire hardware development process should be considered, including the test plans, support equipment, and facility requirements, etc., as well as the mainline flight hardware development.

Your discussion should include the prioritization of science requirements, science and hardware development descope options, and appropriate decision points. Program stretch-out options should also be addressed if appropriate. Obviously, this section should be developed with a high degree of participation by the Principle Investigator and Project and Program Scientists. Overall discussions on cost management, such as how to deal with overruns and underruns at the lower #mWBS#m and project levels, should also be included.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C:\GARDEN\,?CURDIR,'\PPE4_8_4.DAT'),yellow,black,red,6,12,7C
close_window ().
close (concat (C:\GARDEN\,?CURDIR,'\PPE4_8_4.DAT')).
column = ?column - 1.
row = ?row - 1.
end. (* Cost Control Strategy *)

topic 'Section 4.4.8'.
column = ?column + 1.
row = ?row + 1.
related_answer is read(concat(C:\GARDEN\,?curdir,?relatedfile4)).
window ('4.4.8 Facilities',blue,white,white,?column,?row,73,15).
say ('This is your answer for Section 4.4.8: #t #n',
  ?related_answer, '#n #n #n #n
  Press #fyellow SPACE#d to continue.
').
column = ?column - 1.
row = ?row - 1.
close_window ().
end. (* Section 4.4.8 *)

query 'RDR'.
column = ?column + 1.
row = ?row + 1.
window ('RDR',blue,white,white,?row,?column,72,14).
say('Requirements Definition Review

Press #fyellow SPACE#d to continue.

Close window ()
column = ?column - 1.
row = ?row -1.
end. (* RDR *)

query 'WBS'.
column = ?column + 1.
row = ?row + 1.
window ('WBS',blue,white,white,?row,?column,70,10).
say('Work Breakdown Structure

Press #fyellow SPACE#d to continue.

Close window ()
column = ?column - 1.
row = ?row -1.
end. (* WBS *)
(* Projrev.kb *)
(* This is the file for the completion of the Project Reviews and *)
(* Meetings section of the Project Plan. *)

column = 2.
row = 2.
no_edit_key ().
no_debug ().
curdir is read_line ("CURDIR.DAT").
curdir = string_replace (?curdir,"",",",8).
menu_choice = "".
menu_options is ['INTERNAL REVIEWS', 'EXTERNAL REVIEWS',
'DESIGN AND READINESS REVIEWS' , 'RETURN TO PREVIOUS MENU'].

while ?menu_choice <> 'RETURN TO PREVIOUS MENU'
  then do (PRoutln).
topic 'PRoutln'.
column = ?column + 1.
row = ?row + 1.
window ('4.9 Project Reviews and Meetings',white,red,yellow,?column,?row,
76,17).
ask ('#e
  Project Reviews and Meetings includes the following topics:
  menu_choice,?menu_options).

if ?menu_choice = 'EXTERNAL REVIEWS'
  then do ('External Reviews').

if ?menu_choice = 'INTERNAL REVIEWS'
  then do ('Internal Reviews').

if ?menu_choice = 'DESIGN AND READINESS REVIEWS'
  then do ('Design and Readiness Reviews').

if ?menu_choice = 'RETURN TO PREVIOUS MENU'
  then new_kb ('content.hkb').

close_window ().
column = ?column - 1.
row = ?row - 1.

end. (* Project Reviews and Meetings *)
new_kb ('content.hkb').

topic 'Internal Reviews'.
column = ?column + 1.
row = ?row + 1.
window ('4.9.1.1 Internal Reviews',blue,white,white,?column,?row,72,11).
say ('#e
  Please identify weekly or monthly project meetings and
  reviews with the NASA center’s management as appropriate.

  Press #fyellow RETURN KEY#d to enter editor, #fyellow ESC#d
  to leave editor, and #fyellow RETURN#d to confirm save.’).

edit_file (concat (C:\GARDEN\,?CURDIR,'\E4_9_1_1.DAT'),yellow,black,red,6,12,70,
Please identify biweekly telecons, monthly reviews, or periodic program reviews with NASA Headquarters, Mission Management, etc., as appropriate.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

Please make reference to the major project reviews, including those required by the MSAD management plan as well as integrated payload reviews, safety reviews, and flight readiness reviews.

Press #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.
APPENDIX G

Science Requirements Envelope Document Software Listing
(*ENVMMENU.KB This is the main menu of the Science Requirements *)

Envelope Document

no_edit_key ().
no_debug ()
action = ' '.
nasaloop = 1.
glossary_load = 0.
column = 3.
row = 3.
yn = [YES,NO].

menu_option is ['HOW TO USE THE SYSTEM',
'PROJECT SELECTION',
'SCIENCE REQUIREMENTS ENVELOPE DOCUMENT (OVERVIEW)',
'GLOSSARY/ACRONYMS',
'PRINT GLOSSARY/ACRONYMS','EXIT SYSTEM'].
menu_choice = ''.

while ?menu_choice <> 'EXIT SYSTEM'
    then do (nasamenu).

    topic 'nasamenu'.
    window (' ',3,3,77,18).
    ask (['#e #s
        Please enter your choice of activities from the list.'],menu_choice, ?menu_option).

if ?menu_choice = 'HOW TO USE THE SYSTEM'
    then new_kb ('envintro.hkb').

if ?menu_choice = 'SCIENCE REQUIREMENTS ENVELOPE DOCUMENT (OVERVIEW)'
    then new_kb ('envover.hkb').

if ?menu_choice = 'PROJECT SELECTION'
    then new_kb ('eproject.hkb').

if ?menu_choice = QUIT
    then stop ().

if ?menu_choice = 'GLOSSARY/ACRONYMS'
    then glossary_load = (?glossary_load + 1)
    and
do (glossary).

if ?menu_choice = 'PRINT GLOSSARY/ACRONYMS'
    then
        ask ('#e
            The printing of the glossary/acronym list can require
            a significant amount of time (5 - 10 minutes depending
            on your system). Are you sure you want to print the
glossary at this time?',printok,?yn)
        and
        if ?printok = NO
            then new_kb ('ENVMMENU.CKB')
        else
window (,white,red,yellow,1,16,27,4)
and
WRITE ('con:',
' GLOSSARY is being printed.

Please stand by. ')
and
glossary_print is read ('ENVTERMS.DAT','//KSC')
and
glossary_print is string_replace(?glossary_print,'//','')
and
glossary_print is string_replace(?glossary_print,'/end','')
and
print (?glossary_print)
and
glossary_print is read ('ENVTERMS.DAT','//KSC')
and
glossary_print is string_replace(?glossary_print,'//','')
and
glossary_print is string_replace(?glossary_print,'/end','')
and
print (?glossary_print)
and
close_window ().

(* close_window (). *)
(* *********************************************************** *)
topic glossary.
window ('LISTING OF NASA GLOSSARY AND ACRONYMS',blue,white,white,1,1,80,20).

if ?glossary_load = 1
then
  window (,white,red,yellow,1,16,27,4)
  and
  WRITE ('con:',
    ' A slight delay will occur while the glossary is loaded. Please stand by. ')
and
glossary_text is read ('ENVINDEX.DAT')
and
close_window ()
and
close ('ENVTERMS.DAT').
say (?glossary_text).
close_window ().
end. (* glossary *)
topic mark (find_string).
column = ?column + 1.
row = ?row + 1.
text is read ('ENVTERMS.dat',concat('//',?find_string),'/end').
window (?find_string,blue,white,white,?column,?row,72,).
say (?text).
column = column - 1.
row = row - 1.
close_window ().
end.  (* mark *)

end.  (* envmmenu *)
This is the introductory screen for the Science Requirements Envelope Document.

yn is [YES,NO].
column = 3.
row = 3.
no_edit_key ()
no_debug ()
tried = 0.

do (so_you_want_to_fly).
new_kb ('envmmenu.ckb').
topic so_you_want_to_fly.
say (''
#bmagenta So you want to fly on the Shuttle.#d

Well, before you can, we must get a little information about your experiment and its requirements.

If you have flown with us in the past, you may remember a substantial amount of paper documentation was required. This application, the #mAutomated Payload Experiment Tool#m, is designed to alleviate much of the burden of the document preparation and maintenance process, by utilizing a #mhypertext#m, knowledge-based system. This system can be used to prepare one of our support documents, the #mScience Requirements Envelope Document#m, which provides an envelope or volume of science requirements for a type of experimentation.

Press #fyellow SPACE#d to continue.'').

if ?tried = 0 then
  column = ?column + 1
  and
  row = ?row + 1
  and
  window (' ', white, red, white)
  and
  say (''
  For more information on a highlighted topic, just move the mouse to that word and click. The information will immediately be displayed. If you are not using a mouse, please use the function keys as indicated at the bottom of the screen.

  For multiple page definitions, please use the #fyellow Page Up#d and #fyellow Page Down#d keys to scroll back and forth through the pages. Multiple page displays are indicated by the #fyellow Page x of x#d message at the lower right of the screen.

  For help at anytime throughout the application, select the #fyellow Fl#d key. This will retrieve location sensitive
help information, and may be called from the system or system-called edit screens.

This will be the method by which support documentation will be retrieved throughout this application.

Press #fyellow SPACE#D to continue.

and
close_window ()
and
tried = 1
and
column = ?column - 1
and
row = ?row - 1
and
do (so_you_want_to_fly).

topic mark (find_string).
column = ?column + 1.
row = ?row + 1.
text is read ('nasaterm.dat',concat('//',?find_string),'/end').
window (?find_string,blue,white,white,?column,?row,72,).
say (?text).
column = ?column - 1.
row = ?row - 1.
close_window ()
end. (* mark *)

end. (* so_you_want_to_fly *)
This program lists the outline for the Science Requirements Envelope Document.

no_edit_key ()
no_debug ()
row = 3.
column = 3.
(* show_outline = 0. *)
width = 72.
length = 12.

do (soutline).

new_kb ('eoutline.hkb').

(* Topic mark allows for the use of hypertext. *)

_topic mark (find_string).
row = ?row + 1.
column = ?column + 1.
text is read ('NASATERM.DAT',concat ('/','?find_string)','/end').
window (?find_string, blue, white, white,?column,?row, 76,17).
say (?text).
close_window ()
row = ?row - 1.
column = ?column -1.

end. (*topic mark*)

topic soutline.
(*if show_outline = 0 then
show outline = ?show outline + 1. *)
window ('Overview and Explanation',blue,white,white,?column,?row,77,18).
say ('#e

#fyellow PURPOSE#d

The Science Requirements Envelope Document provides an envelope or volume of science requirements for a type of experimentation which is intended to encompass the science requirements generated by individual experiments of that type. The primary purpose of the document is to provide science requirements against which hardware can be conceptualized such that later, when specific PIs are chosen, their individual requirements will fall within the requirements originally stated in the Science Requirements Envelope Document.

#fyellow FUNCTION#d

The Science Requirements Envelope Document should provide:

1. A general description of the type of scientific investigations to be performed.

2. A formal establishment of the scientific rationale for conducting the type of scientific investigation defined.
3. A formal establishment of the science objectives of the type of experiments to be performed.

4. A general description of the observational measurement, environmental, and data requirements which accompany this type of experimentation.

Preparation Approval and Updating

The Discipline Project Scientist, utilizing inputs from a Science Development Team, Discipline Working Groups or other sources, will prepare and submit the Science Requirements Envelope Document. The Discipline Program Scientist will concur with the document and the MSAD Chief Scientist will approve the initial document and changes thereto. The document will be maintained under change control by the appropriate Center and will be used to drive conceptual hardware development until such time as the Hardware Capabilities Document has been developed and approved. At the appropriate time, to be determined by the MSAD Chief Scientist, the document will be abandoned and the individual Science Requirements Documents will be used to establish the requirements on the hardware.

Press SPACE to continue.'}. end. (* soutline *)
(* EOUTLINE.KB *)
(* THIS PROGRAM IS LETS THE USER VIEW THE QUESTIONS TO BE FILLED OUT *)
(* THAT ARE NECESSARY FOR THE COMPLETION OF THE ENVELOPE DOCUMENT *)
no_edit_key ().
no_debug ().
column = 3.
row = 3.
menu_choice = ' '.
menu_option is ['INTRODUCTION/SUMMARY','BACKGROUND',
'JUSTIFICATION FOR CONDUCTING THIS EXPERIMENT IN SPACE',
'DESCRIBPTION OF EXPERIMENT TYPES','SCIENCE REQUIREMENTS ENVELOPE',
'OTHER REQUIREMENTS','RETURN TO MAIN MENU'].

while ?menu_choice <> 'RETURN TO MAIN MENU'
  then do (view_outline).

new_kb ('envmmenu.hkb').
topic 'view outline'.
window ('Overview and Explanation',blue,white,white,?row,?column,77,18).
ask ('#e The Outline for the Science Requirements Envelope Document:',
  menu_choice,?menu_option).

if ?menu_choice = 'INTRODUCTION/SUMMARY'
  then do ('Introduction/Summary').

if ?menu_choice = 'BACKGROUND'
  then do ('Background').

if ?menu_choice = 'JUSTIFICATION FOR CONDUCTING THIS EXPERIMENT IN SPACE'
  then do ('Justification for Conducting This Experiment In Space').

if ?menu_choice = 'DESCRIPTION OF EXPERIMENT TYPES'
  then do ('Description of Experiment Types').

if ?menu_choice = 'SCIENCE REQUIREMENTS ENVELOPE'
  then new_kb ('SHOWSRE.HKB').

if ?menu_choice = 'OTHER REQUIREMENTS'
  then do ('Other Requirements').

if ?menu_choice = 'RETURN TO MAIN MENU'
  then new_kb ('envmmenu.ckb').
end. (* view outline *)

(*-----------------------------Threaded topics-----------------------------*)

topic 'Introduction/Summary'.
  column = ?column + 1.
  row = ?row + 1.
  window ('1.0 Introduction/Summary',blue,white,white,
    ?column,?row,76,17).
  say ('#e Provide a brief discussion describing the following areas:

    1.1 Description of Experiment Type or Class
1.2 Scientific Knowledge to be Gained From This Type of Experimentation

1.3 Value of Knowledge of This Type of Experimentation to Scientific Field

1.4 Necessity for Space Environment to Experiment Type

Press #fyellow SPACE#d to continue.’).

end. (*Introduction/Summary*)

topic 'Background'.
column = ?column + 1.
row = ?row + 1.
window ('2.0 Background',blue,white,white,?column,?row, 76,17).
say ('#e Provide a brief discussion describing the following areas:

2.1 Scientific Field to which the Experiment Type Belongs

2.2 Current Application for Research in the Field

#m2.3 Brief Historical Account of Prior Research in the Field#m

#m2.4 Current Research#m

2.5 Relationship of Proposed Experiment type to Scientific Field

2.6 Anticipated Advance in State of the Art for This Type of Experimentation.

Press #fyellow SPACE#d to continue.’).

end. (*Background*)

topic 'Justification for Conducting This Experiment in Space'.
column = ?column + 1.
row = ?row + 1.
window ('3.0 Justification for Conducting This Experiment in Space', blue,white,white,?column,?row,76,17).
say ('#e Provide a brief discussion describing the following areas:
3.1 Limitations of Ground-Based Testing
3.2 Limitations of Drop Towers
3.3 Limitations of Testing in Aircraft
3.4 Need for Accommodations in the Shuttle
3.5 Limitations of Mathematical Modeling
3.6 Limitations of Other Modeling Approaches

Press #fyellow SPACE#d to continue.

end. (*Justification for Conducting This Experiment in Space*)

Provide a detailed description of the following areas:

4.1 General Description of Type of Experiments

4.2 Types of Experiment Procedures to be Used.

4.3 Types of measurements and ranges of values required

Press #fyellow SPACE#d to continue.

end. (*Description of Experiment Types*)

2.3 Brief Historical Account of Prior Research in the Field

This section summarizes previously conducted studies
not including current research and results.

Press #fyellow SPACE#d to continue.  

close_window ()

row = ?row - 1.
column = ?column - 1.
end. (*2.3 Brief historical account of prior research*)

topic '2.4 Current research'.
row = ?row + 1.
column = ?column + 1.
window ('2.4 Current research',blue,white,white,?column,?row, 76,16).
say ('#e

This section summarizes the most recently conducted studies or related activities and their results.

Press #fyellow SPACE#d to continue.  

close_window ()
row = ?row - 1.
column = ?column - 1.
end. (*2.4 Current research*)

topic 'Other Requirements'.
row = ?row + 1.
column = ?column + 1.
window ('6.0 Other Requirements',blue,white,white,?column,?row,76,17).
say ('#e

Be sure to describe any other applicable material which is not presently addressed in these requirements.

Press #fyellow SPACE#d to continue.').

close_window ().
row = ?row - 1.
column = ?column - 1.
end. (*6.0 Other Requirements*)

(*====================end subtopics====================*)
(* SHOWSRE.KB *)
(* THIS PROGRAM IS FOR DISPLAYING THE SCIENCE REQUIREMENTS ENVELOPE *)
(* SECTION OF THE SCIENCE REQUIREMENTS ENVELOPE *)

column = 3.
row = 3.
no_edit_key ().
no_debug ().

curdir is read_line ('curdir.dat').
curdir = string_replace (?curdir,' ', '', 8).

menu_choice = ' '.

menu_option is ['EXPERIMENT SAMPLE REQUIREMENTS',
'RANGE OF ATMOSPHERIC REQUIREMENTS', 'TEMPERATURE CONTROL AND MEASUREMENT',
'VIBRATION CONTROL AND MEASUREMENT', 'TEST MATRICES',
'IMAGING REQUIREMENTS', 'ELECTROMAGNETIC LIMITATIONS', 'ASTRONAUT INVOLVEMENT',
'DATA REQUIREMENTS', 'TELEPRESENCE, TELEROBOTICS', 'RETURN TO PREVIOUS MENU'].

while ?menu_choice <> 'RETURN TO PREVIOUS MENU'
  then do (outline).

  topic 'outline'.
  
    column = ?column + 1.
    row = ?row + 1.
    window ('Overview and Explanation', ,white,white,3,3,77,18).
    window ('5.0 Science Requirements Envelope',blue,white,white,?column,?row,76,17).
    ask ('#e

      Provide a brief discussion describing the following areas: ',menu_choice, ?menu_option).

    if ?menu_choice = 'EXPERIMENT SAMPLE REQUIREMENTS'
      then do ('Experiment Sample Requirements').

    if ?menu_choice = 'RANGE OF ATMOSPHERIC REQUIREMENTS'
      then do ('Range of Atmospheric Requirements').

    if ?menu_choice = 'TEMPERATURE CONTROL AND MEASUREMENT'
      then do ('Temperature Control and Measurement').

    if ?menu_choice = 'VIBRATION CONTROL AND MEASUREMENT'
      then do ('Vibration Control and Measurement').

    if ?menu_choice = 'TEST MATRICES'
      then do ('Test Matrices').

    if ?menu_choice = 'IMAGING REQUIREMENTS'
      then do ('Imaging Requirements').

    if ?menu_choice = 'ELECTROMAGNETIC LIMITATIONS'
      then do ('Electromagnetic Limitations').

    if ?menu_choice = 'ASTRONAUT INVOLVEMENT'
      then do ('Astronaut Involvement').

    if ?menu_choice = 'DATA REQUIREMENTS'
      then do ('Data Requirements').
if \texttt{?menu\_choice} = 'TELEPRESENCE, TELEROBOTICS' \then\ do ('Telepresence, Telerobotics').

if \texttt{?menu\_choice} = 'RETURN TO PREVIOUS MENU' \then\ new\_kb ('EOUTLINE.HKB').

\begin{verbatim}
close\_window ().
close\_all ().
row = ?row - 1.
column = ?column -1.
end. (* Science Requirements*)
\end{verbatim}

\begin{verbatim}
topic 'Experiment Sample Requirements'.
row = ?row + 1.
column = ?column + 1.
window ('5.1 Experiment Sample Requirements',blue,white,white, ?column,?row,76,16).
say ('#e
Be sure to include the number of samples and the materials used to conduct the experiment. Then make sure each one is justified and/or substantiated in the documentation.

Press \#fyellow SPACE\#d to continue.').
close\_window ().
row = ?row - 1.
column = ?column - 1.
end. (* Experiment sample requirements*)
\end{verbatim}

\begin{verbatim}
topic 'Range of Atmospheric Requirements'.
row = ?row + 1.
column = ?column + 1.
window ('5.2 Atmospheric Requirements',blue,white,white, ?column,?row,76,16).
say ('#e
Be sure to include any information related to, including the range of values for:
\begin{itemize}
  \item Pressure
  \item Gas composition
  \item Humidity
  \item Vacuum
\end{itemize}

Then make sure each one is justified and/or substantiated in the documentation.

Press \#fyellow SPACE\#d to continue.').
close\_window ().
row = ?row - 1.
column = ?column - 1.
\end{verbatim}
end. (* Range of Atmospheric requirements*)

**Temperature Control and Measurement**

```
row = ?row + 1.
column = ?column + 1.
window ('5.3 Temperature Control and Measurement',blue,white,white,
?column,?row,76,16).
say ('#e

As much accuracy as possible is required when describing this section and each part must be justified and/or substantiated in the documentation.

Press #fyellow SPACE#d to continue.’).
```

```
close_window () .

row = ?row - 1.
column = ?column - 1.
end. (* Temperature control and measurement*)

**Vibration Control and Measurement**

```
row = ?row + 1.
column = ?column + 1.
window ('5.4 Vibration Control and Measurement',blue,white,white,
?column,?row,76,16).
say ('#e

As much accuracy as possible is required, along with the frequency of measurement, when describing this section.

Each part must be justified and/or substantiated in the documentation.

Press #fyellow SPACE#d to continue.’).
```

```
close_window () .

row = ?row - 1.
column = ?column - 1.
end. (* Vibration control and measurement*)

**Test Matrices**

```
row = ?row + 1.
column = ?column + 1.
window ('5.5 Test Matrices',blue,white,white,?column,?row,76,16).
say ('#e

Include the number of tests and the required duration of each test when describing this section. Each requirement must be justified and/or substantiated in the documentation.

Press #fyellow SPACE#d to continue.’).
```

```
close_window () .
```
row = ?row - 1.
column = ?column - 1.
end. (* Test matrices*)

topic 'Imaging Requirements'.
row = ?row + 1.
column = ?column + 1.
window ('5.6 Imaging Requirements', blue, white, white, ?column, ?row, 76, 16).
say ('#e
Be sure to include any information related to:

  o Photography
  o Radiography
  o Television
  o Resolution
  o Frame rate

Each requirement must be justified and/or substantiated in the documentation.

    Press #fyellow SPACE#d to continue.').
close_window ()..

row = ?row - 1.
column = ?column - 1.
end. (* Imaging requirements*)

topic 'Electromagnetic Limitations'.
row = ?row + 1.
column = ?column + 1.
window ('5.7 Electromagnetic Limitations', blue, white, white, ?row, ?column, 76, 16).
say ('#e
Please list any type of electromagnetic limitations for this type of experiment.

    Press #fyellow SPACE#d to continue.').
close_window ()..

column = ?column - 1.
row = ?row - 1.
close_window ()..
end. (* Electromagnetic Limitations *)

topic 'Astronaut Involvement'.
row = ?row + 1.
column = ?column + 1.
window ('5.8 Astronaut Involvement', blue, white, white, ?column, ?row, 76, 16).
say ('#e
Be sure to include any information related to:

  o Extravehicular activity (EVA)
  o Activation of experiment
Each of the requirements must be justified and/or substantiated in the documentation.

Press #fyellow SPACE#d to continue.'

close_window ().

row = ?row - 1.
column = ?column - 1.
end. (* Astronaut involvement*)

topic 'Data Requirements'.
column = ?column + 1.
row = ?row + 1.
window ('5.9 Typical Data Requirements',blue,white,white,?row,?column,76,16).
say ('#e

Please list any typical data requirements for this type of experiment.

Press #fyellow SPACE#d to continue.').

column = ?column - 1.
row = ?row - 1.
close_window ()..
end. (* Data Requirements *)

topic 'Telepresence, Telerobotics'.
row = ?row + 1.
column = ?column + 1.
window ('5.10 Telepresence, Telerobotics',blue,white,white, ?column,?row,76,16).
say ('#e

Each requirement must be justified and/or substantiated in the documentation.

Press #fyellow SPACE#d to continue.').

close_window ()

row = ?row - 1.
column = ?column - 1.
end. (*5.10 Telepresence, telerobotics*)
This is the activity menu to allow the user to select an activity to perform on an existing project.

no_edit_key ()
no_debug ()
do_gloss = 1.
today = date ()
month = element(?today, 1).
day = element(?today, 2).
year = element(?today, 3).
today = concat(?month, '/', ?day, '/', ?year).
yn is [YES, NO].
curdir is read line ('CURDIR.DAT').
close (concat('C:\GARDEN', ?CURDIR, '\BASELINE.DAT'))
curdir is string replace (?curdir, '', '', 8).
eof = number_to_char (26).

menu_option is ['FILL OUT DOCUMENTATION', 'PRINT SCIENCE REQUIREMENTS ENVELOPE DOCUMENT', 'DISPLAY ENVELOPE DOCUMENT', 'CREATE AN ASCII FILE OF ENVELOPE DOCUMENT', 'BASELINE DOCUMENT', 'COMPARE BASELINE TO CURRENT REVISION', 'COPY ANSWERS TO DISK', 'RETURN TO PREVIOUS MENU', 'RETURN TO MAIN MENU', 'QUIT']. menu_choice = ' '.

while ?menu_choice <> QUIT then do (nasamenu).

topic nasamenu.

ask ([#{e #s
    The project you have selected is: ', ?curdir, ' #d #n

    Please enter your choice of activities from the list.'}, ?menu_option]).

if ?menu_choice = 'FILL OUT DOCUMENTATION' then new_kb ('envfmenu.hkb').

if ?menu_choice = 'RETURN TO PREVIOUS MENU' then new_kb ('EPROJECT.HKB').

if ?menu_choice = 'RETURN TO MAIN MENU' then new_kb ('envmmenu.CKB').

if ?menu_choice = 'PRINT SCIENCE REQUIREMENTS ENVELOPE DOCUMENT' then new_kb ('ENVPRINT.HKB').

if ?menu_choice = 'DISPLAY ENVELOPE DOCUMENT' then new_kb ('ENVDISP.HKB').

if ?menu_choice = 'CREATE AN ASCII FILE OF ENVELOPE DOCUMENT' then new_kb ('ENVFILE.HKB').

if ?menu_choice = 'BASELINE DOCUMENT' then do (baseline_rtn).

if ?menu_choice = 'COPY ANSWERS TO DISK' then do (copyfiles).
if ?parse Choice = 'COMPARE BASELINE TO CURRENT REVISION' then do (compare_rtn).

if ?parse Choice = QUIT then stop ().

topic 'copyfiles'.

ask ('Do you want to copy your answers to a different drive?', wantcopy, ?yn).
if ?wantcopy = YES then drivelist is [A:, B:, C:, D:, NONE] and
  ask ('Please choose the drive to which you wish to copy the files:', drive_destination, ?drivelst).

if ?wantcopy = YES and ?drive_destination <> NONE then
  say ('Please insert diskette now if you are copying to a floppy drive.

  Please press #fyellow SPACE#d when ready. ')

  and
  move_cursor (1, 10)
  and
  dos (?copy_command, restore)
  and
  say ('#e

  Answers have been moved to drive #s', ?drive_destination, ' #n #n

  Please press #fyellow SPACE#d to exit. ')

  do (nasamenu).
end. (* copyfiles *)

topic 'baseline_rtn'.

ask ('#e

At some point in the documentation process, it is necessary to declare that all documents are complete, and that any changes to be made will be treated as revisions to the baseline document.

Do you want to baseline your answers at this time?', baseline, ?yn).

curbase = ?eof.
overwrite = YES.
if ?baseline = YES then
  curbase is read_line (concat('C:\GARDEN\', ?curdir, '\BASELINE.DAT')) and
  close (concat('C:\GARDEN\', ?curdir, '\BASELINE.DAT')) and
  if ?curbase <> ?eof
then
ask
(’#e
You have already baselined this experiment in the past. Do you want to take all revisions and overwrite your previous baseline to create a new baseline? ’, overwrite, ?yn).

if ?curbase = ?eof and ?baseline = YES
then
   window (,white,red,yellow,1,14,32,6)
   and
   say (’#e
This selection will invoke
a DOS command, which will
cause your screen to blank
out momentarily. Do not
be alarmed. Press #fyellow SPACE#d now. ’)
   and
   md_command = concat (’MD C:\GARDEN’,?curdir,’\BASELINE’)
   and
   dos (?md_command,restore)
   and
   copy_command = concat
   (’COPY C:\GARDEN’,?CURDIR,’*.DAT C:\GARDEN’,?CURDIR,’\BASELINE\*.’)
   and
   dos (?copy_command,restore)
   and
   write (concat(’C:\GARDEN’,?CURDIR,’\BASELINE.DAT’),?today)
   and
   close_window ()
   and
   say (’#e
Baseline document has been created. All changes to this
document will be stored in the revision. A new baseline
must be created to incorporate any revisions into the
final document.

Please press #fyellow SPACE#d to exit. ’).

if ?curbase <> ?eof and ?baseline = YES and ?overwrite = YES
then
   xcopy_command = concat
   (’XCOPY C:\GARDEN’,?CURDIR,’*.DAT C:\GARDEN’,?CURDIR,
   ’\BASELINE\*.’ /D:’,?CURBASE)
   and
   window (,white,red,yellow,1,14,32,6)
   and
   say (’#e
This selection will invoke
a DOS command, which will
cause your screen to blank
out momentarily. Do not
be alarmed. Press #fyellow SPACE#d now. ’)
   and
   dos (?xcopy_command,restore)
   and
   new_file (concat(’C:\GARDEN’,?CURDIR,’\BASELINE.DAT’))
All revisions have been incorporated in the baseline document. Addition changes to this document will be stored in a new revision. A new baseline must be created to incorporate any new revisions into the final document.

Please press #f yellow SPACE #d to exit. ’).

do (nasamenu).

e nd. (* baseline_rtn *)

topic 'compare_rtn'.

close (concat('C:\GARDEN\',?CURDIR,'\BASELINE.DAT')).

curbase is read_line (concat('C:\GARDEN\',?CURDIR,'\BASELINE.DAT'))

and
close (concat('C:\GARDEN\',?CURDIR,'\BASELINE.DAT'))

and

if ?curbase = ?eof
then

say ('#e

You have not yet baselined this experiment; therefore, no comparison is necessary.

Press #f yellow SPACE #d to continue’)

and

new_kb ('NASA_ENV.HKB').

comp_choices = ['RUN COMPARISON PROGRAM','DISPLAY COMPARISONS','PRINT COMPARISONS','RETURN TO PREVIOUS MENU'].

ask ('#e

Do you wish to run the comparison program to generate a new listing of differences between the baseline and revision, print or display the results of the most recent comparison, or exit this menu?’,comp_ans,?comp_choices).

if ?comp_ans = 'RETURN TO PREVIOUS MENU'
then new_kb ('NASA_ENV.HKB').

if ?comp_ans = 'RUN COMPARISON PROGRAM'
then
do (comp_pgm).

topic 'comp_pgm'.

comp_command = concat
('FC /a C:\GARDEN\',?CURDIR,'\*.DAT C:\GARDEN\',?CURDIR,
'\BASELINE\*. > C:\GARDEN',?CURDIR,'DIFFER.DAT').

dos (?comp_command,restore).
close (concat('C:\GARDEN',?CURDIR,'DIFFER.DAT')).

say ('#e

Files have been compared. Please use the display or print options to view the results of the comparison.

Press #yellow SPACE#d to continue.').

end. (* comp_pgm *)

if ?comp_ans = 'DISPLAY COMPARISONS'
then
    comp_file = read (concat(C:\GARDEN,?CURDIR,'DIFFER.DAT'))
    and
    say (?comp_file).

if ?comp_ans = 'PRINT COMPARISONS'
then
    comp_file = read (concat(C:\GARDEN,?CURDIR,'DIFFER.DAT'))
    and
    print (#p,?comp_file,#p).

do (nasamenu).

end. (* compare_rtn *)

end. (* NASA_ENV *)
no_edit_key ().
no_debug ().
eof = number_to_char (26).
curdir is read_line ('CURDIR.DAT').
curdir is string_replace(curdir,' ',',',8).
orig_curdir = curdir.
blankline = ' '.
overwrite = [].
yn = [YES,NO].
line_count = 0.
page_count = 0.
username = blankline.
column = 3.
row = 3.

window ('Create an ASCII File of the Envelope Document',
white,blue,white,column,row,78,19).

read_response ('#e
This routine will create an ASCII file of the Science
Requirements Envelope Document. Please enter the complete
file name you wish to create, including drive, file name,
and extension. For example, #fyellow A:ENVDOC.TXT#d. If you do
not wish to create a file, press the RETURN key without
entering a file name.

,,username).

username = string_replace(username,' ',',',20).

direct is dir(username).
if direct <> [] and username <> []
then
    ask ('#e
This file already exists. Do you wish to overwrite it?','overwrite,
?yn).

if username = blankline or username = ' ' or username = '' or
username = [] or overwrite = NO
then

    window (,white,red,yellow,1,16,27,4)
    and
    say ('#e
No file created.
Press SPACE to continue.')
    and
    new_kb ('NASA_ENV.HKB').

base_dir = [].
curbase is read_line (concat('C:\GARDEN\',CURDIR,'\BASELINE.DAT')).
close (concat('C:\GARDEN\',CURDIR,'\BASELINE.DAT')).
if curbase <> eof
    then
base_choice = ['BASELINE', 'CURRENT REVISION', 'NEITHER']
and
ask ('#e
Do you wish to create a file from the baseline or the current revision?', base_dir, ?base_choice)
and
if ?base_dir = BASELINE
    then
        curdir = concat (?curdir, '\BASELINE')
        and
        curdir = string_replace(?curdir, ' ', '', 8)
        and
        new_file ('C:\GARDEN\CURDIR.DAT')
        and
        write ('C:\GARDEN\CURDIR.DAT', ?curdir).
if ?base_dir <> NEITHER
    then
        do ('print_title_page')
        and
        remove_topic ('print_title_page')
        and
        do (Print_Routine).
if ?base_dir = BASELINE
    then
        curdir = ?orig_curdir
        and
        curdir = string_replace(?curdir, ' ', '', 8)
        and
        new_file ('C:\GARDEN\CURDIR.DAT')
        and
        write ('C:\GARDEN\CURDIR.DAT', ?curdir).

new_kb ('NASA_ENV.HKB').

topic 'print_title_page'.
no_edit_key ()..
filler = '.
eof = number_to_char (26).
blankline = 7.

expname is read(concat(C:\GARDEN, ?CURDIR, '\TITLE.DAT')).

expname is string_replace(?expname, ' ', '', 600).
expname is remove(?expname, '').
lines = list_length(?expname).
line_count = 1.
title_length = 0.

window (,white,red,yellow,1,16,27,4).
write ('con:','FILE CREATION IN #nPROGRESS...').

while ?line_count <= ?lines
    then
        cur_line = element(?expname, ?line_count)
        and
        cur_length = string_length(?cur_line)
line_count = ?line_count + 1
and
title_length = ?title_length + ?cur_length + 1
and
if ?cur_line == '
then
    expname = string_replace(?expname,?cur_line).

expname = concat(element(?expname,1),',
    element(?expname,2),',
    element(?expname,3),',
    element(?expname,4),',
    element(?expname,5),',
    element(?expname,6),',
    element(?expname,7),',
    element(?expname,8),',
    element(?expname,9),',
    element(?expname,10),',
    element(?expname,11),',
    element(?expname,12),',
    element(?expname,13),',
    element(?expname,14),',
    element(?expname,15))
.

title_length = ?title_length + 3.
counter = 1.
write_counter = 1.
line_one = '.
while ?counter < ?title_length
then
    char = string_copy(?expname,?counter,1)
and
    counter = ?counter + 1
and
    write_counter = ?write_counter + 1
and
if ?write_counter > 40 and ?char == '
then
    linelgth = string_length(?line_one)
and
    linelgth = 80 - ?linelgth
and
    linelgth = ?linelgth / 2
and
    linefiller = string_copy (?blankline,1,?linelgth)
and
    line_one = concat(?linefiller,?line_one,'#n',?char)
and
    write_counter = 1
and
    new_expname gets ?line_one
and
    line_one = '.
else
    line_one = concat(?line_one,?char).

linelgth = string_length(?line_one).
linelgth = 80 - ?linelgth.
linelgth = ?linelgth / 2.
linefiller = string_copy (?blankline,1,?linelgth).
line_one = concat(?linefiller,?line_one,'#n').
new_expname gets ?line_one.
new_expname = string_replace(?new_expname, '~/',' ',600).

authorfile = concat(C:\GARDEN, ?CURDIR, '\AUTHOR.DAT').

titlepage = ('#n#n#n#n#n#n#n#n#n#n#n#n#n').
titlepage gets 'Science Requirements Envelope Document'.
titlepage gets 'for:'.
titlepage gets ?new_expname.
titlepage gets '#n#n'.
pagedate = ?date.
mo_num = element(?pagedate,1).
if ?mo_num = 1
  then month = January
else
  if ?mo_num = 2
    then month = February
  else
    if ?mo_num = 3
      then month = March
    else
      if ?mo_num = 4
        then month = April
      else
        if ?mo_num = 5
          then month = May
        else
          if ?mo_num = 6
            then month = June
          else
            if ?mo_num = 7
              then month = July
            else
              if ?mo_num = 8
                then month = August
              else
                if ?mo_num = 9
                  then month = September
                else
                  if ?mo_num = 10
                    then month = October
                  else
                    if ?mo_num = 11
                      then month = November
                    else
                      if ?mo_num = 12
                        then month = December
                      else
                        month = ' '.
year = element(?pagedate,3).
reportdate = concat(?month,' ',?year).
datelgth = string_length(?reportdate).
datelgth = 80 - ?datelgth.
datelgth = ?datelgth / 2.
datefiller = string_copy (?blankline,1,?datelgth).
titlepage gets ' '.
titlepage gets ' '.
titlepage gets concat(?datefiller, ?reportdate).

basefile = concat('C:\GARDEN\', ?CURDIR, '\BASELINE.DAT').
baseline = read(?basefile).
if ?baseline ≠ ?eof
    then
        baseline = 'N/A'.

datelgth = string_length(?baseline).
datelgth = 80 - ?datelgth.
datelgth = ?datelgth / 2.
datefiller = string_copy (?blankline, l, ?datelgth).
titlepage gets ''
titlepage gets ' Baseline: '.
titlepage gets concat(?datefiller, ?baseline).

name is read_line (?authorfile).
name is string_replace(?name, 'NAME: ', '', 1).
name = string_replace(?name, ' ', '', 8).
namelgth = string_length(?name).
namelgth = 80 - ?namelgth.
namelgth = ?namelgth / 2.
namefiller = string_copy (?blankline, l, ?namelgth).
titlepage gets concat('#n#n#n#n#n#n#n#n#n#n#n#n', ?namefiller, ?name).

organization is read_line (?authorfile).
organization is string_replace(?organization, 'ORGANIZATION: ', '', 1).
organization = string_replace(?organization, ' ', '', 8).
organizationlgth = string_length(?organization).
organizationlgth = 80 - ?organizationlgth.
organizationlgth = ?organizationlgth / 2.
organizationfiller = string_copy (?blankline, l, ?organizationlgth).
titlepage gets concat(?organizationfiller, ?organization).

mcode is read_line (?authorfile).
mcode is string_replace(?mcode, 'MAIL CODE: ', '', 1).
mcode = string_replace(?mcode, ' ', '', 8).
mcode = string_replace(?mcode, ' ', ' ', 8).
mcode = string_replace(?mcode, ' ', '', 8).
mcodefiller = string_copy (?blankline, l, ?mcode).
titlepage gets concat(?mcodefiller, ?mcode).

street is read_line (?authorfile).
street is string_replace(?street, 'STREET: ', '', 1).
street = string_replace(?street, ' ', '', 8).
streetlgth = string_length(?street).
streetlgth = 80 - ?streetlgth.
streetlgth = ?streetlgth / 2.
streetfiller = string_copy (?blankline, l, ?streetlgth).
titlepage gets concat(?streetfiller, ?street).

city is read_line (?authorfile).
city = string_replace(?city, 'CITY, STATE, ZIP: ', '', 1).
city = string_replace(?city, ' ', '', 8).
city = string_replace(?city, ' ', ' ', 8).
citylgth = string_length(?city).
citylgth = 80 - ?citylgth.
citylgth = ?citylgth / 2.
cityfiller = string_copy (?blankline,1,?citylgth).
titlepage gets concat(?cityfiller,?city).

phone is read_line (?authorfile).
phone is string_replace(?phone,'PHONE: ',',',1).
phone is string_replace(?phone,' ',',',8).
phonelgth = string_length(?phone).
phonelgth = 80 - ?phonelgth.
phonelgth = ?phonelgth / 2.
phonefiller = string_copy (?blankline,1,?phonelgth).
titlepage gets concat(?phonefiller,?phone).
new_file (?username).
write (?username,titlepage,#p).
close_window ()..

dobeg (* print_title_page *)

topic 'Print_Routine'.

window (,white,red,yellow,1,16,27,4).
write ('con:', 'FILE CREATION IN #nPROGRESS...').
do (Print_ENV).

page_count = ?page_count + 1.
page_line = concat(' ',?page_count).
while ?line_count < 60
then
    env_page gets ?blankline
    and
    line_count = ?line_count + 1.
env_page gets ?page_line.
env_page gets '#p'.
write (?username,?env_page).
close_all ()..
close_window ()..
topic 'Print_ENV'.
env_page gets ?blankline.
env_page gets ?blankline.
filename = concat(C:\GARDEN\,?CURDIR,'\env1_1.DAT').
line is read_line (?filename).
env_page gets ' 1.0 INTRODUCTION/summary'.
env_page gets ?blankline.
env_page gets ' 1.1 Description of Experiment Type or Class'.
env_page gets ?blankline.
line_count = 6.
while ?line <> ?eof
then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
then do (page_advance).

env_page gets ' 1.2 Scientific Knowledge to be Gained From This Type of
env_page gets ?blankline.
line_count = line_count + 2.
filename = concat(C:\GARDEN, ?CURDIR, \env1_2.DAT).
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = line_count + 1.
if line_count > 56
    then do (page_advance).

env_page gets ' 1.3 Value of Knowledge of This Type of Experiment'.
env_page gets ?blankline.
line_count = line_count + 2.
filename = concat(C:\GARDEN, ?CURDIR, \env1_3.DAT).
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = line_count + 1.
if line_count > 56
    then do (page_advance).

filename = concat(C:\GARDEN, ?CURDIR, \env1_4.DAT).
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = line_count + 1.
if line_count > 56
    then do (page_advance).

filename = concat(C:\GARDEN, ?CURDIR, \env2_1.DAT).
line is read_line (?filename).
if line_count > 56
    then do (page_advance).

env_page gets ?blankline.
env_page gets ?blankline.
env_page gets ' 2.0 BACKGROUND'.
env_page gets ?blankline.
env_page gets ' 2.1 Scientific Field to which Experiment Belongs'.
env_page gets ?blankline.
line_count = line_count + 6.
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = line_count + 1.
if line_count > 56
    then do (page_advance).

env_page gets ' 2.2 Current Application for Research in the Field'.
env_page gets ?blankline.
line_count = line_count + 2.
filename = concat(C:\GARDEN, ?CURDIR, '\env2_1.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 2.3 Brief Historical Account of Prior Research'.
env_page gets ?blankline.
line_count = ?line_count + 1.
filename = concat(C:\GARDEN, ?CURDIR, '\env2_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 2.4 Current Research'.
env_page gets ?blankline.
line_count = ?line_count + 1.
filename = concat(C:\GARDEN, ?CURDIR, '\env2_4.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 2.5 Relationship of Proposed Experiment Type to Experiment'.
env_page gets ?blankline.
line_count = ?line_count + 1.
filename = concat(C:\GARDEN, ?CURDIR, '\env2_5.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 2.6 Anticipated Advance in State of the Art'.
env_page gets ?blankline.
line_count = ?line_count + 1.
filename = concat(C:\GARDEN, ?CURDIR, '\env2_6.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if line_count > 56
then do (page_advance).

filename = concat(C:\GARDEN,?CURDIR,’\env3_1.DAT’).
line is read_line (?filename).
if ?line_count > 56
  then do (page_advance).

env_page gets ?blankline.
env_page gets ?blankline.
env_page gets ’ 3.0 JUSTIFICATION FOR CONDUCTING THIS EXPERIMENT IN SPACE’. 
env_page gets ’ 3.1 Limitations of Ground-Based Testing’. 
env_page gets ?blankline.
line_count = ?line_count + 6. 
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1. 
if ?line_count > 56
    then do (page_advance).

env_page gets ’ 3.2 Limitations of Drop Towers’. 
env_page gets ?blankline.
line_count = ?line_count + 2. 
filename = concat(C:\GARDEN,?CURDIR,’\env3_2.DAT’).
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1. 
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ’ 3.3 Limitations of Testing in Aircraft’. 
env_page gets ?blankline.
line_count = ?line_count + 2. 
filename = concat(C:\GARDEN,?CURDIR,’\env3_3.DAT’).
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1. 
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ’ 3.4 Need for Accommodations in the Shuttle’. 
env_page gets ?blankline.
line_count = ?line_count + 2. 
filename = concat(C:\GARDEN,?CURDIR,’\env3_4.DAT’).
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1. 
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).
env_page gets ' 3.5 Limitations of Mathematical Modeling'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN,?CURDIR,'\env3_5.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 3.6 Limitations of Other Modeling Approaches'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN,?CURDIR,'\env3_6.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
do (page_advance).

filename = concat(C:\GARDEN,?CURDIR,'\env4_1.DAT').
line is read_line (?filename).
if ?line_count > 56
    then do (page_advance).
env_page gets ?blankline.
env_page gets ?blankline.
en-page gets ' 4.0 DESCRIPTION OF EXPERIMENT TYPES'.
en-page gets ?blankline.
en-page gets ' 4.1 General Description of Type of Experiments'.
en-page gets ?blankline.
line_count = ?line_count + 6.
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 4.2 Types of Experiment Procedures to be Used'.
en-page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN,?CURDIR,'\env4_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
en-page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 4.3 Types of Measurements and Ranges of Values Required'.

env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR,'\env4_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if line_count > 56
  then do (page_advance).

filename = concat(C:\GARDEN\, ?CURDIR,'\env5_1.DAT').
line is read_line (?filename).
if ?line_count > 56
  then do (page_advance).

env_page gets ?blankline.
env_page gets ?blankline.
env_page gets ' 5.0 SCIENCE REQUIREMENTS ENVELOPE'.
env_page gets ?blankline.
env_page gets ' 5.1 Experiment Sample Requirements'.
env_page gets ?blankline.
line_count = ?line_count + 6.
while ?line <> ?eof
  then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
  then do (page_advance).

env_page gets ' 5.2 Atmospheric Requirements'.
env_page gets ?blankline.
env_page gets ' 5.2.1 Pressure'.
env_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\GARDEN\, ?CURDIR,'\env5_2_1.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

env_page gets ' 5.2.2 Gas Composition'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR,'\env5_2_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets
    ' 5.2.3 Humidity'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_2_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets
    ' 5.2.4 Vacuum'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_2_4.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets
    ' 5.3 Temperature Control and Measurement'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets
    ' 5.4 Vibration Control and Measurement'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_4.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets
5.5 Test Matrices.

env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_5.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

env_page gets ' 5.6 Imaging Requirements'.
env_page gets ?blankline.
env_page gets ' 5.6.1 Photography'.
env_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_6_1.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

env_page gets ' 5.6.2 Radiography'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_6_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

env_page gets ' 5.6.3 Television'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_6_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

env_page gets ' 5.6.4 Resolution'.


env_page gets \texttt{?blankline}.
line\_count = \texttt{?line\_count} + 2.
filename = concat(\texttt{C:\GARDEN\,?CURDIR,\'\env5\_6\_4.DAT'}).
line is read line (?filename).
while \texttt{?line} <> \texttt{?eof}
    then do (read_file).
close (?filename).
line\_count = \texttt{?line\_count} + 1.
env_page gets \texttt{?blankline}.
if \texttt{?line\_count} > 56
    then do (page\_advance).

\texttt{5.6.5 Frame Rate}.

env_page gets \texttt{?blankline}.
line\_count = \texttt{?line\_count} + 2.
filename = concat(\texttt{C:\GARDEN\,?CURDIR,\'\env5\_6\_5.DAT'}).
line is read line (?filename).
while \texttt{?line} <> \texttt{?eof}
    then do (read_file).
close (?filename).
line\_count = \texttt{?line\_count} + 1.
env_page gets \texttt{?blankline}.
if \texttt{?line\_count} > 56
    then do (page\_advance).

\texttt{5.7 Electromagnetic Limitations}.

env_page gets \texttt{?blankline}.
line\_count = \texttt{?line\_count} + 2.
filename = concat(\texttt{C:\GARDEN\,?CURDIR,\'\env5\_7.DAT'}).
line is read line (?filename).
while \texttt{?line} <> \texttt{?eof}
    then do (read_file).
close (?filename).
line\_count = \texttt{?line\_count} + 1.
env_page gets \texttt{?blankline}.
if \texttt{?line\_count} > 56
    then do (page\_advance).

\texttt{5.8 Astronaut Involvement}.

\texttt{5.8.1 Extravehicular Activity}.

env_page gets \texttt{?blankline}.
line\_count = \texttt{?line\_count} + 4.
filename = concat(\texttt{C:\GARDEN\,?CURDIR,\'\env5\_8\_1.DAT'}).
line is read line (?filename).
while \texttt{?line} <> \texttt{?eof}
    then do (read_file).
close (?filename).
line\_count = \texttt{?line\_count} + 1.
env_page gets \texttt{?blankline}.
if \texttt{?line\_count} > 56
    then do (page\_advance).

env_page gets \texttt{?blankline}.

\texttt{5.8.2 Activation of Experiment}.

env_page gets \texttt{?blankline}.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR,'\envs_8_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 5.9 Data Requirements'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR,'\envs_9.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 5.10 Telepresence and Telerobotics'.
env_page gets ?blankline.
env_page gets ' 5.10.1 Telepresence'.
env_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\GARDEN\, ?CURDIR,'\envs_10_1.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 5.10.2 Telerobotics'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR,'\envs_10_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if line_count > 56
    then do (page_advance).

env_page gets ?blankline.
env_page gets ?blankline.
env_page gets ' 6.0 OTHER REQUIREMENTS'.
env_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\GARDEN\,$CURDIR,'\ENV6_1.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
do (page_advance).
end. (* Print_ENV *)

topic 'page_advance'.
page_count = ?page_count + 1.
page_line = concat(',
while ?line_count < 58
    then
        env_page gets ?blankline
        and
        line_count = ?line_count + 1.
    env_page gets ?page_line.
    env_page gets '#p'.
    write (?username,?env_page).
    env_page = [].
    env_page gets ?blankline.
    env_page gets ?blankline.
    line_count = 2.
collect ()..
end. (* page_advance *)

end. (* Print_Routine *)

topic 'read_file'.
if ?line_count > 56
    then do (page_advance).
    line = concat(,'?,?line).
    env_page gets ?line.
line is read_line (?filename).
line_count = ?line_count + 1.
end. (* read_file *)

end. (* Print_Routine * )
/* ENVDISP.KB is used to display all sections of the Envelope Document.           */
no_edit_key ().
no_debug ().
choice = [].
env_page = [ ].
eof = number_to_char (26).
curdir is read_line ('CURDIR.DAT').
curdir is string_replace(?curdir,' ','','8').
orig_curdir = ?curdir.
blankline = ' '</blankline = 0.
base_dir = [ ].
curbase is read_line (concat('C:\GARDEN\',?CURDIR,\'\BASELINE.DAT')).
close (concat('C:kGARDENk',?CURDIR,'kBASELINE.DAT')).
if ?curbase <> ?eof then
    base_choice = ['BASELINE','CURRENT REVISION','NEITHER']
    and
    ask ('#e Do yo wish to display documents from the baseline or from the current revision? ',base_dir,?base_choice)
    and
    if ?base_dir = BASELINE then
        curdir = concat (?curdir,'\BASELINE')
        and
        curdir = string_replace (?curdir,' ','','8')
        and
        new file ('C:\GARDEN\CURDIR.DAT')
        and
        write ('C:\GARDEN\CURDIR.DAT',?curdir)
        and
        do (display_routine).
if ?base_dir <> NEITHER then while ?choice <> QUIT then do (display_routine).
new_kb ('NASA_ENV.HKB').
topic 'display_Routine'.
sections = ['I.0 INTRODUCTION/SUMMARY',
    '2.0 BACKGROUND',
    '3.0 JUSTIFICATION FOR CONDUCTING THE EXPERIMENT IN SPACE',
    '4.0 DESCRIPTION OF EXPERIMENT TYPES',
    '5.0 SCIENCE REQUIREMENTS ENVELOPE',
    '6.0 OTHER REQUIREMENTS',
    'QUIT'].

window ('Display Science Requirements Envelope Document'
    ,white,blue,white,3,3,78,19).
ask ('#e Which section do you want to display?',choice,?sections).
if ?choice = 'I.0 INTRODUCTION/SUMMARY' then do ('I.0 INTRODUCTION/SUMMARY').
if ?choice = '2.0 BACKGROUND' then do ('2.0 BACKGROUND').
if ?choice = '3.0 JUSTIFICATION FOR CONDUCTING THE EXPERIMENT IN SPACE'
then do ('3.0 JUSTIFICATION FOR CONDUCTING THE EXPERIMENT IN SPACE').

if ?choice = '4.0 DESCRIPTION OF EXPERIMENT TYPES'
then do ('4.0 DESCRIPTION OF EXPERIMENT TYPES').

if ?choice = '5.0 SCIENCE REQUIREMENTS ENVELOPE'
then do ('5.0 SCIENCE REQUIREMENTS ENVELOPE').

if ?choice = '6.0 OTHER REQUIREMENTS'
then do ('6.0 OTHER REQUIREMENTS').

if ?choice = QUIT
then
  if ?base_dir = BASELINE
  then
    curdir = ?orig_curdir
    and
    curdir = string_replace (?curdir, ' ', ' ', 8)
    and
    new_file ('C:\GARDEN\CURDIR.DAT')
    and
    write ('C:\GARDEN\CURDIR.DAT', ?curdir)
    and
    new_kb ('nasa_env.hkb')
  else
    new_kb ('nasa_env.hkb')

end. (* display_ENV *)

topic '1.0 INTRODUCTION/SUMMARY'.
  page_count = 0.
  env_page gets ?blankline.
  env_page gets ?blankline.
  filename = concat(C:\GARDEN\, ?CURDIR, '\env_l_1.dat').
  line is read_line (?filename).
  env_page gets ' 1.0 INTRODUCTION/SUMMARY'.
  env_page gets ?blankline.
  env_page gets ' 1.1 Description of Experiment Type or Class'.
  env_page gets ?blankline.
  line_count = 6.
  while ?line <> ?eof
    then do (read_file).
  close (?filename).
  env_page gets ?blankline.
  line_count = ?line_count + 1.
  if ?line_count > 56
    then do (page_advance).
  env_page gets ' 1.2 Scientific Knowledge to be Gained From This Type of Experimentation'.
  env_page gets ?blankline.
  line_count = ?line_count + 3.
  filename = concat(C:\GARDEN\, ?CURDIR, '\env_l_2.DAT').
  line is read_line (?filename).
  while ?line <> ?eof
    then do (read_file).
  close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 1.3 Value of Knowledge of This Type of Experiment to Scientific Field'.
env_page gets ?blankline.
line_count = ?line_count + 3.
filename = concat(C:\GARDEN\,?CURDIR,'\env1_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 1.4 Necessity for Space Environment to Experiment Type'.
env_page gets ?blankline.
line_count = ?line_count + 3.
filename = concat(C:\GARDEN\,?CURDIR,'\env1_4.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
do (page_advance).
end. (* 1.0 INTRODUCTION/SUMMARY *)

topic '2.0 BACKGROUND'.
page_count = 0.
filename = concat(C:\GARDEN\,?CURDIR,'\env2_1.DAT').
line is read_line (?filename).
if ?line_count > 54
    then do (page_advance).

env_page gets ?blankline.
env_page gets ?blankline.
env_page gets ' 2.0 BACKGROUND'.
env_page gets ?blankline.
env_page gets ' 2.1 Scientific Field to which Experiment Type Belongs'.
env_page gets ?blankline.
line_count = ?line_count + 6.
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 2.2 Current Application for Research in the Field'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,'\env2_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
   then do (page_advance).

env_page gets ' 2.3 Brief Historical Account of Prior Research in the
env_page gets ?blankline.
line_count = ?line_count + 1.
filename = concat(\GARDEN\,?CURDIR,'\env2_3.dat').
line is read_line (?filename).
while ?line <> ?eof
   then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
   then do (page_advance).

env_page gets ' 2.4 Current Research'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(\GARDEN\,?CURDIR,'\env2_4.dat').
line is read_line (?filename).
while ?line <> ?eof
   then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
   then do (page_advance).

env_page gets ' 2.5 Relationship of Proposed Experiment Type to Experi
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(\GARDEN\,?CURDIR,'\env2_5.DAT').
line is read_line (?filename).
while ?line <> ?eof
   then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
   then do (page_advance).

env_page gets ' 2.6 Anticipated Advance in State of the Art for This ?
   Experimentation'.
env_page gets ?blankline.
line_count = ?line_count + 3.
filename = concat(\GARDEN\,?CURDIR,'\env2_6.DAT').
line is read_line (?filename).
while ?line <> ?eof
   then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
do (page_advance).
end. (* 2.0 BACKGROUND *)
3.0 JUSTIFICATION FOR CONDUCTING THE EXPERIMENT IN SPACE

3.1 Limitations of Ground-Based Testing

3.2 Limitations of Drop Towers

3.3 Limitations of Testing in Aircraft

3.4 Need for Accommodations in the Shuttle
3.5 Limitations of Mathematical Modeling

line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,'\env3_5.DAT').
line is read line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

3.6 Limitations of Other Modeling Approaches

line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,'\env3_6.DAT').
line is read line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
do (page_advance).
end. (* 3.0 JUSTIFICATION FOR CONDUCTING THE EXPERIMENT IN SPACE *)

4.0 DESCRIPTION OF EXPERIMENT TYPES

page_count = 0.
filename = concat(C:\GARDEN\,?CURDIR,'\env4_1.DAT').
line is read line (?filename).
if ?line_count > 54
  then do (page_advance).

4.1 General Description of Type of Experiments

line_count = ?line_count + 6.
while ?line <> ?eof
  then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
  then do (page_advance).

4.2 Types of Experiment Procedures to be Used

line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,'\env4_2.DAT').
line is read line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
then do (page_advance).

env_page gets
' 4.3 Types of Measurements and Ranges of Values Required'.

env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,'\env4_3.DAT').
line is read line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
do (page_advance).
end. (* 4.0 DESCRIPTION OF EXPERIMENT TYPES *)

topic '5.0 Science Requirements Envelope'.
page_count = 0.
filename = concat(C:\GARDEN\,?CURDIR,'\env5_1.DAT').
line is read line (?filename).
if ?line_count > 56
  then do (page_advance).

env_page gets ?blankline.
env_page gets ?blankline.
env_page gets ' 5.0 SCIENCE REQUIREMENTS ENVELOPES'.
env_page gets ?blankline.
env_page gets ' 5.1 General Description of Experiment Sample Requirements'.
env_page gets ?blankline.
line_count = ?line_count + 6.
while ?line <> ?eof
  then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 54
  then do (page_advance).

env_page gets
' 5.2 Range of Atmospheric Requirements'.
env_page gets ?blankline.
env_page gets ' 5.2.1 Pressure'.
env_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\GARDEN\,?CURDIR,'\env5_2_1.DAT').
line is read line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
en
v_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

env_page gets
' 5.2.2 Gas Composition'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN, ?CURDIR,'\env5_2_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 5.2.3 Humidity'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN, ?CURDIR,'\env5_2_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 5.2.4 Vacuum'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN, ?CURDIR,'\env5_2_4.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 5.3 Temperature Control and Measurement'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN, ?CURDIR,'\env5_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 5.4 Vibration Control and Measurement'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN, ?CURDIR,'\env5_4.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

env_page gets
  5.5 Test Matrices'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN, ?CURDIR,'\env5_5-DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 54
  then do (page_advance).

env_page gets
   5.6 Imaging Requirements Envelope'.
env_page gets ?blankline.
env_page gets
      5.6.1 Photography'.
env_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\GARDEN, ?CURDIR,'\env5_6_1-DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

env_page gets
  5.6.2 Radiography'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN, ?CURDIR,'\env5_6_2-DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

env_page gets
  5.6.3 Television'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN, ?CURDIR,'\env5_6_3-DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = line_count + 1.
env_page gets ?blankline.
if line_count > 56
    then do (page_advance).

env_page gets
  ' 5.6.4 Resolution'.
env_page gets ?blankline.
line_count = line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_6_4.DAT')
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = line_count + 1.
env_page gets ?blankline.
if line_count > 56
    then do (page_advance).

env_page gets
  ' 5.6.5 Frame Rate'.
env_page gets ?blankline.
line_count = line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_6_5.DAT')
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = line_count + 1.
env_page gets ?blankline.
if line_count > 56
    then do (page_advance).

env_page gets
  ' 5.7 Electromagnetic Limitations'.
env_page gets ?blankline.
line_count = line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_7.DAT')
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = line_count + 1.
env_page gets ?blankline.
if line_count > 54
    then do (page_advance).

env_page gets
  ' 5.8 Astronaut Involvement'.
env_page gets ?blankline.
env_page gets
    ' 5.8.1 Extravehicular Activity'.
env_page gets ?blankline.
line_count = line_count + 4.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_8_1.DAT')
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).
  
env_page gets ' 5.8.2 Activation of Experiment'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN,?CURDIR,\'\env5_8_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).
  
env_page gets ' 5.9 Data Requirements'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN,?CURDIR,\'\env5_9.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

env_page gets ' 5.10 Telepresence and Telerobotics'.
env_page gets ?blankline.
env_page gets ' 5.10.1 Telepresence'.
env_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\GARDEN,?CURDIR,\'\env5_10-1.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

env_page gets ' 5.10.2 Telerobotics'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN,?CURDIR,\'\env5_10_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
do (page_advance).
end. (* 5.0 SCIENCE REQUIREMENTS ENVELOPE *)

topic '6.0 OTHER REQUIREMENTS'.
page_count = 0.
filename = concat(C:\GARDEN, CURDIR,'\env6_I.DAT').
line is read line (?filename).
if ?line_count > 54
  then do (page_advance).
    env_page gets ?blankline.
    env_page gets ?blankline.
    env_page gets ' 6.0 OTHER REQUIREMENTS'.
    env_page gets ?blankline.
    line_count = ?line_count + 4.
    while ?line <> ?eof
      then do (read_file).
    close (?filename).
    env_page gets ?blankline.
    line_count = ?line_count + 1.
    do (page_advance).
end. (* 6.0 OTHER REQUIREMENTS *)

topic 'page_advance'.
page_count = ?page_count + 1.
window_name = concat(?choice,' Page: ',?page_count).
window (?window_name,white,blue,white,3,3,78,[9).
say (?env_page).
env_page = [].
env_page gets ?blankline.
env_page gets ?blankline.
line_count = 2.
collect ()
close_window ()
end. (* page_advance *)

topic 'read_file'.
if ?line_count > 58
  then do (page_advance).
    line = concat(' ',?line).
    env_page gets ?line.
    line is read line (?filename).
    line_count = ?line_count + 1.
end. (* read_file *)
end. (* display_Routine *)
(* ENVPRINT.KB is used to print all sections of the Envelope Document. *)

no_edit_key ().
no_debug ().
eof = number_to_char (26).
curdir is read_line ('CURDIR.DAT').
curdir is string_replace(?curdir, ',',' ',8).
orig_curdir = ?curdir.
blankline = '.
choice = '.
line_count = 0.
page_count = 0.
base_dir = [].
curbase is read_line (concat('C:\GARDEN\',?CURDIR,'\BASELINE.DAT')).
close (concat('C:\GARDEN\',?CURDIR,'\BASELINE.DAT')).

if ?curbase <> ?eof
  then
    base_choice = ['BASELINE','CURRENT REVISION','NEITHER']
    and
    ask ('#e

        Do you wish to print documents from the baseline or from the current revision?',base_dir,?base_choice)

and

if ?base_dir = BASELINE
  then
    curdir = concat (?curdir, '\BASELINE')
    and
    curdir = string_replace (?curdir, ' ', ' ',8)
    and
    new_file ('C:\GARDEN\CURDIR.DAT')
    and
    write ('C:\GARDEN\CURDIR.DAT', ?curdir)
    and
    do (print_sections).

if ?base_dir <> NEITHER
  then
    while ?choice <> QUIT
      then
        do (print_sections).

new_kb ('NASA_ENV.HKB').
topic 'Print sections'.
env_page = [].
sections = ['1.0 INTRODUCTION/SUMMARY','2.0 BACKGROUND',
'3.0 JUSTIFICATION FOR CONDUCTING THIS EXPERIMENT IN SPACE',
'4.0 DESCRIPTION OF EXPERIMENT TYPES','5.0 SCIENCE REQUIREMENTS ENVELOPE',
'6.0 OTHER REQUIREMENTS','PRINT ENTIRE DOCUMENT','QUIT'].

window ('Print Science Requirements Envelope',white,blue,white,3,3,78,18).
ask ('#e

    Which section do you wish to Print?',choice,?sections).

if ?choice = '1.0 INTRODUCTION/SUMMARY'
  then
    do ('1.0 INTRODUCTION/SUMMARY')
    and
    if ?line_count > 2
then
do (page_advance).

if ?choice = '2.0 BACKGROUND'
then do ('2.0')
and
if ?line_count > 2
then
do (page_advance).

if ?choice = '3.0 JUSTIFICATION FOR CONDUCTING THIS EXPERIMENT IN SPACE'
then do ('3.0')
and
if ?line_count > 2
then do (page_advance).

if ?choice = '4.0 DESCRIPTION OF EXPERIMENT TYPES'
then do ('4.0')
and
if ?line_count > 2
then do (page_advance).

if ?choice = '5.0 SCIENCE REQUIREMENTS ENVELOPE'
then do ('5.0')
and
if ?line_count > 2
then do (page_advance).

if ?choice = '6.0 OTHER REQUIREMENTS'
then do ('6.0')
and
if ?line_count > 2
then do (page_advance).

if ?choice = 'PRINT ENTIRE DOCUMENT'
then
load ('envtitle.hkb')
and
do ('print_title_page')
and
remove_topic ('print_title_page')
and
page_count = 0
and
window (,white,red,yellow,1,16,27,4)
and
write ('con:','PRINT IN PROGRESS...')
and
do ('1.0 INTRODUCTION/SUMMARY')
and
do ('2.0')
and
do ('3.0')
and
do ('4.0')
and
do ('5.0')
and
do ('6.0')
and print (#p,' ',#p).
close_window().

if ?choice = 'QUIT'
then
  if ?base_dir = BASELINE
  then
    curdir = ?orig_curdir
    and
    curdir = string_replace (?curdir,' ','',8)
    and
    new_file ('C:\GARDEN\CURDIR.DAT')
    and
    write ('C:\GARDEN\CURDIR.DAT',?curdir)
    and
    new_kb ('nasa_env.hkb')
  else
    new_kb ('nasa_env.hkb').
end. (* print sections *)

topic '1.0 INTRODUCTION/SUMMARY'.
  window (,white,red,yellow,1,16,27,4)
  and
  write ('con:','PRINT IN PROGRESS...')
  and
  env_page gets ?blankline.
  env_page gets ?blankline.
  filename = concat(C:\GARDEN\,?CURDIR,'\env1_1.DAT').
  line is read_line (?filename).
  env_page gets ' 1.0 INTRODUCTION/SUMMARY'.
  env_page gets ?blankline.
  env_page gets ' 1.1 Description of Experiment Type or Class'.
  env_page gets ?blankline.
  line_count = 6.
  while ?line <> ?eof
    then do (read_file).
  close (?filename).
  env_page gets ?blankline.
  line_count = ?line_count + 1.
  if ?line_count > 56
    then do (page_advance).

  env_page gets ' 1.2 Scientific Knowledge to be Gained From This Type of...
  env_page gets ?blankline.
  line_count = ?line_count + 2.
  filename = concat(C:\GARDEN\,?CURDIR,'\env1_2.DAT').
  line is read_line (?filename).
  while ?line <> ?eof
    then do (read_file).
  close (?filename).
  env_page gets ?blankline.
  line_count = ?line_count + 1.
  if ?line_count > 56
    then do (page_advance).

1.3 Value of Knowledge of This Type of Experiment

line_count = line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,\'\env1_3.DAT\').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = line_count + 1.
if ?line_count > 56
    then do (page_advance).

1.4 Necessity for Space Environment to Experiment Type

filename = concat(C:\GARDEN\,?CURDIR,\'\env1_4.DAT\').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = line_count + 1.
if ?line_count > 56
    then do (page_advance).
end. (* 1.0 *)

2.0 Background

filename = concat(C:\GARDEN\,?CURDIR,\'\env2_1.DAT\').
line is read_line (?filename).
if ?line_count > 56
    then do (page_advance).

2.1 Scientific Field to which Experiment Belongs

env_page gets ?blankline.
env_page gets ?blankline.
env_page gets ' 2.1 Scientific Field to which Experiment Belongs'.
env_page gets ?blankline.
env_page gets ?blankline.
line_count = line_count + 6.
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = line_count + 1.
if ?line_count > 56
    then do (page_advance).

2.2 Current Application for Research in the Field

filename = concat(C:\GARDEN\,?CURDIR,\'\env2_1.DAT\').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
  then do (page_advance).

env_page gets ' 2.3 Brief Historical Account of Prior Research'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:|GARDEN|\,?CURDIR,'\env2_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

env_page gets ' 2.4 Current Research'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:|GARDEN|\,?CURDIR,'\env2_4.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).

env_page gets ' 2.5 Relationship of Proposed Experiment Type to Experiments'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:|GARDEN|\,?CURDIR,'\env2_5.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (pageAdvance).

env_page gets ' 2.6 Anticipated Advance in State of the Art'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:|GARDEN|\,?CURDIR,'\env2_6.DAT').
line is read_line (?filename).
while ?line <> ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (pageAdvance).
end. (* 2.0 *)
topic '3.0'.
    filename = concat(C:\GARDEN\, ?CURDIR, '\env3_1.DAT').
    line is read_line (?filename).
    if ?line_count > 56
        then do (page_advance).

    window (, white, red, yellow, 1, 16, 27, 4)
    and
    write ('con:', 'PRINT IN PROGRESS...')
    and
    env_page gets ?blankline.
    env_page gets ?blankline.
    env_page gets ' 3.0 JUSTIFICATION FOR CONDUCTING THIS EXPERIMENT IN SPACE'.
    env_page gets ?blankline.
    env_page gets ' 3.1 Limitations of Ground-Based Testing'.
    line_count = ?line_count + 6.
    while ?line <> ?eof
        then do (read_file).
    close (?filename).
    env_page gets ?blankline.
    line_count = ?line_count + 1.
    if ?line_count > 56
        then do (page_advance).

    env_page gets ' 3.2 Limitations of Drop Towers'.
    env_page gets ?blankline.
    line_count = ?line_count + 2.
    filename = concat(C:\GARDEN\, ?CURDIR, '\env3_2.DAT').
    line is read_line (?filename).
    while ?line <> ?eof
        then do (read_file).
    close (?filename).
    line_count = ?line_count + 1.
    env_page gets ?blankline.
    if ?line_count > 56
        then do (page_advance).

    env_page gets ' 3.3 Limitations of Testing in Aircraft'.
    env_page gets ?blankline.
    line_count = ?line_count + 2.
    filename = concat(C:\GARDEN\, ?CURDIR, '\env3_3.DAT').
    line is read_line (?filename).
    while ?line <> ?eof
        then do (read_file).
    close (?filename).
    line_count = ?line_count + 1.
    env_page gets ?blankline.
    if ?line_count > 56
        then do (page_advance).

    env_page gets ' 3.4 Need for Accommodations in the Shuttle'.
    env_page gets ?blankline.
    line_count = ?line_count + 2.
    filename = concat(C:\GARDEN\, ?CURDIR, '\env3_4.DAT').
    line is read_line (?filename).
    while ?line <> ?eof
        then do (read_file).
    close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 3.5 Limitations of Mathematical Modeling'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,\'\env3_5.DAT\').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 3.6 Limitations of Other Modeling Approaches'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,\'\env3_6.DAT\').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).
end. (* 3.0 *)

topic '4.0'.
filename = concat(C:\GARDEN\,?CURDIR,\'\env4_1.DAT\').
line is read_line (?filename).
if ?line_count > 56
    then do (page_advance).

window ,(white,red,yellow,1,16,27,4)
and
write ('con:','PRINT IN PROGRESS...')
and
env_page gets ?blankline.

env_page gets ?blankline.
env_page gets ' 4.0 DESCRIPTION OF EXPERIMENT TYPES'.
env_page gets ?blankline.
env_page gets ' 4.1 General Description of Type of Experiments'.
env_page gets ?blankline.
line_count = ?line_count + 6.
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 4.2 Types of Experiment Procedures to be Used'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,'\env4_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 4.3 Types of Measurements and Ranges of Values Required'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,'\env4_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).
end. (* 4.0 *)

topic '5.0'.
filename = concat(C:\GARDEN\,?CURDIR,'\env5_1.DAT').
line is read_line (?filename).
if ?line_count > 56
    then do (page_advance).

window (,white,red,yellow,1,16,27,4)
and
write ('con:','PRINT IN PROGRESS...')
and
env_page gets ?blankline.
env_page gets ?blankline.
env_page gets ' 5.0 SCIENCE REQUIREMENTS ENVELOPE'.
env_page gets ?blankline.
env_page gets ' 5.1 Experiment Sample Requirements'.
env_page gets ?blankline.
line_count = ?line_count + 6.
while ?line <> ?eof
    then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 5.2 Atmospheric Requirements'.
env_page gets ?blankline.
env_page gets ' 5.2.1 Pressure'.
env_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\GARDEN\,?CURDIR,'\env5_2_1.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 5.2.2 Gas Composition'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,\'\env5_2_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 5.2.3 Humidity'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,\'\env5_2_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 5.2.4 Vacuum'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,\'\env5_2_4.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets ' 5.3 Temperature Control and Measurement'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\,?CURDIR,\'\env5_3.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets
'  5.4 Vibration Control and Measurement'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN,?CURDIR,'\env5_4.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets
'  5.5 Test Matrices'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN,?CURDIR,'\env5_5.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets
'  5.6 Imaging Requirements'.
env_page gets ?blankline.
env_page gets
'  5.6.1 Photography'.
env_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\GARDEN,?CURDIR,'\env5_6_1.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets
'  5.6.2 Radiography'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN,?CURDIR,'\env5_6_2.DAT').
line is read_line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets
  ' 5.6.3 Television'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_6_3.DAT').
line is read line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets
  ' 5.6.4 Resolution'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_6_4.DAT').
line is read line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets
  ' 5.6.5 Frame Rate'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_6_5.DAT').
line is read line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).

env_page gets
  ' 5.7 Electromagnetic Limitations'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN\, ?CURDIR, '\env5_7.DAT').
line is read line (?filename).
while ?line <> ?eof
    then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
    then do (page_advance).
5.8 Astronaut Involvement.

5.8.1 Extravehicular Activity.

5.8.2 Activation of Experiment.

5.9 Data Requirements.

5.10 Telepresence and Telerobotics.

5.10.1 Telepresence.
then do (page_advance).

env_page gets
'
  5.10.2 Telerobotics'.
env_page gets ?blankline.
line_count = ?line_count + 2.
filename = concat(C:\GARDEN, ?,CURDIR, '\en5_10_2.DAT').
line is read_line (?filename).
while ?line < > ?eof
  then do (read_file).
close (?filename).
line_count = ?line_count + 1.
env_page gets ?blankline.
if ?line_count > 56
  then do (page_advance).
end. (* 5.0 *)

topic '6.0'.
window (,white,red,yellow,1,16,27,4)
and
write ('con:','PRINT IN PROGRESS...')
and
env_page gets ?blankline.
env_page gets ?blankline.
env_page gets ' 6.0 OTHER REQUIREMENTS'.
env_page gets ?blankline.
line_count = ?line_count + 4.
filename = concat(C:\GARDEN, ?,CURDIR, '\ENV6_1.DAT').
line is read_line (?filename).
while ?line < > ?eof
  then do (read_file).
close (?filename).
env_page gets ?blankline.
line_count = ?line_count + 1.
do (page_advance).
end. (* 6.0 *)

topic 'page advance'.
page_count = ?page_count + 1.
page_line = concat(''
while ?line_count < 58
  then
    env_page gets ?blankline
    and
    line_count = ?line_count + 1.
if ?choice = 'PRINT ENTIRE DOCUMENT'
  then
    env_page gets ?page_line.
    print (#p, ?env_page).
    env_page = [].
    env_page gets ?blankline.
    env_page gets ?blankline.
    line_count = 2.
collect ().
end. (* page advance *)

topic 'read file'.
if ?line_count > 56
  then do (page_advance).

line = concat(′   ′,?line).
env_page gets ?line.
line is read_line (?filename).
line_count = ?line_count + 1.
end.  (* read_file *)

end.  (* Print_Routine *)
(* ENVTITLE.KB prints the Title Page for the Science Requirements *)
(* Envelope Document. *)

no_edit_key ().
no_debug ().

topic 'print_title_page'.
no_edit_key ().
filler = '
.eof = number_to_char (26).
blankline = '

expname is read(concat(C:\GARDEN\,?CURDIR,'\TITLE.DAT')).

expname is string_replace(?expname, '','','',600).
expname is remove(?expname,'').
lines = list_length(?expname).
line_count = 1.
title_length = 0.

window (,white,red,yellow,1,16,27,4).
write ('con:','REPORT BEING PRINTED...').

while ?line_count <= ?lines
then
  cur_line = element(?expname,?line_count)
  and
cur_length = string_length(?cur_line)
  and
line_count = ?line_count + 1
  and
title_length = ?title_length + ?cur_length + 1
  and
  if ?cur_line = ''
  then
    expname = string_replace(?expname,?cur_line).
expname = concat(element(?expname,1),',','
  element(?expname,2),','
  element(?expname,3),','
  element(?expname,4),','
  element(?expname,5),','
  element(?expname,6),','
  element(?expname,7),','
  element(?expname,8),','
  element(?expname,9),','
  element(?expname,10),','
  element(?expname,11),','
  element(?expname,12),','
  element(?expname,13),'
  element(?expname,14),',
  element(?expname,15)).

title_length = ?title_length + 3.
counter = 1.
write_counter = 1.
line_one = ' '.
while ?counter < ?title_length
then
  char = string_copy(?expname,?counter,1)
  and
counter = ?counter + 1
and
write_counter = ?write_counter + 1
and
if ?write_counter > 40 and ?char = ' ' then
  linelgth = string_length(?line_one)
  and
  linelgth = 80 - ?linelgth
  and
  linelgth = ?linelgth / 2
  and
  linefiller = string_copy (?blankline,1,?linelgth)
  and
  line_one = concat(?linefiller,?line_one,'#n',?char)
  and
  write_counter = 1
  and
  new_expname gets ?line_one
  and
  line_one = ''
else
  line_one = concat(?line_one,?char).
linelgth = string_length(?line_one).
linelgth = 80 - ?linelgth.
linelgth = ?linelgth / 2.
linefiller = string_copy (?blankline,1,?linelgth).
line_one = concat(?linefiller,?line_one,'#n').
new_expname gets ?line_one.
new_expname = string_replace(?new_expname,'-',' ',600).
authorfile = concat(C:\GARDEN\?,CURDIR,'\AUTHOR.DAT').
titlepage = ('

Science Requirements Envelope Document'.
  for:

.titlepage gets ?new_expname.
titlepage gets '#n'.
titlepage gets '#n'.
pagedate = ?date.
mo_num = element(?pagedate,1).
if ?mo_num = 1
  then month = January
else
  if ?mo_num = 2
    then month = February
else
  if ?mo_num = 3
    then month = March
else
  if ?mo_num = 4
    then month = April
else
  if ?mo_num = 5
    then month = May
else
  if ?mo_num = 6
    then month = June
else
if ?mo_num = 7
then month = July
else
if ?mo_num = 8
then month = August
else
if ?mo_num = 9
then month = September
else
if ?mo_num = 10
then month = October
else
if ?mo_num = 11
then month = November
else
if ?mo_num = 12
then month = December
else
month = ' '.

year = element(?pagedate,3).
reportdate = concat(?month,' ',?year).

datelgth = string_length(?reportdate).
datelgth = 80 - ?datelgth.
datelgth = ?datelgth / 2.
datefiller = string_copy (?blankline,1,?datelgth).
titlepage gets ' '.
titlepage gets ' '.
titlepage gets concat(?datefiller,?reportdate).

basefile = concat ('C:\GARDEN\',?CURDIR,'\BASELINE.DAT').
baseline = read(?basefile).
if ?baseline = ?eof
then
    baseline = 'N/A'.

datelgth = string_length (?baseline).
datelgth = 80 - ?datelgth.
datelgth = ?datelgth / 2.
datefiller = string_copy (?blankline,1,?datelgth).
titlepage gets ' '.
titlepage gets ' '.
titlepage gets concat(?datefiller,?baseline).

name is read line (?authorfile).
name is string_replace(?name,'NAME: ',',',1).
name is string_replace(?name,' ',',',8).
namelgth = string_length(?name).
namelgth = 80 - ?namelgth.
namelgth = ?namelgth / 2.
namefiller = string_copy (?blankline,1,?namelgth).
titlepage gets concat('#n#n#n#n#n#n#n#n#n#n#n#n',?namefiller,?name).

organization is read line (?authorfile).
organization is string_replace(?organization,'ORGANIZATION: ',',',1).
organization is string_replace(?organization,' ',',',8).
organizationlngth = string_length(?organization).
organizationlngth = 80 - ?organizationlngth.
organizationlngth = ?organizationlngth / 2.
organizationfiller = string_copy (?blankline,1,?organizationLength).
titlepage gets concat(?organizationfiller,?organization).

mcode is read_line (?authorfile).
mcode is string_replace(?mcode,'MAIL CODE: ',''',1).
mcode is string_replace(?mcode,' ''',8).
mcodeLength = string_length(?mcode).
mcodeLength = 80 - ?mcodeLength.
mcodeLength = ?mcodeLength / 2.
mcodefiller = string_copy (?blankline,1,?mcodeLength).
titlepage gets concat(?mcodefiller,?mcode).

street is read_line (?authorfile).
street is string_replace(?street,'STREET: ',''',1).
street is string_replace(?street,' ''',8).
streetLength = string_length(?street).
streetLength = 80 - ?streetLength.
streetLength = ?streetLength / 2.
streetfiller = string_copy (?blankline,1,?streetLength).
titlepage gets concat(?streetfiller,?street).

city is read_line (?authorfile).
city is string_replace(?city,'CITY, STATE, ZIP: ',''',1).
city is string_replace(?city,' ''',8).
city is string_replace(?city,' ''',8).
cityLength = string_length(?city).
cityLength = 80 - ?cityLength.
cityLength = ?cityLength / 2.
cityfiller = string_copy (?blankline,1,?cityLength).
titlepage gets concat(?cityfiller,?city).

phone is read_line (?authorfile).
phone is string_replace(?phone,'PHONE: ',''',1).
phone is string_replace(?phone,' ''',8).
phonelength = string_length(?phone).
phonelength = 80 - ?phonelength.
phonelength = ?phonelength / 2.
phonefiller = string_copy (?blankline,1,?phonelength).
titlepage gets concat(?phonefiller,?phone).

print (#p,?titlepage,#p).
close_window ().
end. (* print_title_page *)
This is the menu provided to the user to determine what he/she is to do on a project.

no_edit_key ().
noc_debug ().
fdaction = ' '.

curdir is read_line ('CURDIR.DAT').
curdir is string_replace (?curdir,' ','','','',8).

while ?fdaction <> 'Exit System'
    then do (filldoc).

  topic 'filldoc'.

  fdchoices = ['Enter Project Initialization Information',
               'Complete Science Requirements Envelope Document',
               'Mark Answers in Envelope as "Not Applicable"',
               'Return to Previous Menu',
               'Return to Main Menu',
               'Exit System'].

  window (,white,red,yellow,5,5,75,16).
    set_number_of_values (fdaction,1).

    ask ([['#e #s
        Please select the activity you wish to perform on #n
        the',?curdir,'#dproject.'],fdaction,?fdchoices]).

  close_window ().

  if ?fdaction = 'Enter Project Initialization Information'
      then new_kb ('einitial.hkb').

  if ?fdaction = 'Complete Science Requirements Envelope Document'
      then new_kb ('fillev.hkb').

  if ?fdaction = 'Mark Answers in Envelope as "Not Applicable"'
      then new_kb ('envelim.hkb').

  if ?fdaction = 'Return to Main Menu'
      then new_kb ('ENVMMENU.ckb').

  if ?fdaction = 'Return to Previous Menu'
      then new_kb ('nasa_env.hkb').

  if ?fdaction = 'Exit System'
      then exit ().
This program is used to allow the user to enter standard project initialization information, i.e. name, address, title, etc.

no_edit_key ().
no_debug ().
curdir is read_line ('CURDIR.DAT').
curdir = string_replace(?curdir,' ','',8).

eof = number_to_char (26).
yn is [YES,NO].
chgwant = ' '.
do (personal_info).
new_kb ('envfmenu.hkb').

topic 'personal_info'.

    blankline = ' '.
close_window ().
oldtext is read (concat (C:\GARDEN\,?CURDIR,\'\AUTHOR.DAT')).
if ?oldtext = ?eof
    then do (new_personal)
else
    chgwant = ' ' and
    while ?chgwant <> QUIT
        then do (edit_personal).

(* ------------------------ get new personal information ----------------------*)

topic 'new_personal'.
WRITE ('Con:', '#eIn the window below, please provide some general information about yourself and your experiment.').

window (,white,red,yellow,5,5,75,16).

read_response ('#e
    #fyellow Please enter your first and last names, i.e. Dr. John Doe.#d
    #n ',name,?
    blankline).
name = concat ('NAME: ',?name).
personal gets ?name.

read_response ('#e
    #fyellow Please enter the name of your organization.#d
    #n ',organization,?
    blankline).
organization = concat ('ORGANIZATION: ',?organization).
personal gets ?organization.

read_response ('#e
    #fyellow Please enter the mail code, P.O Box, room number, or other needed address information of your organization.#d
    #n ',mail_code,?
    blankline).
mail_code = concat ('MAIL CODE: ',?mail_code).
personal gets ?mail_code.
read_response ("Please enter the street address of your organization.", street, ?blankline).

street = concat ("STREET:", ?street).
personal gets ?street.

read_response ("Please enter the city, state, and zip code of your organization.", city_st_zip, ?blankline).

city_st_zip = concat ("CITY, STATE, ZIP: ", city_st_zip).
city_st_zip = string_replace (?city_st_zip, ' ', 'T', 8).
personal gets city_st_zip.

read_response ("Please enter your phone number.", phone, ?blankline).

phone = concat ("PHONE: ", phone).
personal gets ?phone.

say ("
Please enter the title of your experiment.

Press the RETURN KEY to enter the editor, ESC to leave the editor, and RETURN to confirm save.")

edit_file (concat (C:\GARDEN\?CURDIR\'\TITLE.DAT'), yellow, black, red, 9, 11, 70, ?author_info is [?name, ?organization, ?mail_code, ?street, ?city_st_zip, ?phone, ?n].

new_file (concat (C:\GARDEN\?CURDIR\'\AUTHOR.DAT')).

write (concat (C:\GARDEN\?CURDIR\'\AUTHOR.DAT'), ?author_info, ?n).

if ?chgwant <> QUIT
then
  ask
  ([?author_info, 'n' yellow
    The information listed has been written to a file.
    Do you wish to change any of these entries?', change_again, ?yn, 60, 16)
  and
  if ?change_again = YES
  then
    oldtext is read (concat (C:\GARDEN\?CURDIR\'\AUTHOR.DAT'))
    and
    change_again = NO
    and
    chgwant = ' '
    and
    while ?chgwant <> QUIT
    then
      do (edit_personal).

close_window ().
WRITE ('con:', '#e ', ' ').
end. (* new_personal *)

(* ---------------- get corrected personal information ----------------*)

topic 'edit_personal'.

change_info is [NAME, ORGANIZATION, 'MAIL CODE', STREET, 'CITY STATE ZIP', PHONE, TITLE, QUIT].

ask ('Which entry in the below list do you wish to change?', chgwant, ?change_info).

if ?chgwant = NAME
then
    old_name = element(?oldtext, 1)
    and
    old_value = string_replace (?old_name, 'NAME: ', '', 1)
    and
    read_response ([#e #fyellow Your original entry for name was#s ', ?old_value, '.#d #s #n#n Please enter the corrected name in its entirety.#n'], new_name, ?old_value)

if ?chgwant = ORGANIZATION
then
    old_org = element(?oldtext, 2)
    and
    old_value = string_replace (?old_org, 'ORGANIZATION: ', '', 1)
    and
    read_response ([#e #fyellow Your original entry for organization was#s #n', ?old_value, '.#d #s #n#n Please enter the corrected organization in its entirety.#n'], new_org, ?old_value)

if ?chgwant = 'MAIL CODE'
then
    old_mc = element(?oldtext, 3)
    and
    old_value = string_replace (?old_mc, 'MAIL CODE: ', '', 1)
    and
    read_response ([#e #fyellow Your original entry for mail code was#s #n', ?old_value, '.#d #s #n#n Please enter the corrected mail code in its entirety.#n'], new_mc, ?old_value).
if ?chgwant = 'MAIL CODE'
    then
        new_mc = concat ('MAIL CODE: ', ?new_mc)
        and
        oldtext is replace(?oldtext, ?old_mc, ?new_mc).

if ?chgwant = STREET
    then
        old_street = element(?oldtext, 4)
        and
        old_value = string_replace (?old_street, 'STREET: ', '', 1)
        and
        read_response (["#e #yellow Your original entry for street was#s #n", ?old_value,
        Please enter the corrected street address in its entirety.#n"], new_street, ?old_value).

if ?chgwant = STREET
    then
        new_street = concat ('STREET: ', ?new_street)
        and
        oldtext is replace(?oldtext, ?old_street, ?new_street).

if ?chgwant = 'CITY STATE ZIP'
    then
        old_city = element(?oldtext, 5)
        and
        old_value = string_replace (?old_city, 'CITY, STATE, ZIP: ', '', 1)
        and
        read_response (["#e #yellow Your original entry for city, state and zip was#s #n", ?old_value,
        Please enter the corrected city, state, and zip address in its entirety.#n"], new_city, ?old_value).

if ?chgwant = 'CITY STATE ZIP'
    then
        new_city = concat ('CITY, STATE, ZIP: ', ?new_city)
        and
        new_city = string_replace (?new_city, ' ', '8')
        and
        oldtext is replace(?oldtext, ?old_city, ?new_city).

if ?chgwant = PHONE
    then
        old_phone = element(?oldtext, 6)
        and
        old_value = string_replace (?old_phone, 'PHONE: ', '', 1)
        and
        read_response (["#e #yellow Your original entry for phone was#s #n", ?old_value,
        Please enter the corrected phone number in its entirety.#n"], new_phone, ?old_value).

if ?chgwant = PHONE
    then
        new_phone = concat ('PHONE: ', ?new_phone)
        and
        oldtext is replace(?oldtext, ?old_phone, ?new_phone).
if ?chgwant = 'TITLE'
    then
        say ('Please enter the corrected title of your experiment.

        Press the RETURN KEY to enter the editor, ESC to leave the editor, and RETURN to confirm save.'
        and
        edit_file (concat (C:\GARDEN\, ?CURDIR,'\TITLE.DAT'),yellow,black,red,5,9,70

if ?chgwant = QUIT
    then
        new_file (concat (C:\GARDEN\, ?CURDIR,'\AUTHOR.DAT'))
        and
        write (concat (C:\GARDEN\, ?CURDIR,'\AUTHOR.DAT'), ?oldtext, #n)
        and
        close (concat (C:\GARDEN\, ?CURDIR,'\AUTHOR.DAT')).

if ?chgwant = QUIT
    then
        say (['?oldtext,' 'The information listed has been written to a file. Please press SPACE to continue.
        ']).

end. (* edit_personal *)

end. (* personal_info *)

(* ============== end personal information ==============*)
no_edit_key().
no_debug().
do (nasasys).
topic nasasys.

column = 3.
row = 3.
ckount = [].

curdir is read_line('CURDIR.DAT').
curdir = string_replace(?curdir,' ','',','8).

yn is [YES,NO].
chgwant = ' '.
tried = 0.
type_change = ' '.
change_type = ' '.

ask('Have you already begun to fill out the Science Requirements Envelope Document in a previous session?',begun,?yn)
and
if ?begun = NO
then
    new_one = ' ' 
and 
do (fillENV)
and 
    new_kb (ENVFMENU.hkb')
else
    type_change is ['CHANGE ONE ITEM','CHANGE AND CONTINUE','COMPLETE UNANSWERED TOPICS',QUIT]
and

while ?change_type <> QUIT
then ask('Do you wish to change only one item, resume at a point and continue sequentially through the remainder of the Envelope Document, or complete all topics previously unanswered?',change_type,?type_change)
and
if ?change_type = 'CHANGE ONE ITEM'
then
    WRITE ('con:','#e ') 
and 
    window (,white,red,yellow,1,16,27,4)
and 
    WRITE ('con:',
'One moment please...
'),
and 
do (ENV_start)
else
    if ?change_type = 'COMPLETE UNANSWERED TOPICS'
then
    WRITE ('con:', '#e ')
    and
    window ((white, red, yellow, 1, 15, 32, 5)
    and
    do (ENV_complete)
else
    if ?change_type = 'CHANGE AND CONTINUE'
    then
        WRITE ('con:', '#e ')
        and
        window ((white, red, yellow, 1, 16, 27, 4)
        and
        WRITE ('con-
        "One moment please..."
        )
        and
        resume = 1
        and
        choice = ' '.
        do (ENV_continue)
        and
        new_kb ('envfmenu.hkb')
    else
        new_kb ('envfmenu.hkb').

    if ?change_type = QUIT
        then new_kb ('envfmenu.hkb').

topic 'ENV_start'.
    resume = 1.
    choice = ' '.
    while ?choice <> Quit
        then do (ENV_begin).
    close_window ()
    WRITE ('con:', '#e ').
    reset (ENV_begin).
    collect ()
end. (* ENV_start *)

(* ************************** BEGIN FILLING OUT ENVELOPE ************************** *)

topic 'fillENV'.
    if ?new_one = ' ':
        then Load ('envquest.hkb')
        and fs gets children(fillENV)
        and fs is remove (?fs, 'related answer')
        and stopENV = 'N'
        and new_one = 'X'
        and x = 1.
    y = (element(?fs, ?x)).
    if ?y <> ' ' and ?stopENV <> 'Y'
        then do (?y).
        x = ?x + 1.
    if ?x = 8
or \( x = 16 \)
or \( x = 24 \)
or \( x = 32 \)
then collect ().

WRITE ('con:','\#e ').

continuex = NO.
ask ('\#e

Do you wish to continue with the next question?',continuex,?yn).

if ?continuex = NO
then
stopENV = 'Y'.

if ?y = last(?fs)
then stopENV = 'Y'.

if ?stopENV <> 'Y'
then do (fillENV).

end. ( * fillENV *)

(* ******************************************************************************* *)

topic 'ENV_begin'.
if ?resume = 1
then load ('envquest.hkb').
close_window ().

if ?resume = 1
then begin is 'Quit'.
if ?resume = 1
then begin gets children(ENV_begin).
if ?resume = 1
then begin gets 'Quit'
and
begin is remove (?begin,'related answer')
and
choice = ' '.

while ?choice <> Quit
then
window ('Change One Item',blue,white,white,?column,?row,)
and
ask ('Which subtopic do you wish to change?',choice, ?begin)
and
close_window ()
and
if ?choice <> 'Quit'
then do (?choice)
and
resume = 2
and
close_window ().
if ?choice = 'Quit'
then
close_window ()
and
WRITE ('con:','e   ') and
window (,white,red,yellow,1,16,27,4) and
WRITE ('con:',
'A slight delay will occur
while the next segment of
this application is loaded.
Please stand by.  (') and
stop_at = where (?begin,Quit,2) and
kounter = 1 and
stop_at = remove(?stop_at,1) and
while ?kounter < ?stop_at then
  eraser = element (?begin,?kounter) and
  remove_topic (?eraser) and
  kounter = ?kounter + 1.
end. (* ENV_begin *)
(* **************************** RESUME FILLING OUT ENVELOPE **************************** *)
topic 'ENV_continue'.
  choice = '  '.
   if ?resume = 1 then load ('envquest.hkb').
   close_window ().
   window ('Change and Continue',blue,white,white,?column,?row).
   if ?resume = 1 then
     continue is children (ENV_continue).
   if ?resume = 1 then
     continue gets 'Quit'
     and
     continue is remove (?continue,'related answer').
   ask ('With which subtopic do you wish to resume your activity? ',
    choice, ?continue).
   if ?choice <> 'Quit'
     then
     cont_where = where(?continue,?choice) and
     ckount = 1 and
     while ?ckount < ?cont_where then
       rem_top = element(?continue,?ckount)
       and
       remove_topic (?rem_top)
       and
       ckount = ?ckount + 1.
     collect ().
   while ?choice <> 'Quit'
then choice = element(?continue,?ckount) and
if ?choice <> ' ' and ?choice <> 'Quit'
then
  do (?choice)
  and
  ckount = ?ckount + 1
  and
  remove_topic (?choice)
  and
  continuey = NO
  and
  ask ('#e
Do you wish to continue with the next question?',continuey,?yn)
  and
  if ?continuey = NO
    then
      choice = 'Quit'.

if ?ckount = 8
  or ?ckount = 16
  or ?ckount = 24
  or ?ckount = 32
  then collect ().
  resume = 2.
  close_window ().
  WRITE ('con:','#e')
end. (* ENV_continue *)

(* ***********************************************************************)
topic 'ENV_complete'.
say ('#e
This selection will invoke
a DOS command, which will
cause your screen to blank
out momentarily. Do not
be alarmed. Press #fyellow SPACE#d now. ').
  close (concat (C:\GARDEN\,?curdir,'\ELISTING.OUT')).
  dos (esearch,restore).
  eof = number_to_char (26).

  WRITE ('con:','#eOne moment please...'
 ).
  searchtext is read_line (concat (C:\GARDEN\,?CURDIR,'\ELISTING.OUT')).
  if ?searchtext = ?eof
    then
      close_window ()
      and
      column = ?column + 1
      and
      row = ?row + 1
      and
      window (,white,red,yellow,10,10,60,8)
      and
      say
All required sections of the Envelope Document have been addressed. Please use the CHANGE ONE ITEM option to choose individual items to edit.

Please press the yellow SPACE key to continue.

and column = ?column - 1
and row = ?row - 1
and bypass_unload = Y
and close_window ()
else
    load ('ENVQUEST.HKB')
    and searchlist is read (concat (C:\GARDEN,\?curdir,\'ELISTING.OUT'))
    and close (concat (C:\GARDEN,\?curdir,\'ELISTING.OUT'))
    and searchlist gets 'Quit'
    and chgwant = '
    and while ?chgwant <> Quit
        then
d          do (complete_ENV).

if ?bypass_unload <> Y
    then
        window (,white,red,yellow,l,16,27,4)
        and WRITE ('con:',
'A slight delay will occur while the remainder of the application is loaded. Please stand by. ',')
        and stop_at = where (?searchlist,Quit,2)
        and kounter = 1
        and stop_at = remove(?stop_at,1)
        and while ?kounter < ?stop_at
            then
                eraser = element (?searchlist,?kounter)
                and remove_topic (?eraser)
                and kounter = ?kounter + 1.
        close_window ()
.

(*+++++++++++++++++++
*...) topic 'complete_ENV'.

bypass unload = N.
close window ()
window ('Complete Unanswered Questions',blue,white,white,?column,?row).
close window()
complete gets ?searchlist.
ask ('#e
These subtopics have not yet been addressed. Please
choose the one you wish to complete, or choose Quit
to exit this screen.',chgwant,?complete).
if ?chgwant <> Quit
then
do (?chgwant).
close (concat (C:\GARDEN\?curdir,'\ELISTING.OUT')).
if ?chgwant <> Quit
then
dos (esearch, restore).
close (concat (C:\GARDEN\?curdir,'\ELISTING.OUT')).
searchtext is read line (concat (C:\GARDEN\?curdir,'\ELISTING.OUT')).
if ?searchtext = ?eof
then
close window()
and
window (,white,red,yellow,10,10,60,8)
and
say ('#e
All required sections of the Envelope Document
have now been addressed. Please use the CHANGE
ONE ITEM option to choose individual items to edit.’)
and
close window()
and
chgwant = 'Quit'
else
searchlist is read
(concat (C:\GARDEN\?curdir,'\ELISTING.OUT'))
and
searchlist gets 'Quit'.
close window ()
end. (* complete_ENV *)
end. (* ENV_complete *)

(* ************************************************** *)
topic mark (find_string).
column = ?column + 1.
row = ?row + 1.
text is read ('nasaterm.dat',concat('//',?find_string),'/end').
window (?find_string,blue,white,white,?column,?row,72,).
say (?text).
column = ?column - 1.
row = ?row - 1.
close window ()
end. (* mark *)
end. (* nasasys *)
This program allows the user to eliminate questions within the ENV by selecting topics from a list. Each selected topic's associated file is filled with a 'Not Applicable' entry.

no_edit_key ().
no_debug ().
column = 3.
row = 3.
eof = number_to_line ('CURDIR.DAT').
curdir = string_replace(?curdir, ', ', '', 8).
yn is [YES,NO].
na = 'Not Applicable'.
file = [].

do (nasasys).
new_kb ('envfmenu.hkb').

topic nasasys.
    elim_choices = [].
    WRITE ('con:','#e ') and window (white,red,yellow,1,16,27,4) and WRITE ('con: '
    'One moment please...').
    do (ENV_begin).
    if ?elim_choices = QUIT then new_kb ('envfmenu.hkb').

topic 'ENV begin'.
    load ('ENVquest.hkb').
    remove_topic ('Imaging Requirements').
    remove_topic ('Atmospheric Requirements').
    ENV_topics gets children(ENV_begin).
    ENV_topics gets 'Quit'.
    ENV_topics is remove (?ENV_topics,'related answer').
    close_window ().
end. (* ENV_begin *)

window ('Select Topics',blue,white,white,white,?column,?row,).
ask ('#e
    Which topics do you wish to answer as 'Not Applicable'? 
    Press F4 for #mInstructions#m.,elim_choices,?ENV_topics).

topic 'Instructions'.
    window ('Instructions for Marking Questions as Not Applicable',blue,white,
    white,?column,?row,15).
Please choose the topics to mark as 'Not Applicable' by one of the below two methods.

Mouse Users: Please use the RIGHT side mouse button and click on each topic to be selected. When the list is complete, click the LEFT side mouse button.

Non-mouse users: Please use the arrow keys to move to the topic to be selected, then use the INSERT key to select it. To move from one page to another, use the Page Up/Page Down keys. When the list is complete, press the RETURN key.

Press yellow SPACE to continue.

close window ()
end. (* Instructions *)

close window ()
if ?elim choices <> 'Quit'
    then do (?eliminate_questions).
*)

(topic mark (find_string).
column = ?column + 1.
row = ?row + 1.
text is read ('nasaterm.dat',concat('/','?find_string'),'/end').
window (find_string,blue,white,white,?column,?row,72,).
say (?text).
column = ?column - 1.
row = ?row - 1.
close window ()
end. (* mark *)
end. (* nasasys *)

topic 'eliminate_questions'.
do (write_message).
topic 'write_message'.
    WRITE ('con:','#e ')
    and
    window (,white,red,yellow,1,16,27,4)
    and
    WRITE ('con:','
    'Answers being written...
    ')
end. (* write_message *)

if one_of (?elim_choices,'Description of Experiment Type or Class')
    then
    file = (concat (C:\GARDEN,?CURDIR,'\ENV1_1.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
        then
            new_file (?file)
            and
write (?file, ?na)
else
  close_window ()
  and
  say ('#e

An answer already exists for:

Description of Experiment Type or Class

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.’)
  and
  do (write_message).

close (?file).

if one_of (?elim_choices, 'Scientific Knowledge to be Gained')
then
  file = (concat (C:\GARDEN\, ?CURDIR, '\ENV1_2.DAT'))
  and
  empty_check = read_line (?file)
  and
  if ?empty_check = ?eof
    then
      new_file (?file)
      and
      write (?file, ?na)
    else
      close_window ()
      and
      say ('#e

An answer already exists for:

Scientific Knowledge to be Gained

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.’)
  and
  do (write_message).

close (?file).

if one_of (?elim_choices, 'Value of Knowledge to Scientific Field')
then
  file = (concat (C:\GARDEN\, ?CURDIR, '\ENV1_3.DAT'))
  and
  empty_check = read_line (?file)
  and
  if ?empty_check = ?eof
    then
      new_file (?file)
      and
      write (?file, ?na)
    else
      close_window ()
      and
An answer already exists for:

Value of Knowledge to Scientific Field

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.

and
do (write_message).
close (?file).

if one_of (?elim_choices,'Necessity for Space Environment')
then
  file = (concat (C:\GARDEN,?CURDIR,'\ENV1_4.DAT'))
  and
  empty_check = read_line (?file)
  and
  if ?empty_check = ?eof
    then
      new_file (?file)
      and
      write (?file,?na)
    else
      close_window ()
      and
      say ('#e

An answer already exists for:

Necessity for Space Environment

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.

and
do (write_message).
close (?file).

if one_of (?elim_choices,'Description of Scientific Field')
then
  file = (concat (C:\GARDEN,?CURDIR,'\ENV2_1.DAT'))
  and
  empty_check = read_line (?file)
  and
  if ?empty_check = ?eof
    then
      new_file (?file)
      and
      write (?file,?na)
    else
      close_window ()
      and
      say ('#e

An answer already exists for:
Description of Scientific Field

This answer was retained. Please edit that question if necessary.

Press #yellow SPACE#d to continue.

and
do (write_message).
close (?file).

if one_of (?elim_choices,'Current Applications for Research')
then
  file = (concat (C:\GARDEN\,?CURDIR, '\ENV2_2.DAT'))
  and
  empty_check = read_line (?file)
  and
  if ?empty_check = ?eof
    then
      new_file (?file)
      and
      write (?file,?na)
    else
      close_window ()
      and
      say ('#e

An answer already exists for:

Current Applications for Research

This answer was retained. Please edit that question if necessary.

Press #yellow SPACE#d to continue.

and
do (write_message).
close (?file).

if one_of (?elim_choices,'Account of Prior Research')
then
  file = (concat (C:\GARDEN\,?CURDIR, '\ENV2_3.DAT'))
  and
  empty_check = read_line (?file)
  and
  if ?empty_check = ?eof
    then
      new_file (?file)
      and
      write (?file,?na)
    else
      close_window ()
      and
      say ('#e

An answer already exists for:

Account of Prior Research

This answer was retained. Please edit that question if necessary.
Press #fyellow SPACE#d to continue.

and
do (write_message).
close (?file).

if one_of (?elim_choices, 'Current Research')
then
    file = (concat (C:\GARDEN\, ?CURDIR, '\ENV2_4.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
        then
            new_file (?file)
            and
            write (?file, ?na)
        else
            close_window ()
            and
            say ('#e

An answer already exists for:

Current Research

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.

and
    do (write_message).
close (?file).

if one_of (?elim_choices, 'Relationship to Scientific Field')
then
    file = (concat (C:\GARDEN\, ?CURDIR, '\ENV2_5.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
        then
            new_file (?file)
            and
            write (?file, ?na)
        else
            close_window ()
            and
            say ('#e

An answer already exists for:

Relationship to Scientific Field

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.'
and
do (write_message).
close (?file).

if one_of (?elim_choices,'Anticipated Advance')
then
    file = (concat (C:\GARDEN\,?CURDIR,\'\ENV2_6.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
        then
            new_file (?file)
            and
            write (?file,?na)
        else
            close_window ()
            and
            say ('#e

An answer already exists for:

Anticipated Advance

This answer was retained. Please edit that question if necessary.

    Press #fyellow SPACE#d to continue.')
    and
    do (write_message).
close (?file).

if one_of (?elim_choices,'Limitations of Ground-Based Testing')
then
    file = (concat (C:\GARDEN\,?CURDIR,\'\ENV3_I.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
        then
            new_file (?file)
            and
            write (?file,?na)
        else
            close_window ()
            and
            say ('#e

An answer already exists for:

Limitations of Ground-Based Testing

This answer was retained. Please edit that question if necessary.

    Press #fyellow SPACE#d to continue.')
    and
    do (write_message).
close (?file).
if one_of (?elim_choices,'Limitations of Drop Towers')
then
  file = (concat (C:\GARDEN,\?CURDIR,\'ENV3_2.DAT'))
and
  empty_check = read_line (?file)
and
  if ?empty_check = ?eof
    then
      new_file (?file)
      and
      write (?file,?na)
    else
      close_window ()
      and
      say ('#e

An answer already exists for:

   Limitations of Drop Towers

This answer was retained. Please edit that question if necessary.

    Press #fyellow SPACE#d to continue.')
and
  do (write_message).
close (?file).

if one_of (?elim_choices,'Limitations of Testing in Aircraft')
then
  file = (concat (C:\GARDEN,\?CURDIR,\'ENV3_3.DAT'))
and
  empty_check = read_line (?file)
and
  if ?empty_check = ?eof
    then
      new_file (?file)
      and
      write (?file,?na)
    else
      close_window ()
      and
      say ('#e

An answer already exists for:

   Limitations of Testing in Aircraft

This answer was retained. Please edit that question if necessary.

    Press #fyellow SPACE#d to continue.')
and
  do (write_message).
close (?file).

if one_of (?elim_choices,'Need for Acomodations in the Shuttle')
then
  file = (concat (C:\GARDEN,\?CURDIR,\'ENV3_4.DAT'))
and
empty_check = read_line (?file)
and
if ?empty_check = ?eof
    then
        new_file (?file)
        and
        write (?file,?na)
    else
        close_window ()
        and
        say ('#e
An answer already exists for:

    Need for Acomodations in the Shuttle

This answer was retained. Please edit that question
if necessary.

    Press #fyellow SPACE#d to continue.')

    and
    do (write_message).

    close (?file).

if one_of (?elim_choices,'Limitations of Mathematical Modeling')
    then
        file = (concat (C:\GARDEN\,?CURDIR, '\ENV3_5.DAT'))
        and
        empty_check = read_line (?file)
        and
        if ?empty_check = ?eof
            then
                new_file (?file)
                and
                write (?file,?na)
            else
                close_window ()
                and
                say ('#e
An answer already exists for:

    Limitations of Mathematical Modeling

This answer was retained. Please edit that question
if necessary.

    Press #fyellow SPACE#d to continue.')

    and
    do (write_message).

    close (?file).

if one_of (?elim_choices,'Limitations of Other Modeling Approaches')
    then
        file = (concat (C:\GARDEN\,?CURDIR, '\ENV3_6.DAT'))
        and
        empty_check = read_line (?file)
        and
        if ?empty_check = ?eof
            then
                new_file (?file)
                and
                write (?file,?na)
            else
                close_window ()
                and
                say ('#e
An answer already exists for:

    Limitations of Other Modeling Approaches

This answer was retained. Please edit that question
if necessary.

    Press #fyellow SPACE#d to continue.')

    and
    do (write_message).

    close (?file).
An answer already exists for:

Limitations of Other Modeling Approaches

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.

An answer already exists for:

Description of Type of Experiment

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.

An answer already exists for:

Experiment Procedures to be Used

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.
else
    close_window ()
    and
    say ('#e

An answer already exists for:

Experiment Procedures to be Used

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.'
)
    and
    do (write_message).
    close (?file).

if one_of (?elim_choices,'')
then
    file = (concat (C:\GARDEN,?CURDIR,'\ENV4_3.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
        then
            new_file (?file)
            and
            write (?file,?na)
        else
            close_window ()
            and
            say ('#e

An answer already exists for:

Measurements and Range of Values Required

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.'
)
    and
    do (write_message).
    close (?file).

if one_of (?elim_choices,'Experiment Sample Requirements')
then
    file = (concat (C:\GARDEN,?CURDIR,'\ENV5_1.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
        then
            new_file (?file)
            and
            write (?file,?na)
        else
            close_window ()
            and
            say ('#e
An answer already exists for:

Experiment Sample Requirements

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.

and

do (write_message).

close (?file).

if one_of (?elim_choices,'Atmospheric Requirements (Pressure)')
then
    file = (concat (C:\GARDEN,?CURDIR,'\ENV5_2_1.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
        then
            new_file (?file)
            and
            write (?file,?na)
        else
            close_window ()
            and
            say ('#e

An answer already exists for:

Atmospheric Requirements (Pressure)

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.

and

do (write_message).

close (?file).

if one_of (?elim_choices,'Atmospheric Requirements (Gas Composition)')
then
    file = (concat (C:\GARDEN,?CURDIR,'\ENV5_2_2.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
        then
            new_file (?file)
            and
            write (?file,?na)
        else
            close_window ()
            and
            say ('#e

An answer already exists for:

Atmospheric Requirements (Gas Composition)
This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.’

and
do (write_message).
close (?file).

if one_of (?elim_choices,'Atmospheric Requirements (Humidity)')
then
file = (concat (C:\GARDEN,\?CURDIR,'\ENV5_2_3.DAT'))
and
empty_check = read_line (?file)
and
if ?empty_check = ?eof
then
    new_file (?file)
    and
    write (?file,?na)
else
    close_window ()
    and
    say ('#e

An answer already exists for:

Atmospheric Requirements (Humidity)

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.’

and
do (write_message).
close (?file).

if one_of (?elim_choices,'Atmospheric Requirements (Vacuum)')
then
file = (concat (C:\GARDEN,\?CURDIR,'\ENV5_2_4.DAT'))
and
empty_check = read_line (?file)
and
if ?empty_check = ?eof
then
    new_file (?file)
    and
    write (?file,?na)
else
    close_window ()
    and
    say ('#e

An answer already exists for:

Atmospheric Requirements (Vacuum)

This answer was retained. Please edit that question if necessary.
Press #fyellow SPACE#d to continue.'

and
do (write_message).
close (?file).

if one_of (?elim_choices, 'Temperature Control and Measurement')
then
    file = (concat (C:\GARDEN\, ?CURDIR, '\'ENV5_3.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
    then
        new_file (?file)
        and
        write (?file, ?na)
    else
        close_window ()
        and
        say ('#e

An answer already exists for:

    Temperature Control and Measurement

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.'

and
do (write_message).
close (?file).

if one_of (?elim_choices, 'Vibration Control and Measurement')
then
    file = (concat (C:\GARDEN\, ?CURDIR, '\'ENV5_4.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
    then
        new_file (?file)
        and
        write (?file, ?na)
    else
        close_window ()
        and
        say ('#e

An answer already exists for:

    Vibration Control and Measurement

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.'

and
do (write_message).
close (?file).
if one_of (?elim_choices, 'Test Matrices')
then
  file = (concat (C:\GARDEN\, ?CURDIR, '\ENV5_5.DAT'))
  and
  empty_check = read_line (?file)
  and
  if ?empty_check = ?eof
     then
       new_file (?file)
       and
       write (?file, ?na)
     else
       close_window ()
       and
       say ('#e
An answer already exists for:

Test Matrices

This answer was retained. Please edit that question
if necessary.

Press #fyellow SPACE#d to continue.')
  and
  do (write_message).
  close (?file).

if one_of (?elim_choices, 'Imaging Requirements (Photography)')
then
  file = (concat (C:\GARDEN\, ?CURDIR, '\ENV5_6_1.DAT'))
  and
  empty_check = read_line (?file)
  and
  if ?empty_check = ?eof
     then
       new_file (?file)
       and
       write (?file, ?na)
     else
       close_window ()
       and
       say ('#e
An answer already exists for:

Imaging Requirements (Photography)

This answer was retained. Please edit that question
if necessary.

Press #fyellow SPACE#d to continue.')
  and
  do (write_message).
  close (?file).

if one_of (?elim_choices, 'Imaging Requirements (Radiography)')
then
  file = (concat (C:\GARDEN\, ?CURDIR, '\ENV5_6_2.DAT'))
and
empty_check = read_line (?file)
and
if ?empty_check = ?eof
  then
    new_file (?file)
    and
    write (?file,?na)
  else
    close_window ()
    and
    say ('#e

An answer already exists for:

  Imaging Requirements (Radiography)

This answer was retained. Please edit that question
if necessary.

  Press #fyellow SPACE#d to continue.’)
  and
  do (write_message).

close (?file).

if one_of (?elim_choices,'Imaging Requirements (Television)')
  then
    file = (concat (C:\GARDEN\\?CURDIR,\'\ENV5_6_3.DAT\'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
      then
        new_file (?file)
        and
        write (?file,?na)
      else
        close_window ()
        and
        say ('#e

An answer already exists for:

  Imaging Requirements (Television)

This answer was retained. Please edit that question
if necessary.

  Press #fyellow SPACE#d to continue.’)
  and
  do (write_message).

close (?file).

if one_of (?elim_choices,'Imaging Requirements (Resolution)')
  then
    file = (concat (C:\GARDEN\\?CURDIR,\'\ENV5_6_4.DAT\'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
      then
        new_file (?file)
        and
        write (?file,?na)
      else
        close_window ()
        and
        say ('#e
then
    new_file (?file)
    and
    write (?file,?na)
else
    close_window ()
    and
    say ("#e

An answer already exists for:

    Imaging Requirements (Resolution)

This answer was retained. Please edit that question if necessary.

    Press #fyellow SPACE#d to continue."
    and
    do (write_message).
close (?file).

if one_of (?elim_choices,'Imaging Requirements (Frame Rate)')
then
    file = (concat (C:\GARDE\,?CURDIR,'\ENV5_6_5.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
        then
            new_file (?file)
            and
            write (?file,?na)
        else
            close_window ()
            and
            say ("#e

An answer already exists for:

    Imaging Requirements (Frame Rate)

This answer was retained. Please edit that question if necessary.

    Press #fyellow SPACE#d to continue."
    and
    do (write_message).
close (?file).

if one_of (?elim Choices,'Electromagnetic Limitations')
    then
        file = (concat (C:\GARDEN\,?CURDIR,'\ENV5_7.DAT'))
        and
        empty_check = read_line (?file)
        and
        if ?empty_check = ?eof
            then
                new_file (?file)
                and
                write (?file,?na)
else
  close_window ()
  and
  say ('e

An answer already exists for:

Electromagnetic Limitations

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.

and
do (write_message).
close (?file).

if one_of (?elim_choices, 'Astronaut Involvement (Extravehicular Activity)')
then
  file = (concat (C:\GARDEN, ?CURDIR, '\ENV5_8_1.DAT'))
  and
  empty_check = read_line (?file)
  and
  if ?empty_check = ?eof
    then
      new_file (?file)
      and
      write (?file, ?na)
    else
      close_window ()
      and
      say ('e

An answer already exists for:

Astronaut Involvement (Extravehicular Activity)

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.

and
do (write_message).
close (?file).

if one_of (?elim_choices, 'Astronaut Involvement (Activation of Experiment)')
then
  file = (concat (C:\GARDEN, ?CURDIR, '\env5_8_2.DAT'))
  and
  empty_check = read_line (?file)
  and
  if ?empty_check = ?eof
    then
      new_file (?file)
      and
      write (?file, ?na)
    else
      close_window ()
      and
      say ('e
An answer already exists for:

Astronaut Involvement (Activation of Experiment)

This answer was retained. Please edit that question if necessary.

Press #ff yellow SPACE #d to continue.

and

do (write_message).

close (?file).

if one_of (?elim_choices, 'Data Requirements')
then
    file = (concat (C:\GARDEN\, ?CURDIR, '\ENV5_9.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
        then
            new_file (?file)
            and
            write (?file, ?na)
        else
            close_window ()
            and
            say ('

An answer already exists for:

Data Requirements

This answer was retained. Please edit that question if necessary.

Press #ff yellow SPACE #d to continue.

and

do (write_message).

close (?file).

if one_of (?elim_choices, 'Telepresence')
then
    file = (concat (C:\GARDEN\, ?CURDIR, '\EN5_10_1.DAT'))
    and
    empty_check = read_line (?file)
    and
    if ?empty_check = ?eof
        then
            new_file (?file)
            and
            write (?file, ?na)
        else
            close_window ()
            and
            say ('

An answer already exists for:

Telepresence
This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.’)

and
do (write_message).
close (?file).

if one_of (?elim_choices,'Telerobotics')
then
  file = (concat (C:\GARDEN\,?CURDIR,\'\EN5_10_2.DAT\'))
  and
  empty_check = read_line (?file)
  and
  if ?empty_check = ?eof
    then
      new_file (?file)
      and
      write (?file,?na)
    else
      close_window ()
      and
      say ('#e

An answer already exists for:

Telerobotics

This answer was retained. Please edit that question if necessary.

Press #fyellow SPACE#d to continue.’)

and
do (write_message).
close (?file).

if one_of (?elim_choices,'Other Requirements')
then
  file = (concat (C:\GARDEN\,?CURDIR,\'\ENV6_1.DAT\'))
  and
  empty_check = read_line (?file)
  and
  if ?empty_check = ?eof
    then
      new_file (?file)
      and
      write (?file,?na)
    else
      close_window ()
      and
      say ('#e

An answer already exists for:

Other Requirements

This answer was retained. Please edit that question if necessary.
Press #fyellow SPACE#d to continue.

and
do (write_message).
close (?file).
end. (* eliminate_questions *)
(* Envquest.kb *)
(*) These are the questions to be asked, and the files where the answers are to be stored, for the completion of the Science Requirements Envelope Document.

no_edit_key ()
no_debug ()

**topic 'Description of experiment type or class'**.
window ('Description of Experiment Type or Class (Question 1 of 39)',blue,white,white,2,2,70,6)
say ('#e Please enter a narrative description of the type or class of the #mexperiment#m. This topic is also addressed under the headings "Experiment Procedures to be Used" and "General Description of Type of Experiments".

Press the #fyellow RETURN KEY#d to enter the editor,
#fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.‘)
edit_file (concat (C:\GARDEN\,?CURDIR,\'\ENV1_1.DAT’),yellow,black,red,5,10,70,1)
close_window ().
close -- (concat (C:\GARDEN\,?CURDIR,\'\ENV1_1.DAT’)).
end. (* Description of experiment *)

**topic 'Scientific Knowledge to be Gained'**.
window ('Scientific Knowledge to be Gained (Question 2 of 39)',blue,white,white,2,2,70,6)
say ('#e Please enter the scientific knowledge to be gained through this experiment. This topic is also addressed under the heading "Anticipated Advance in State of the Art".

Press the #fyellow RETURN KEY#d to enter the editor,
#fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.‘)
edit_file (concat (C:\GARDEN\,?CURDIR,\'\ENV1_2.DAT’),yellow,black,red,5,9,70,1)
close_window ().
close -- (concat (C:\GARDEN\,?CURDIR,\'\ENV1_2.DAT’)).
end. (* Scientific Knowledge to be Gained *)

**topic 'Value of Knowledge to Scientific Field'**.
window ('Value of Knowledge to Scientific Field (Question 3 of 39)',blue,white,white,2,2,70,6)
say (' Please enter a brief narrative describing the value of knowledge of this type of experimentation to scientific field.

Press the #fyellow RETURN KEY#d to enter the editor,
#fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.’)
edit_file (concat (C:\GARDEN\,?CURDIR,\'\ENV1_3.DAT’),yellow,black,red,5,9,70,1)
close_window ().
close -- (concat (C:\GARDEN\,?CURDIR,\'\ENV1_3.DAT’)).
end. (* Value of Knowledge to Scientific Field *)

**topic 'Necessity for Space Environment'**.
window ('Necessity for Space Environment to Experiment (Question 4 of 39)', blue, white, white, 2, 2, 70, 6).

say ('Please enter a narrative justifying the need for a space environment.

Press the #fyellow RETURN KEY#d to enter the editor,
#fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.'

edit_file (concat (C:\GARDEN\?CURDIR,\ENV1_4.DAT'), yellow, black, red, 5, 9, 70, 1
close_window ()
close (concat (C:\GARDEN\?CURDIR,\ENV1_4.DAT')).
end. (* Justification of the Need for Space Environment *)

topic 'Description of Scientific Field'.
window ('Description of Scientific Field (Question 5 of 39)', blue, white, white,

say ('Please enter a description of the scientific field to which the experiment belongs.

Press the #fyellow RETURN KEY#d to enter the editor,
#fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.')

edit_file (concat (C:\GARDEN\?CURDIR,\ENV2_1.DAT'), yellow, black, red, 5, 9, 70, 1
close_window ()
close (concat (C:\GARDEN\?CURDIR,\ENV2_1.DAT')).
end. (* Description of Scientific Field *)

topic 'Current Applications for Research'.
window ('Current Applications for Research (Question 6 of 39)', blue, white, white,

say ('Please enter a narrative of the current applications for research in the field.

Press the #fyellow RETURN KEY#d to enter the editor,
#fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.'

edit_file (concat (C:\GARDEN\?CURDIR,\ENV2_2.DAT'), yellow, black, red, 5, 9, 70, 1
close_window ()
close (concat (C:\GARDEN\?CURDIR,\ENV2_2.DAT')).
end. (* Current Applications for Research *)

topic 'Account of Prior Research'.
window ('Account of Prior Research (Question 7 of 39)', blue, white, white, 2, 2, 70

say ('Please enter a brief historical account of the prior research in the field.

Press the #fyellow RETURN KEY#d to enter the editor,
#fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.'

edit_file (concat (C:\GARDEN\?CURDIR,\ENV2_3.DAT'), yellow, black, red, 5, 9, 70, 1
close_window ()
close (concat (C:\GARDEN\?CURDIR,\ENV2_3.DAT')).
end. (* Account of Prior Research *)
topic 'Current Research'.

window ('Current Research (Question 8 of 39)',blue,white,white,2,2,70,6).

say ('

Please enter a brief account of the current research in the field.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

')

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV2_4.DAT'),yellow,black,red,5,9,70,1)
close_window ()
close (concat (C:\GARDEN\,?CURDIR,'\ENV2_4.DAT'))
end. (* Current Research *)

topic 'Relationship to Scientific Field'.

window ('Relationship to Scientific Field (Question 9 of 39)',blue,white,white,2,2,70,8).

say ('

Please enter an account of the relationship of proposed experiment to the scientific field.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

')

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV2_5.DAT'),yellow,black,red,5,9,70,1)
close_window ()
close (concat (C:\GARDEN\,?CURDIR,'\ENV2_5.DAT'))
end. (* Relationship to Scientific Field *)

topic 'Anticipated Advance'.

window ('Anticipated Advance (Question 10 of 39)',blue,white,white,2,2,70,8).

relatedfile = '\ENVI 2.DAT'.
relatedtopic = 'Scientific Knowledge to be Gained'.
filename = '\ENV2 6.DAT'.

say ('In a #mrelated answer#m you were asked to describe the "Scientific Knowledge to be Gained". Please elaborate on that answer to give a brief account of the anticipated advance in state of the art.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

')

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV2_6.DAT'),yellow,black,red,5,12,70,1)
close_window ()
close (concat (C:\GARDEN\,?CURDIR,'\ENV2_6.DAT'))
end. (* Anticipated Advance *)

topic 'Scientific Knowledge to be Gained'.

window ('Scientific Knowledge to be Gained',blue,white,white,2,2,70,6).

say ('Please enter the scientific knowledge to be gained through this experiment.'
Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.'

edit_file (concat (C:\GARDEN\,?CURDIR,'\\ENV1_2.DAT')),yellow,black,red,5,9,70,1
close_window ()
close (concat (C:\GARDEN\,?CURDIR,'\\ENV1_2.DAT')).
end. (* Scientific Knowledge to be Gained *)
end. (* Anticipated Advance *)
topic 'Limitations of Ground-Based Testing'.
window ('Limitations of Ground-Based Testing (Question 11 of 39)',blue,white,white,2,2,
say ('Please enter the limitations of ground-based testing.
Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.'

edit_file (concat (C:\GARDEN\,?CURDIR,'\\ENV3_1.DAT')),yellow,black,red,5,9,70,1
close_window ()
close (concat (C:\GARDEN\,?CURDIR,'\\ENV3_1.DAT')).
end. (* Limitations of Ground-Based Testing *)
topic 'Limitations of Drop Towers'.
window ('Limitations of Drop Towers (Question 12 of 39)',blue,white,white,2,2,
say ('Please enter the limitations of drop towers.
Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.')

edit_file (concat (C:\GARDEN\,?CURDIR,'\\ENV3_2.DAT')),yellow,black,red,5,9,70,1
close_window ()
close (concat (C:\GARDEN\,?CURDIR,'\\ENV3_2.DAT')).
end. (* Limitations of Drop Towers *)
topic 'Limitations of Testing in Aircraft'.
window ('Limitations of Testing in Aircraft (Question 13 of 39)',blue,white,white,2,2,
say ('Please enter the limitations of testing in aircraft.
Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.'

edit_file (concat (C:\GARDEN\,?CURDIR,'\\ENV3_3.DAT')),yellow,black,red,5,9,70,1
close_window ()
close (concat (C:\GARDEN\,?CURDIR,'\\ENV3_3.DAT')).
end. (* Limitations of Testing in Aircraft *)
topic 'Need for accommodations in the Shuttle'.
window ('Need for Accommodations in the Shuttle (Question 14 of 39)',blue,white,
say ('Please enter the need for accommodations in the Shuttle.
Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.
topic 'Limitations of Mathematical Modeling'.
  window ('Limitations of Mathematical Modeling (Question 15 of 39)',blue,white,
  say ('
    Please enter the limitations of mathematical modeling.

    Press the #fyellow RETURN KEY#d to enter the editor, and
    #fyellow ESC#d to leave editor.').

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV3_5.DAT'),yellow,black,red,5,9,70,1
   close_window ()).
   close (concat (C:\GARDEN\,?CURDIR,'\ENV3_5.DAT')).
end. (* Limitations of Mathematical Modeling *)

topic 'Limitations of Other Modeling Approaches'.
  window ('Limitations of Other Modeling Approaches (Question 16 of 39)',blue,white,wh
  say ('
    Please enter the limitations of other modeling approaches.

    Press the #fyellow RETURN KEY#d to enter the editor,
    #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV3_6.DAT'),yellow,black,red,5,9,70,
   close_window ()).
   close (concat (C:\GARDEN\,?CURDIR,'\ENV3_6.DAT')).
end. (* Limitations of Other Modeling Approaches *)

topic 'Description of Type of Experiment'.
  window ('Description of Type of Experiment (Question 17 of 39)',blue,white,wh
  relatedfile = '\ENV1_1.DAT'.
  relatedtopic = 'Description of Experiment Type or Class'.
  filename = '\ENV4_1.DAT'.
  say ('
    In a #mrelated answer#m you were asked to describe the
    "Description of Experiment Type or Class". Please elaborate
    on that answer to give a general description of the TYPE
    of experiments.

    Press the #fyellow RETURN KEY#d to enter the editor,
    #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV4_1.DAT'),yellow,black,red,5,12,70,
   close_window ()).
   close (concat (C:\GARDEN\,?CURDIR,'\ENV4_1.DAT')).

topic 'Description of Experiment Type or Class'.
  window ('Description of Experiment Type or Class (Question 1 of 39)',blue,white,wh
  say ('#
     Please enter a narrative description of the type or class
     of the #mexperiment#m. This topic is also addressed under
     the headings "Experiment Procedures to be Used" and "General
     Description of Type of Experiments".

     Press the #fyellow RETURN KEY#d to enter the editor,
     #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV4_1.DAT'),yellow,black,red,5,12,70,
   close_window ()).
   close (concat (C:\GARDEN\,?CURDIR,'\ENV4_1.DAT')).
Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.'

edit_file (concat (C:\GARDEN,\?CURDIR,'\ENV1_1.DAT'),yellow,black,red,5,9,70,1
close_window ()).
close (concat (C:\GARDEN,\?CURDIR,'\ENV1_1.DAT')).
end. (* Description of experiment type or class *)
end. (* Description of type of experiments *)

topic 'Experiment Procedures to be Used'.
window ('Experiment Procedures to be Used (Question 18 of 39)',blue,white,white,2,2,70,6).
relatedfile = '\ENV1_1.DAT'.
relatedtopic = 'Description of Experiment Type or Class'.
filename = '\ENV4_2.DAT'.
say ('In a #mrelated answer#m you were asked to describe the "Description of Experiment Type or Class". Please elaborate on that answer to describe the TYPES of experiment procedures to be used.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C:\GARDEN,\?CURDIR,'\ENV4_2.DAT'),yellow,black,red,5,12,70,1
close_window ()).
close (concat (C:\GARDEN,\?CURDIR,'\ENV4_2.DAT')).

end. (* Description of experiment type or class *)
end. (* Description of type of experiments *)

topic 'Measurements and Range of Values Required'.
window ('Measurements and Range of Values Required (Question 19 of 39)',blue,white,white,2,2,70,6).
say ('Please enter the types of measurements and the ranges of values required for the experiment.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C:\GARDEN,\?CURDIR,'\ENV4_3.DAT'),yellow,black,red,5,9,70,1
topic 'Experiment Sample Requirements'.
window ('Experiment Sample Requirements (Question 20 of 39)', blue, white, white,
say ('Please enter the experiment sample requirements,
both in terms of number and materials.

Press the #fyellow RETURN KEY#d to enter the editor,
#fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save

edit_file (concat (C:\GARDEN\, ?CURDIR, '\ENV5_1.DAT'), yellow, black, red, 5, 9, 70, 1)
close_window ().
close (concat (C:\GARDEN\, ?CURDIR, '\ENV5_1.DAT')).
end. (* Experiment Sample Requirements *)

topic 'Atmospheric Requirements'.
window ('Atmospheric Requirements', blue, white, white, 2, 2, 70, 6).
changen = NO.
na = 'NOT APPLICABLE'.

ask ('Are there any atmospheric requirements for your experiments?',
atmosreqt, ?yn).
eof = number to char (26).
if ?atmosreqt = 'YES'
  then
    do ('Atmospheric Requirements (Pressure)')
    and
doi ('Atmospheric Requirements (Gas Composition)')
    and
doi ('Atmospheric Requirements (Humidity)')
    and
doi ('Atmospheric Requirements (Vacuum)')
else
  pressure = read_line (concat(C:\GARDEN\, ?CURDIR, '\ENV5_2_1.DAT'))
  and
  pressure is string_copy(?pressure, 1, 14)
  and
  gas = read_line (concat(C:\GARDEN\, ?CURDIR, '\ENV5_2_2.DAT'))
  and
  gas is string_copy(?gas, 1, 14)
  and
  hum = read_line (concat(C:\GARDEN\, ?CURDIR, '\ENV5_2_3.DAT'))
  and
  hum is string_copy(?hum, 1, 14)
  and
  vacuum = read_line (concat(C:\GARDEN\, ?CURDIR, '\ENV5_2_4.DAT'))
  and
  vacuum is string_copy(?vacuum, 1, 14)
  and
  if (?pressure <> ?NA and ?pressure <> ?eof)
or
or
   (?hum  <> ?NA and ?hum  <> ?eof)
then
window (" ',yellow,blue,yellow,2,2,70,14)
and
ask ("
There are answers on file that indicate this question was previously answered YES, rather than NO. If the correct answer is NO, the system will need to change the answers from your previous session that were directly related to the YES response, since they are no longer applicable. Do you authorize the system to change these previous answers?' ,em,?yn)
and
close_window ()
else
    changem = YES.

if changem = YES
then
new_file (concat(C:\GARDEN\,?CURDIR,'\ENV5_2_1.DAT'))
and
write (concat(C:\GARDEN\,?CURDIR,'\ENV5_2_1.DAT'),?NA)
and
close (concat(C:\GARDEN\,?CURDIR,'\ENV5_2_1.DAT'))
and
new_file (concat(C:\GARDEN\,?CURDIR,'\ENV5_2_2.DAT'))
and
write (concat(C:\GARDEN\,?CURDIR,'\ENV5_2_2.DAT'),?NA)
and
close (concat(C:\GARDEN\,?CURDIR,'\ENV5_2_2.DAT'))
and
new_file (concat(C:\GARDEN\,?CURDIR,'\ENV5_2_3.DAT'))
and
write (concat(C:\GARDEN\,?CURDIR,'\ENV5_2_3.DAT'),?NA)
and
close (concat(C:\GARDEN\,?CURDIR,'\ENV5_2_3.DAT'))
and
new_file (concat(C:\GARDEN\,?CURDIR,'\ENV5_2_4.DAT'))
and
write (concat(C:\GARDEN\,?CURDIR,'\ENV5_2_4.DAT'),?NA)
and
close (concat(C:\GARDEN\,?CURDIR,'\ENV5_2_4.DAT')).

topic 'Atmospheric Requirements (Gas Composition)'.
window ('Atmospheric Requirements (Pressure) (Question 21 of 39)',blue,white,w
say ("
Please enter the experiment requirements in terms of atmospheric pressure.
Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.'

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV5_2_1.DAT'),yellow,black,red,5,9,70
close_window ()
.
close (concat (C:\GARDEN\,?CURDIR,'\ENV5_2_1.DAT')).
end. (* Atmospheric Requirements (Pressure) *)

topic 'Atmospheric Requirements (Gas Composition)'.
window('Atmospheric Requirements (Gas Composition) (Question 22 of 39)',blue,white)
say('Please enter the experiment requirements in terms of atmospheric gas composition.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

directory_file (concat (C:\GARDEN\?CURDIR,'\ENV5_2_2.DAT'),yellow,black,red,5,9,70)
close_window ()
close (concat (C:\GARDEN\?CURDIR,'\ENV5_2_2.DAT')).
end. (* Atmospheric Requirements (Gas Composition) *)

topic 'Atmospheric Requirements (Humidity)'.
window ('Atmospheric Requirements (Humidity) (Question 23 of 39)',blue,white,white)
say('Please enter the experiment requirements in terms of atmospheric humidity.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

directory_file (concat (C:\GARDEN\?CURDIR,'\ENV5_2_3.DAT'),yellow,black,red,5,9,70)
close_window ()
close (concat (C:\GARDEN\?CURDIR,'\ENV5_2_3.DAT')).
end. (* Atmospheric Requirements (Humidity) *)

topic 'Atmospheric Requirements (Vacuum)'.
window ('Atmospheric Requirements (Vacuum) (Question 24 of 39)',blue,white,white)
say('Please enter the experiment requirements in terms of atmospheric vacuum.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

directory_file (concat (C:\GARDEN\?CURDIR,'\ENV5_2_4.DAT'),yellow,black,red,5,9,70)
close_window ()
close (concat (C:\GARDEN\?CURDIR,'\ENV5_2_4.DAT')).
end. (* Atmospheric Requirements (Vacuum) *)
close_window ().
END. (*Atmospheric Requirements*)

topic 'Temperature Control and Measurement'.
window ('Temperature Control and Measurement (Question 25 of 39)',blue,white,white)
say('Please enter the experiment requirements in terms of temperature control, measurement range, and the accuracy required.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

directory_file (concat (C:\GARDEN\?CURDIR,'\ENV5_3.DAT'),yellow,black,red,5,10,70)
close_window ()
close (concat (C:\GARDEN\?CURDIR,'\ENV5_3.DAT')).
end. (* Temperature Control and Measurement *)

topic 'Vibration Control and Measurement'.
    window ('Vibration Control and Measurement (Question 26 of 39)',blue,white,wh.
    say ('Please enter the experiment requirements in terms of vibration control and measurement range, along with the accuracy and frequency of measurements required.

    Press the yellow RETURN KEY to enter the editor, yellow ESC to leave editor, and yellow RETURN to confirm sa

    edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV5_4.DAT'),yellow,black,red,5,10,70, close_window ()).
    close (concat (C:\GARDEN\,?CURDIR,'\ENV5_4.DAT')).
end. (* Vibration Control and Measurement *)

topic 'Test Matrices'.
    window ('Test Matrices (Question 27 of 39)',blue,white,white,2,2,70,6).
    say ('Please enter the experiment requirements in the form of test matrices to describe the number and duration of tests required.

    Press the yellow RETURN KEY to enter the editor, yellow ESC to leave editor, and yellow RETURN to confirm sa

    edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV5_5.DAT'),yellow,black,red,5,9,70, close_window ()).
    close (concat (C:\GARDEN\,?CURDIR,'\ENV5_5.DAT')).
end. (* Test Matrix *)

topic 'Imaging Requirements'.
    window ('Imaging Requirements',blue,white,white,2,2,70,6).
    changem = NO.
    na = 'NOT APPLICABLE'.
    ask ('Are there any imaging requirements for your experiments?', imagereqt,?yn).
    eof = number_to_char (26).
    if ?imagereqt = YES then
        do ('Imaging Requirements (Photography)')
        and
        do ('Imaging Requirements (Radiography)')
        and
        do ('Imaging Requirements (Television)')
        and
        do ('Imaging Requirements (Resolution)')
        and
        do ('Imaging Requirements (Frame Rate)')
    else
        photo = read_line (concat(C:\GARDEN\,?CURDIR,'\ENV5_6_1.DAT'))
        and
        photo is string_copy(?photo,1,14)
        and
        radio = read_line (concat(C:\GARDEN\,?CURDIR,'\ENV5_6_2.DAT'))
        and
radio is string_copy(?radio,1,14)
and
    tv = read_line (concat(C:\GARDEN\,?CURDIR,\'\ENV5_6_3.DAT'))
and
tv is string_copy(?tv,1,14)
and
resolution = read_line (concat(C:\GARDEN\,?CURDIR,\'\ENV5_6_4.DAT'))
and
resolution is string_copy(?resolution,1,14)
and
frame = read_line (concat(C:\GARDEN\,?CURDIR,\'\ENV5_6_5.DAT'))
and
frame is string_copy(?frame,1,14)
and
    if (?photo <> ?NA and ?photo <> ?eof) 
or
    (?radio <> ?NA and ?radio <> ?eof) 
or
    (?tv <> ?NA and ?tv <> ?eof) 
or
    (?resolution <> ?NA and ?resolution <> ?eof) 
or
    (?frame <> ?NA and ?frame <> ?eof)
then
    window (' ',yellow,blue,yellow,2,2,70,14)
    and
    ask ('There are answers on file that indicate this question was previously answered YES, rather than NO. If the correct answer is NO, the system will need to change the answers from your previous session that were directly related to the YES response, since they are no longer applicable. Do you authorize the system to change these previous answers?',changem,?yn)
    and
    close_window ()
else
    changem = YES.

if ?changem = YES
then
    new_file (concat(C:\GARDEN\,?CURDIR,\'\ENV5_6_1.DAT'))
and
    write (concat(C:\GARDEN\,?CURDIR,\'\ENV5_6_1.DAT’),?NA)
and
    close (concat(C:\GARDEN\,?CURDIR,\'\ENV5_6_1.DAT’))
and
    new_file (concat(C:\GARDEN\,?CURDIR,\'\ENV5_6_2.DAT’))
and
    write (concat(C:\GARDEN\,?CURDIR,\'\ENV5_6_2.DAT’),?NA)
and
    close (concat(C:\GARDEN\,?CURDIR,\'\ENV5_6_2.DAT’))
and
    new_file (concat(C:\GARDEN\,?CURDIR,\'\ENV5_6_3.DAT’))
and
    write (concat(C:\GARDEN\,?CURDIR,\'\ENV5_6_3.DAT’),?NA)
and
    close (concat(C:\GARDEN\,?CURDIR,\'\ENV5_6_3.DAT’))
and
    new_file (concat(C:\GARDEN\,?CURDIR,\'\ENV5_6_4.DAT’))
and write (concat(C: \GARDEN\, ?CURDIR, 'ENV5_6_4.DAT'), ?NA)
and close (concat(C: \GARDEN\, ?CURDIR, 'ENV5_6_4.DAT'))
and new_file (concat(C: \GARDEN\, ?CURDIR, 'ENV5_6_5.DAT'))
and write (concat(C: \GARDEN\, ?CURDIR, 'ENV5_6_5.DAT'), ?NA)
and close (concat(C: \GARDEN\, ?CURDIR, 'ENV5_6_5.DAT')).

**Imaging Requirements (Photography)**

window ('Imaging Requirements (Photography) (Question 28 of 39)', blue, white, white)
say ('Please enter the imaging requirements of your experiment in regards to photography needs.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C: \GARDEN\, ?CURDIR, 'ENV5_6_1.DAT'), yellow, black, red, 5, 9, 7)
close_window ().
close (concat (C: \GARDEN\, ?CURDIR, 'ENV5_6_1.DAT')).
end. (* Imaging Requirements (Photography) *)

**Imaging Requirements (Radiography)**

window ('Imaging Requirements (Radiography) (Question 29 of 39)', blue, white, white)
say ('Please enter the imaging requirements of your experiment in regards to radiography needs.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C: \GARDEN\, ?CURDIR, 'ENV5_6_2.DAT'), yellow, black, red, 5, 9, 7)
close_window ().
close (concat (C: \GARDEN\, ?CURDIR, 'ENV5_6_2.DAT')).
end. (* Imaging Requirements (Radiography) *)

**Imaging Requirements (Television)**

window ('Imaging Requirements (Television) (Question 30 of 39)', blue, white, white)
say ('Please enter the imaging requirements of your experiment in regards to television needs.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C: \GARDEN\, ?CURDIR, 'ENV5_6_3.DAT'), yellow, black, red, 5, 9, 7)
close_window ().
close (concat (C: \GARDEN\, ?CURDIR, 'ENV5_6_3.DAT')).
end. (* Imaging Requirements (Television) *)

**Imaging Requirements (Resolution)**

window ('Imaging Requirements (Resolution) (Question 31 of 39)', blue, white, white)
Please enter the imaging requirements of your experiment in regards to resolution needs.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV5_6_4.DAT'),yellow,black,red,5,9,70).
close_window ().
close (concat (C:\GARDEN\,?CURDIR,'\ENV5_6_4.DAT')).
end. (* Imaging Requirements (Resolution) *)

Please enter the imaging requirements of your experiment in regards to frame rate needs.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV5_6_5.DAT'),yellow,black,red,5,9,70).
close_window ().
close (concat (C:\GARDEN\,?CURDIR,'\ENV5_6_5.DAT')).
end. (* Imaging Requirements (Frame Rate) *)

Please enter the electromagnetic limitations for the type of your experiment.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV5_7.DAT'),yellow,black,red,5,9,70).
close_window ().
close (concat (C:\GARDEN\,?CURDIR,'\ENV5_7.DAT')).
end. (* Electromagnetic Limitations *)

Please enter the astronaut involvement in regards to extravehicular activity.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV5_8_1.DAT'),yellow,black,red,5,9,70).
close_window ().
close (concat (C:\GARDEN\,?CURDIR,'\ENV5_8_1.DAT')).
end. (* Astronaut Involvement (Extravehicular Activity) *)

Please enter the astronaut involvement in regards to activation of experiment.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV5_8_2.DAT'),yellow,black,red,5,9,70).
close_window ().
close (concat (C:\GARDEN\,?CURDIR,'\ENV5_8_2.DAT')).
end. (* Astronaut Involvement (Activation of Experiment) *)
window ('Astronaut Involvement (Activation of Experiment) (Question 35 of 39)', blue, white, white, 2, 2, 70, 6).

say ('Please enter the astronaut involvement in regards to their activating the experiment.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save

edit_file (concat (C:\GARDEN\, ?CURDIR, '\env5_8_2.DAT'), yellow, black, red, 5, 9, 70)
close_window ().
close (concat (C:\GARDEN\, ?CURDIR, '\env5_8_2.DAT')).
end. (* Astronaut Involvement (Activation of Experiment) *)

topic 'Data Requirements'.
  window ('Data Requirements (Question 36 of 39)', blue, white, white, 2, 2, 70, 6).

say ('Please enter the data requirements for your experiment.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save

edit_file (concat (C:\GARDEN\, ?CURDIR, '\env5_9.DAT'), yellow, black, red, 5, 9, 70, 1)
close_window ().
close (concat (C:\GARDEN\, ?CURDIR, '\env5_9.DAT')).
end. (* Data Requirements *)

topic 'Telepresence'.
  window ('Telepresence (Question 37 of 39)', blue, white, white, 2, 2, 70, 6).

say ('Please enter the telepresence requirements for your experiment.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save

edit_file (concat (C:\GARDEN\, ?CURDIR, '\EN5_10_1.DAT'), yellow, black, red, 5, 9, 70)
close_window ().
close (concat (C:\GARDEN\, ?CURDIR, '\EN5_10_1.DAT')).
end. (* Telepresence *)

topic 'Telerobotics'.
  window ('Telerobotics (Question 38 of 39)', blue, white, white, 2, 2, 70, 6).

say ('Please enter the telerobotics requirements for your experiment.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save

edit_file (concat (C:\GARDEN\, ?CURDIR, '\EN5_10_2.DAT'), yellow, black, red, 5, 9, 70)
close_window ()
close (concat (C:\GARDEN\, ?CURDIR, '\EN5_10_2.DAT')).
end. (* Telerobotics *)
As the Principal Investigator (#mPI#m), please describe other applicable material not addressed in these requirements.

Press the #fyellow RETURN KEY#d to enter the editor, #fyellow ESC#d to leave editor, and #fyellow RETURN#d to confirm save.

edit_file (concat (C:\GARDEN\,?CURDIR,'\ENV6_1.DAT'),yellow,black,red,5,9,70,1).
close_window ()..
close (concat (C:\GARDEN\,?CURDIR,'\ENV6_1.DAT')).
end. (* Other Requirements *)

topic 'related answer'.
related_answer is read(concat(C:\GARDEN\,?CURDIR,?RELATEDFILE)).
window ('Related Topic',blue,white,white,2,2,76,14).
say ('
This is your answer for the: #fyellow #t',
?relatedtopic,#d #n #n,
?related_answer,#n
Press #fyellow SPACE #d to continue.').
window ('Use this answer?',yellow,blue,red,2,12,74,6).
ask ('Would you like to incorporate this answer into your current response?',
incorporate,?yn).
if ?incorporate = YES
then write (concat(C:\GARDEN\,?CURDIR,?FILENAME),?related_answer).
close_window ()..
close_window ()..
close (concat(C:\GARDEN\,?CURDIR,?FILENAME)).