112980

AgRISTARS

SR-PO-04025
NAS9-15466

A Joint Program for Agriculture and Resources Inventory Surveys Through Aerospace Remote Sensing

Supporting Research

November 1980

Final Report

Vol. IV
Computer Processing Support

by J.L. Kast, P.L. Jobusch, S.K. Hunt, L.A. Kraemer

Purdue University
Laboratory for Applications of Remote Sensing
West Lafayette, Indiana 47907

SR-PO-04025
NAS9-15466
LARS 112980

Final Report

COMPUTER PROCESSING SUPPORT

J.L. Kast, S.K. Hunt, L. Kraemer and P. Jobusch

Purdue University
Laboratory for Applications of Remote Sensing
West Lafayette, Indiana 47907

November 1980

Star Information Form

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
SR-PO-04025		

4. Title and Subtitle Computer Processing Support	5. Report Date November 30, 1980
	6. Performing Organization Code
7. Author(s) James L. Kast, S. Kay Hunt, Luke Kraemer, Peter Jobusch	8. Performing Organization Report No. 112980
9. Performing Organization Name and Address Laboratory for Applications of Remote Sensing Purdue University 1220 Potter Dr. West Lafayette, IN 47906	10. Work Unit No. 11. Contract or Grant No. NAS9-15466
12. Sponsoring Agency Name and Address NASA/Johnson Space Center Earth Observations Division Houston, TX 77058	13. Type of Report and Period Covered Final Report 12-1-79 to 11-30-80 14. Sponsoring Agency Code

15. Supplementary Notes

Principal Investigator - M.E. Bauer
Technical Monitor - F. Hall

16. Abstract

Results of a task providing a shared data processing environment to geographically dispersed groups researching remote sensing of agriculture are presented.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Remote sensing, remote terminal network, earth observation, technology transfer, data processing network, research, communication, cooperative program, data base, research computing.	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*

*For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA - JSC

/

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF APPENDICES

COMPUTER PROCESSING SUPPORT*

A. OVERVIEW

by J. L. Kast

1. Background

For the past three years investigators at Purdue and Johnson Space Center (JSC) have shared the computer facility at the Laboratory for Applications of Remote Sensing (LARS) as their primary data processing environment for research. The Computer Processing Support Task has supported JSC's remote sensing research programs by:

* providing access to a modern computer facility designed and implemented to support remote sensing research needs;
* providing training in the use of the hardware and software available on the computer system;
* providing consulting support to the remote users of the facility;
* developing and acquiring software supporting and enhancing the utility of the facility;

---

* The work conducted under Task 3A, Computer Processing Support, is the product of a team effort at Purdue and JSC. For JSC to receive responsive computer service from an installation 1100 miles distant (at Purdue), requires considerable cooperation and understanding. I believe JSC and Purdue have been largely successful in sharing a computational environment through very hard work by people at both institutions.

I offer a special thanks to Ken Baker at JSC for two years of tireless collection and communication of user needs and his excellent representation of research computing requirements from JSC's perspective. His many hours on the phone with me and others at LARS has greatly magnified our ability to provide good computer services.

* demonstrating the benefits which may be derived when geographically dispersed research centers working on a common problem share a computational environment.

2. Objectives

The objective of the Computer Processing Support Task has been to provide JSC and its associated research community with the environment necessary for the implementation of a shared data processing system for researching remote sensing.

The full implementation of a shared data processing environment would provide the following benefits:

* The opportunity to better mold geographically-dispersed research groups into a more informed and integrated research team;

* A mechanism for efficient transfer of information between research centers, NASA, and other participating government agencies;

* Faster, less redundant software development;

* Faster transfer of newly developed analysis techniques and research results to and from participating research groups;

* Effective, streamlined provision of systems programming, data acquisition, data access and specialized services.

These benefits accrue largely through communications features accessible to all users of the shared system and the elimination of the need to reprogram techniques to be compatible with several different operating systems, data storage schemes, etc. Numerous instances where these benefits have been realized have been demonstrated over the three years of experience with this task.

A second major objective during the past contract year has been to help JSC prepare for the installation of a computer system at JSC, and begin to develop an effective computing network for current and future earth observation research programs. Purdue has attempted to share its fourteen years of experience in remote sensing research computing by training and consulting with JSC support contractors who will be responsible for the implementation, operation and maintenance of the computer system to be installed at JSC. LARS has provided systems and operations training to NASA, IBM and LEMSCO and has consulted with IBM, NASA, ERIM and UCB on the design of a modular, baseline Earth Resources Analysis System (ERSYS) which is to provide a framework for new technique development.

3. Approach

The Computer Processing Support Task has a very broad scope. Detailed discussion of the subtasks in this report is organized under headings of Systems Support, Software and Procedures Support, Data Base Support, Consulting and Training, and Recommendations.

In order for a promising new analysis technique to be shared within the research community, those seeking to use the technique must have:

* Access to the software supporting the technique,

* Access to hardware which supports the software,

* Access to the data required by the technique,

* A technical understanding of the technique, and

* Knowledge of how to operationally use the software and other components of the technique.

To build a suitable environment for the implementation, evaluation and exchange of remote sensing data processing techniques, Purdue has concentrated its efforts on providing access to suitable hardware, software utilities and data. Purdue has also provided consulting and training support to foster user knowledge of the tools available on the shared system. The means for transferring technical understanding of fruitful new techniques and their software implementations are the joint responsibility of the technique's developer, the intended recipient of the new technique and the sponsor. A common Data Processing network can aid in the transfer of knowledge through online documentation and test cases, and through the user-to-user communication facilities the shared system provides.

4. Usage Statistics

Figures A-1 through A-4 show some research community usage statistics for the past two years. CPU time consumed (Figure A-1) is expected to be more than twice as high during 1980 as it was during 1979, just as 1979's usage was double that of 1978. To this point, only one monthly usage figure for 1980 has fallen below the greatest monthly total use for 1979. During May, June, and July of 1980 the new 3031 system at LARS was saturated between the hours of the 8am to 6pm. At the end of July, JSC decided to restrict usage to approximately eighty hours a month which is reflected in the August through October figures.

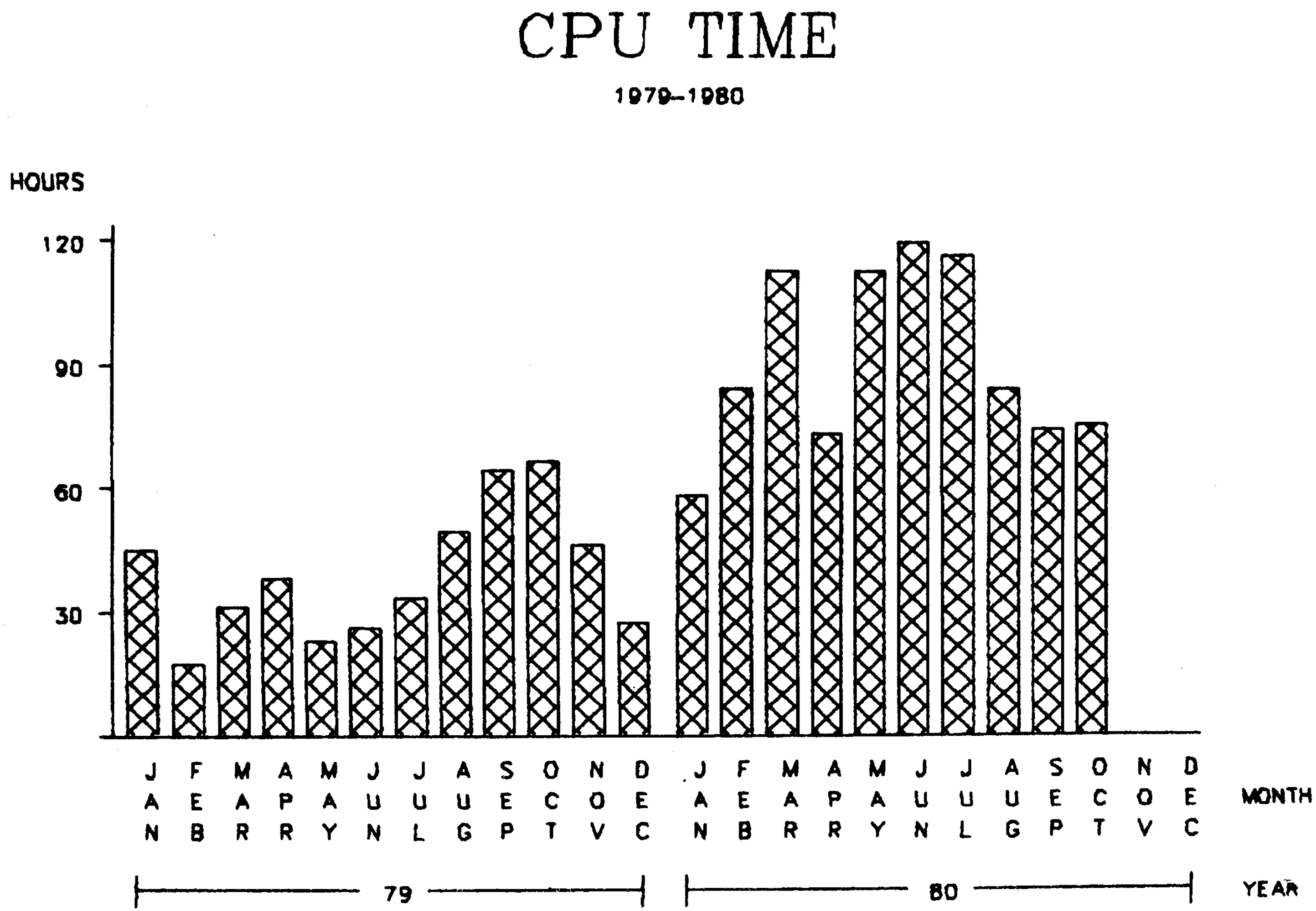


Figure A-1. FY80 CPU Time Utilization.

Figure A-2 displays the time users spend logged on to terminals connected to the system. During 1980 the number of keyboard terminal ports at JSC was increased from 8 to 16 and four ports were added at ERIM. Like CPU consumption, attached time during 1980 will more than doubled that of 1979. The CPU limitations imposed during August have not effected attached time as much as CPU time, this may indicate that an increasing amount of software development is being pursued even though the CPU time available for testing and production has been limited.

The number of virtual machines available to the research community (Figure A-3) continues to climb, more than doubling the number of 1979 users. User Groups identified on the User Group Accounting Report have grown from 15 in 1979 to 23 major groups as of October 1980. This statistic also points to the continued increase in the number of remote sensing investigators utilizing the computer as a major research tool.

The disk storage utilized by users at JSC (Figure A-4) has shown modest growth during 1980. The large increase during October is a result on disk demands of the CMS and VM operating systems being implemented by NASA and IBM in preparation for installation of the computer being acquired by JSC.

The growth in all resource areas is expected to expand as the AgRISTARS program continues. Substantial additional computing requirements will be placed on the machines at Purdue and JSC should significant efforts in Renewable Resource Inventories and Basic Research be initiated, and the benefits of pursuing these programs on a shared system be desired.

5. Accomplishments

During 1980 the accomplishments of the Computer Processing Support Task were:

* Improved and Expanded Systems Access:

    - Installation of four keyboard terminals and a remote job entry station at ERIM;
    - Doubling of the terminal ports to JSC;
    - Implementation of a patch panel and communications organization to provide more reliable access to the facility.

* Development of Additional System Capabilities:

    - Doubling of online direct access data storage;
    - Installation of the Basic Systems Extension Program Package (BSEPP);

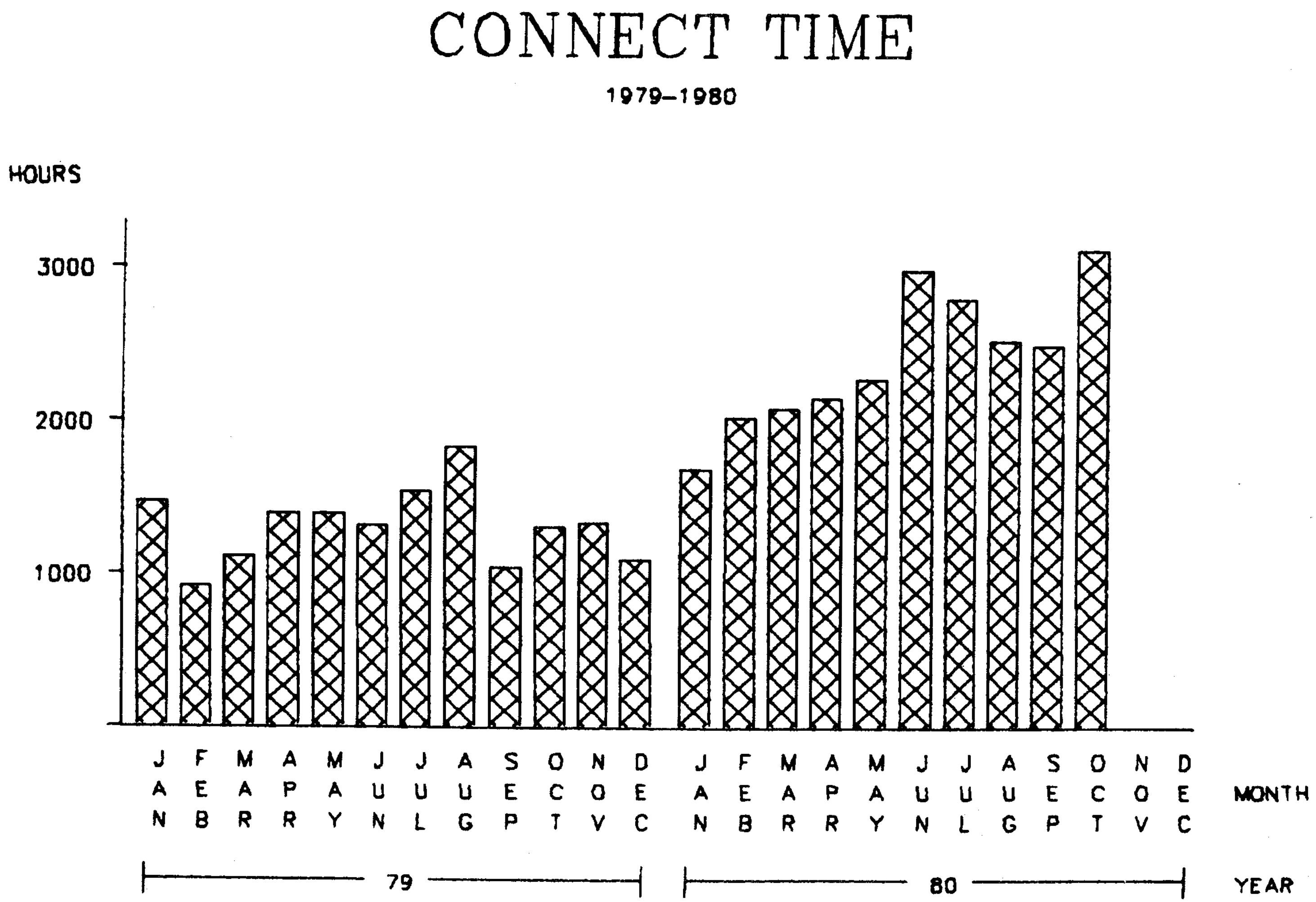


Figure A-2. FY80 Connect Time Utilization

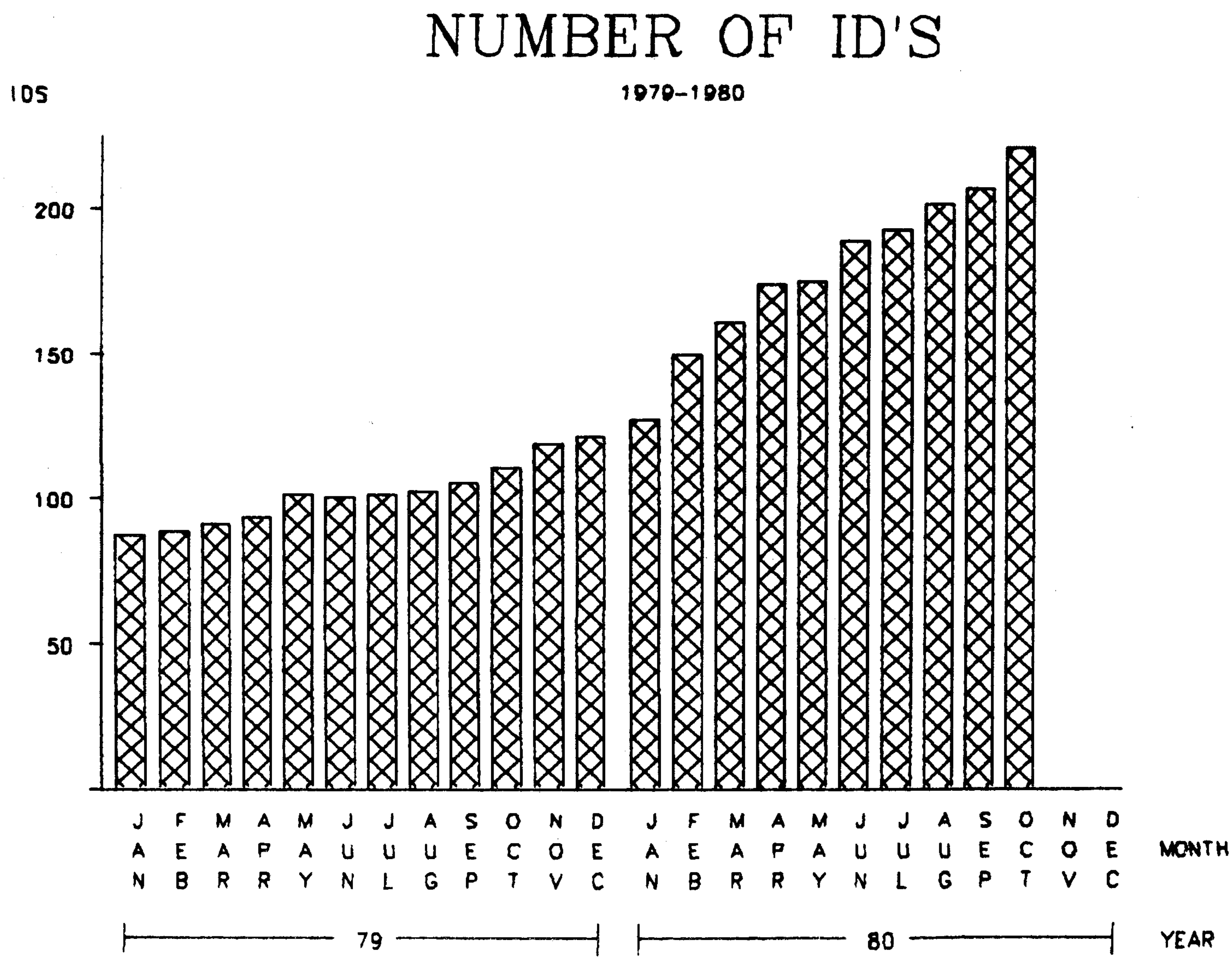


Figure A-3. FY80 Active User ID's

DISK STORAGE

1979-1980

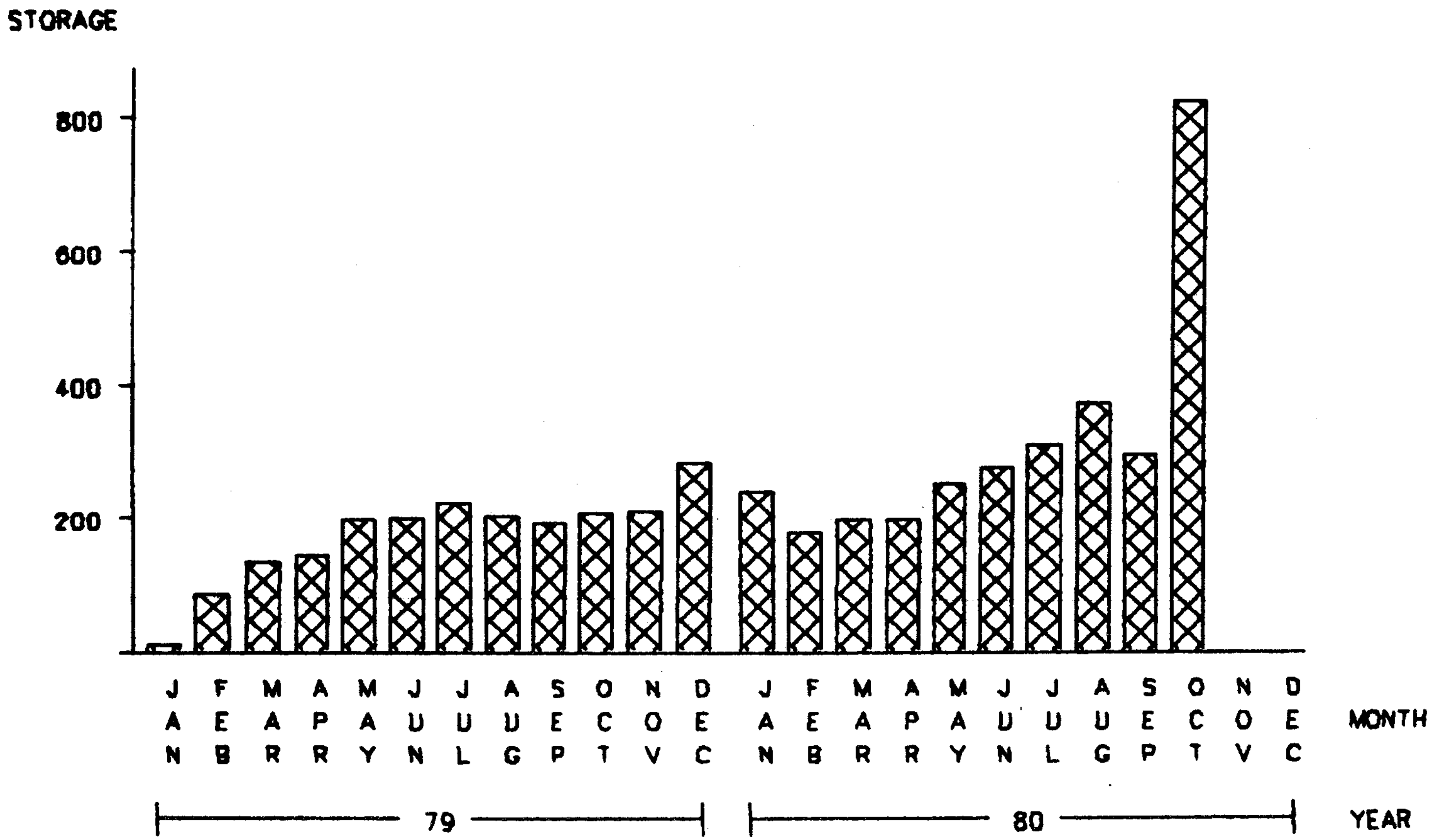


Figure A-4. FY80 Disk Storage Utilization.

- Re-design and expansion of the Batch System;
- Acquisition and installation of the Statistical Analysis System (SAS) and SAS/Graph;
- Development of a CMS version of the Continuous State Modeling Program (CSMP)
- Acquisition of a word processing capability (University of Waterloo SCRIPT);
- Development of a Tektronix driver for the Graphics Control System (GCS).

* Increases in the Quantity, Reliability and Accessibility of Data:
  - Added the crop year 1978 and 1979 image and ground truth data sets to the RT&E data base;
  - Verified data file structures in the RT&E data base;
  - Provided online documentation for all RT&E data base routines;
  - Designed, implemented and documented the Dot Data Base;
  - Reformatted the 1975 historical Synoptic Meteorological Data;
  - Designed a meteorological Data Query System.

* Development of Educational Materials and Provision of Consulting Support:

  - Trained JSC personnel in dot data entry and retrieved procedures;
  - Presented CMS Short Courses at JSC and ERIM;
  - Developed and distributed Tape/Slide modules for CMS self-study;
  - Provided four weeks of visiting consultant time at JSC;
  - Hosted two training and review visits by ERIM personnel, five by JSC personnel.

* Initiation of preparation for an Earth Observations Data System Network:

  - Acquired RSCS Networking for installation when JSC acquires a machine;
  - Reprogrammed all RT&E Data Base routines to function on a two-machine network;
  - Participated in four ERSYS and EODL design reviews.

B. SYSTEMS SUPPORT

by Peter Jobusch

1. CP/CMS Operating System Support

1.1 Installation of VM/370 Release 6 with BSEPP

The major effort in the area of CP/CMS support was the installation in February of Release 6 of VM/370, augmented with the Basic Systems Extension Program Product. In addition to continuing support for extended microcode in the IBM 3031, and support for the new 4341 central processor and 3370 disks which LARS has on order, the system contains significant functional enhancements for the end users of the LARS system.

* A new fair share schedule which more equally allocates system resources among users.

* Enhancements to the CMS file system which virtually eliminate file size restrictions and allow more efficient utilization of disk space.

* Basic support for labeled tapes under CMS.

* The ability to dump and restore spool files to tape.

* An online HELP facility to assist users in using CMS.

During the conversion to Release 6/BSEPP, the decision was made to eliminate as many LARS modifications to VM/370 as possible because each modification impacts our ability to support users and many of them were originally developed to ease the transition from CP-67 to VM/370 and to support CMS 360. Each modification was evaluated using the following criteria:

* The modification must supply an essential feature not currently available, or it must significantly improve an existing feature.

* The modification should be as limited in scope as possible and change as few routines as possible, preferably one.

* System structure such as control block formats should not be altered.

* If a feature is essential, then it should be implemented as a CMS routine or an EXEC file if at all possible.

* Documented features of VM and CMS must continue to operate correctly.

The following table shows the impact of this policy on the major enhancements to VM/370.

Release 5	Release 6
* Prompt for user name at logon and print name on separator pages, accounting cards, etc.	Unchanged, except to shorten name on separator to 8 characters
* YFER available as a synonym CP Spool command	dropped
* 'REALUNIT' option of CP Query command	replaced with CMS QREAL command
* CONTENTS module in CMS	CONTENTS Exec and MEMO option to LISTFILE

Enhancements carried forward with no changes to end users included:

* Additional CP commands (REMOTE, PCHACNT, and PCHARD)

* Accounting data collection

* Support of 8 lines per inch printing as standard (as opposed to 6 lines per inch)

* Local DIAGNOSE instruction codes to support BATCH and Accounting functions

* Enhanced CMS LISTFILE command (the MEMO option was added)

* Enhanced CP TRANSFER command

* SAVE/RESTORE options for certain CP SET variables

* Force sharing of CMS regardless of the alias used to IPL by, and configure the user's VM based upon IPL name

* Enhanced EXEC processor. Functions added include new built-in functions (e.g. &DATE - the current date) and arithmetic operators * (multiplication), / (division), and ! (modules)

1.2 Batch Subsystem Redesign

A second major area of software effort has been the replacement of our current Batch subsystem.

Because a very large percentage of remote sensing investigators work during the day, interactive computer usage is highly concentrated between the hours of 8am and 6pm. A good Batch capability will allow more effective utilization of the computer by making it easy for users to run jobs, which do not require interaction, during off-peak hours. Neither the standard CMS Batch capability provided by IBM, nor the Batch System developed by LARS provide good diagnostics, flexible machine configuration or adequate submission, tracking and cancellation capabilities. LARS has designed and is in the process of implementing a new Batch system which eliminates these inadequacies and adds some additional user-oriented multiple job definition capabilities. Preliminary design work was done during the Spring and in June the principal designer visited JSC to consult with users of the current Batch system and discuss the design of the new system.

Experimental use of the new Batch system began in November, and the project is still on target for July 81 production.

2. EODLS Planning Support

During the course of the past year, LARS personnel were on several occasions called upon to assist in the planning for the EODLS. LARS' experience in managing a computer facility has contributed to this activity in many ways.

2.1 Benchmarking Support

In January LARS personnel participated in the benchmarking of an IBM 4341 processor in Poughkeepsie, New York with personnel from NASA and IBM/FSD. Running independently prepared tests, estimates of performance of this processor relative to the presently installed IBM 3031 processor were made, revealing that the 4341 had between 80% and 120% the power of the 3031 depending on the type of work.

During February IBM/FSD personnel prepared a series of six ASTEP jobs for use in benchmarking systems' performance of projected EODLS workloads. LARS personnel then reviewed the packaging of the benchmark and, after minor corrections and reshipment, NASA/LANGLEY successfully executed the benchmark and returned the results to LARS for analysis.

Following this successful test, copies of the benchmark were prepared and delivered to NASA for distribution to prospective bidders on the EODLS. In August LARS hosted a visit from NASA personnel to analyze the results.

In parallel with the above efforts, LARS personnel also executed the benchmark jobs on the LARS 3031 system in a variety of operating system environments. The jobs were run under the CMS system normally in use at LARS, and under OS/VS1 both in a virtual machine and directly on the 3031 hardware. The results of this study showed that the six jobs used an average of ten percent fewer resources when run under CMS than under either OS/VS1 configuration (and that fewer resources were used when OS/VS1 was run under VM than when OS/VS1 was run in native mode).

2.2 VM/370 Maintenance Training

During the week of October 13-17 four individuals from NASA and IBM/FSD visited LARS to receive hands-on training in the installation and maintenance of the VM/370 operating system. During the course of the week they were able to begin with a 'bare' virtual machine and simulate the process of initially installing VM/370 and applying both scheduled maintenance (as received on a PUT tape) and remedial maintenance (as received from either the IBM Support Center or developed locally). The LARS maintenance philosophy and techniques were contrasted with the 'standard' method, pointing out the improved quality assurance possible with LARS procedures. As a byproduct of this training effort, the virtual version of CP and CMS generated by the students remains available to support software transfer activities.

2.3 Software Transfer Planning

An area of continuing effort all through the year has been the identification of software on the LARS computer which the SR&T research community wishes to have available at both LARS and on EODLS. On numerous occasions, LARS has provided input to requests for information about software at LARS, and we are currently providing on request source codes for many system modules and enhancements to IBM/FSD for evaluation and possible inclusion into the 'vanilla' VM/370 system generated as a byproduct of the VM/370 Maintenance Training activity.

2.4 RSCS Networking

To facilitate the establishment of a peer network between EODLS and LARS, work has begun on the installation of Release 2.0 of the networking version of RSCS. Installation of RSCS Networking will allow users of one of the computers to route output to sites attached to the other machine in the network and provides a convenient means for the transfer of files from a user on one machine to a user on the other.

3. Hardware Report

The configuration of the LARS computer hardware as of 1 December 1980 is shown in Figure B-1.

3.1 Disk Storage Subsystem

In response to a need for increased disk storage on the LARS computer system a CDC 33502 disk storage unit with a 1270 megabyte capacity was installed in April. At that time requirements for the transfer of the ERIPS software to EODLS and the device dependent nature of the dump tapes obtained by IBM/FSD of that software dictated the retention of all three 3330-11 compatable drives. In late November, these requirements had abated and one drive was returned, resulting in the current configuration.

3.2 Communication Subsystem

Communication facilities for ERIM were continually being upgraded throughout the year. In May a dedicated phone line was installed between LARS and ERIM to provide more cost effective communication as ERIM's use of the LARS system increased. In August a DATA 100 was installed at ERIM to supplement the COPE system (which is normally used to connect to the University of Michigan's computer), and in September, LEC personnel visited ERIM to provide instruction in the running of tape transfer jobs. Most recently, the bandwidths of the ERIM line has been redistributed to provide improvement throughout for the DATA 100.

Re-organization of communications equipment at LARS took place over the summer. During this period all communications equipment was relocated and re-organized to provide readier access to the facilities for diagnostic purposes.

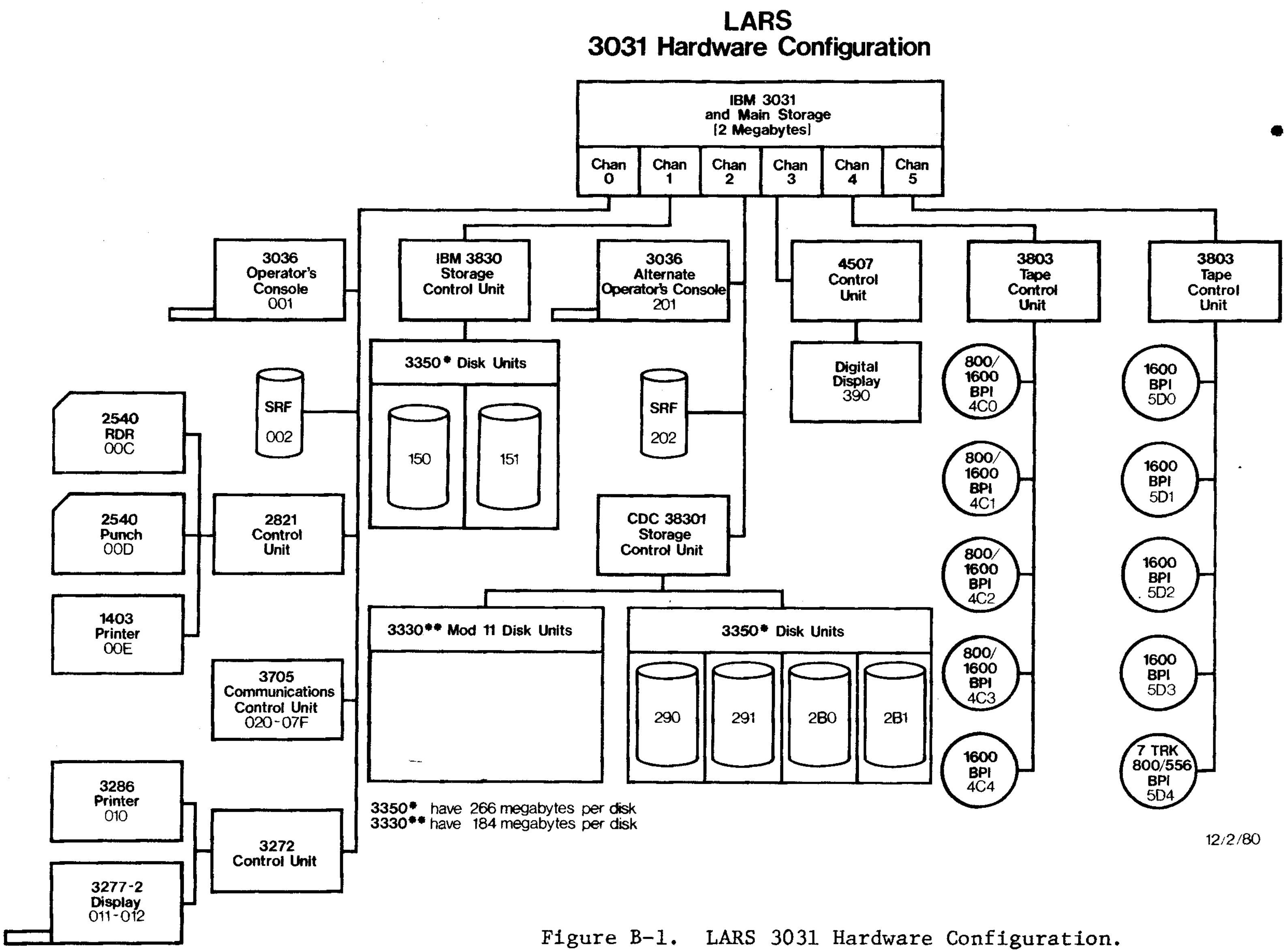


Figure B-1. LARS 3031 Hardware Configuration.

In addition, all communications lines now appear on both analog and digital patch fields, giving us greatly increased problem isolation capabilities and allowing standbys for certain types of critical equipment to be patched in to restore service quickly.

4. Future Plans and Needs

4.1 Planned Changes in Software

Installation of the VM/System Product (VM/SP) is planned for late spring. VM/SP is a new IBM product which combines the features of previous products (including BSEPP), and contains additional functional enhancements. Also, IBM has announced that any further enhancements to VM/370 will be made through VM/SP.

One way the LARS facility may provide valuable service is as a test bed for new systems capabilities. Since it is likely that highly controlled environments and time-critical applications tests will be resident on the machine at JSC, it makes sense to first install and test new systems code on the LARS system, allowing bugs to be uncovered and resolved without endangering production schedules. The installation of VM/SP should serve as a "pilot experiment" for this concept.

RSCS Networking Release 2.0 will be installed to facilitate the networking of EODLS and LARS computer systems.

This system will allow the routine of spool files to and from remote job entry stations on any network node. Additional software should be generated to assist users in inter-systems communications (e.g. network news files, MAIL, message and trouble reporting capabilities). File transfer facilities should also be developed.

The new BATCH system will be made available for parallel runs in March 1981 and will become the production system with the removal of the old system in July.

4.2 Planned Changes in Hardware

The IBM 4341 central processor which LARS has had on order since its announcement in April 1979 is scheduled to be installed in September 1981. This processor offers almost as much processing power as the current 3031 at greatly reduced cost.

6250 bpi tape processing capabilities are being studied, and a recommendation to upgrade the tape subsystem was made in September. Installation of the upgraded system could occur during 1981.

Replacement of the 3705 front end processor with a COMTEN 3670 will occur in early 1981. The new processor offers exciting capabilities not available on the 3705 which expand LARS capability to interface with other networks (such as CTC TELENET) and extended support of TTY compatible terminals (with features such as XON/XOFF flow control protocol support).

Based on the selection of a COMTEN communications controller at JSC, it is also possible to allow node selection and line balancing between the JSC and LARS facilities to be handled by the communications controllers. LARS will investigate alternatives and make recommendations on the network implementation to JSC during February, 1981.

4.3 Equipment Needs

Communications test equipment is needed to improve our diagnostic capabilities relative to our communications network. By monitoring digital signals with equipment similar to the Atlantic Research 3500 we would have the ability to diagnose line protocal difficulties. A small oscilloscope rack mounted with the communications equipment would allow diagnosis of the analog communications circuits. As the sophistication of our network increases, so does our need for test equipment such as this.

5. Recommendations

As we head into an era of networking EODLS and LARS computer systems, the need for close technical coordination of these two facilities needs to be recognized. It is therefore recommended that two working groups be established to coordinate the systems at a technical level. One group should deal with issues directly related to the network itself (e.g. the coordination of the telecommunications links) and a separate group should deal with coordination of non-networking related issues (e.g. definition of a common subset of software to be maintained on both systems).

C. SOFTWARE AND PROCEDURES SUPPORT

by S. Kay Hunt

During the past contract year the accomplishments in software development maintained a balance between work on planned tasks and responsiveness to immediate user needs. In providing software services, this balance must be carefully maintained in order to avoid either being entirely reactionary with no overall plan to provide direction or having work planned in such detail that resources are not available to react to user needs. Figure C-1 summarizes the software package now resident on the LARS computer.

1. Software Package Reports

For supported software packages on the LARS system, documentation in the form of User's Manuals should be available in terminal areas at each research site. LARS tests software packages with the installation of each new package update and after installation of new operating system releases. An individual at LARS has been assigned the responsibility for the maintenance and test of each package. This individual is able to serve as a consultant in the characteristics and users of the packages. S. Kay Hunt has the overall responsibility for Applications Software at LARS and all problems or questions may be addressed to her.

1.1 Statistical Software

SAS, the Statistical Analysis System was installed during the second quarter of FY80. It provides a wide range of statistical procedures, an ability to read complex files, and report-writing capabilities.

Use of SAS, the Statistical Analysis System, grew rapidly throughout the year. During the past few months execution of SAS programs accounted for over ten percent of the total CPU usage. This growth was aided by increasing knowledge of SAS and by training sessions at LARS and JSC. A LARS statistician, Carol Jobusch, attended the SAS

Users Group annual conference in San Antonio, Texas in February and a short course on Writing SAS Procedures in October, in order to assist users with SAS and help the conversion of FORTRAN programs into SAS procedures.

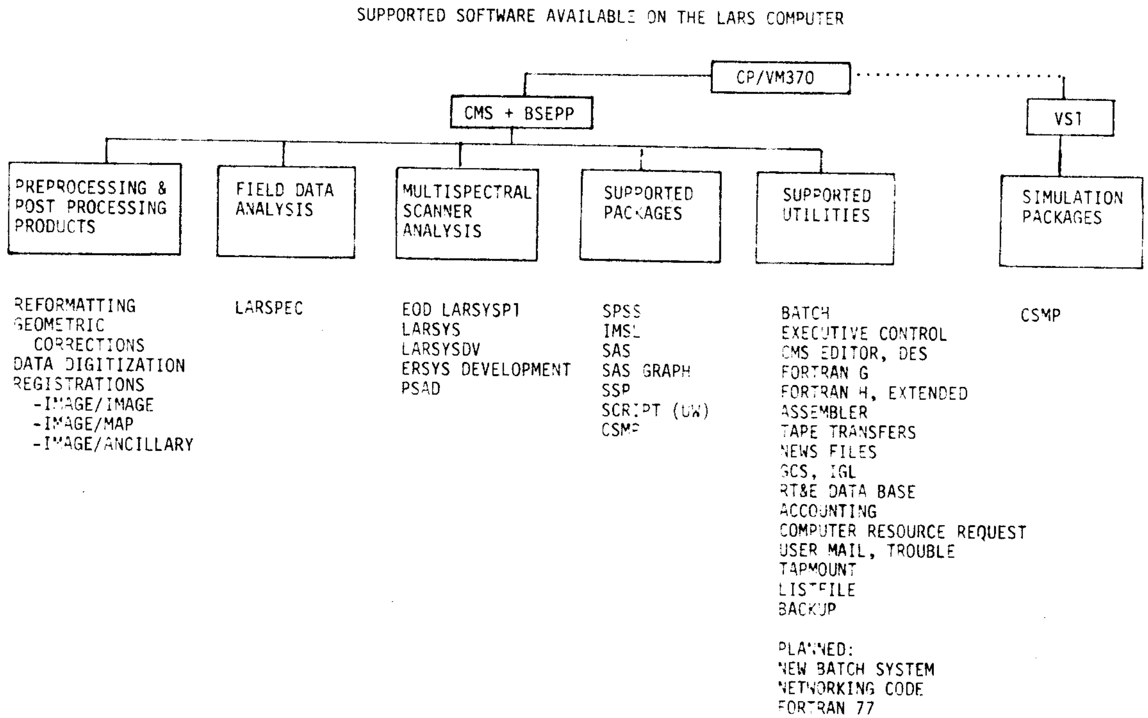


Figure C-1. Supported Software Available on the LARS Computer

Release 79.3 of SAS was installed in May, and upgraded to release 79.4 with the addition of SAS/GRAPH in June. SAS/GRAPH is an interactive computer graphics system for producing high-resolution plots, charts, and graphs - including three-dimensional response surface plots. Figure C-2 showing use of statistical systems at LARS was reproduced from output of a hardcopy unit attached to LARS' Tektronix 4054 screen graphics terminal. SAS/GRAPH output may also be generated on the Tektronix 4003 at JSC.

Use of SPSS, the Statistical Package for Social Sciences, has decreased as users have learned about the power and versatility of SAS. The current version of SPSS will be retained, but future releases will not be acquired.

Edition 7 of the International Mathematical and Statistical Library (IMSL) was updated in March 1980 with approximately 40 routines being upgraded. Subsequently, Edition 8 of IMSL was received and installed September 5, 1980. Many documentation and programming errors were corrected by these new editions. JSC is currently paying the rental for IMSL and plans to transfer the program product to the JSC machine when it is installed. LARS will be acquiring an IMSL lease when the transfer occurs so it will be available on both systems in the future.

Purdue/LARS has a statistical consultant to assist users of SAS, SPSS, and IMSL routines. Development or conversion of special purpose statistical programs may also be requested. If a user encounters a problem of a statistical nature, he may contact Carol Jobusch at LARS.

1.2 CSMP

The Continuous State Modeling Program (CSMP) is an IBM program product used for the simulation of continuous systems. As provided by IBM, CSMP does not run under the CMS operating system. JSC uses CSMP for modeling soil moisture. In order to make CSMP available, LARS initially installed a VS1 operating system running under VM. This system is not as interactive as CMS, requiring CSMP users to learn both VS1 and CMS and requiring the LARS Systems and Operating Groups to maintain and operate two systems. In order to reduce the LARS workload and make CSMP easier to use, LARS modified this package to run under CMS.

1.3 SCRIPT

SCRIPT, the University of Waterloo's text processing software, was installed on the LARS computer. On September 19, 1980 the latest upgrade was installed. Available with the latest release of SCRIPT are

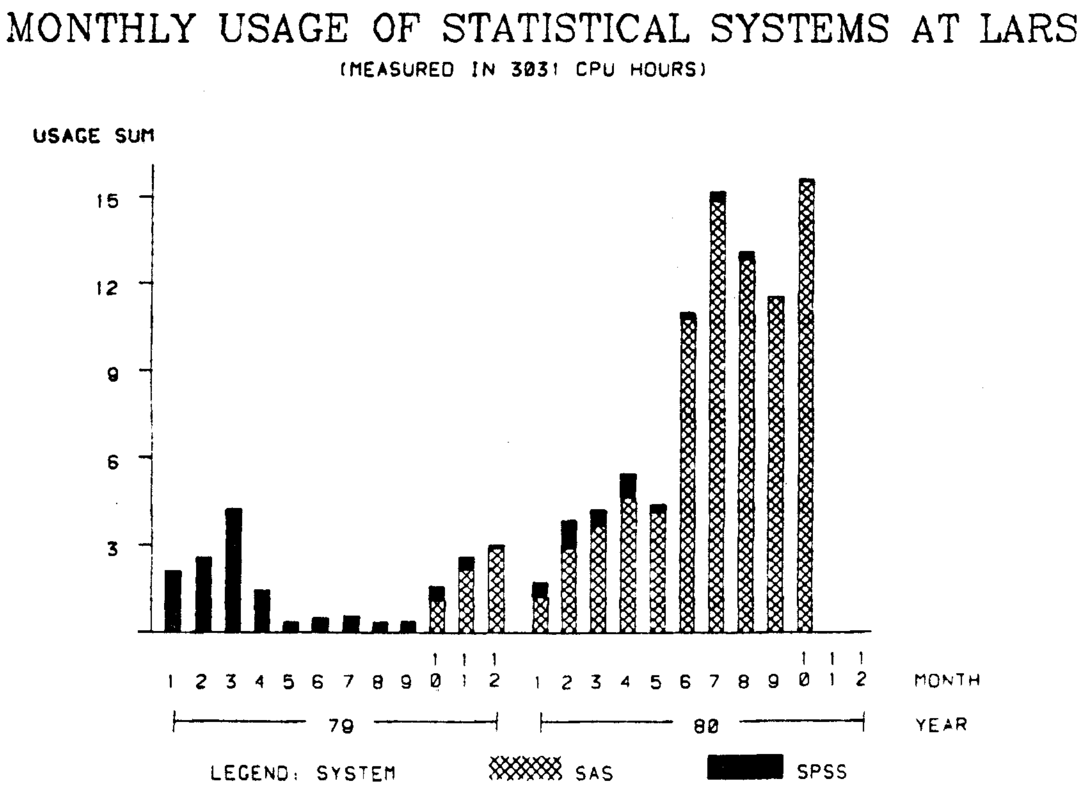


Figure C-2. An Example of SAS/GRAPH: Monthly Usage of Statistical Systems at LARS two powerful sets of SCRIPT macro instructions called SYSPUB and SYSPAPER to facilitate the preparation of documents.

To facilitate the use of SCRIPT the Display Edit System (DES-formerly known as EDGAR) has been installed at LARS. Use of DES requires a device plug compatible with an IBM 327X terminal. Using DES and 327X-class terminals a user may edit files at two to three times the rate he could achieve using the CMS editor and TTY terminals. JSC is currently planning to acquire 327X-compatible terminals with its computer acquisition. It is recommended that these terminals be attachable to both the system at JSC and the system at LARS to facilitate the editing of files on both systems.

1.4 GCS

GCS, the USMA Graphics Compatibility System, is available in two versions, a two-dimensional version, and a three-dimensional version. GCS provides a simplified, easy to learn and easy to use approach to computer graphics while simultaneously providing a powerful tool which the sophisticated programmer may use for demanding highly interactive applications.

The graphics display on the following devices is supported by the GCS systems implemented at LARS;

*Line Printers
*Alphanumeric terminals
*Varian Electrostatic Printer/Plotter
*DECwriter II terminals with Graphics Board
*Tektronix 4054 graphics terminal

The Tektronix 4003 at JSC will support a subset of the GCS commands and may be used for many graphics applications.

In general, the 2-D GCS is a proper subset of the 3-D GCS version meaning that all elements of the 2-D GCS are contained within the 3-D version.

1.5 LARSPEC

LARSPEC, the LARS Spectral Analysis System, was upgraded over the past year. In January, 1980, the new LARSPEC User's Manual was made available for distribution. The DECwriter and Tektronix 4054 were included as valid graphics devices. Expanded punch capabilities for the data selection processor were implemented. LARSPEC has three overall capabilities: Printing and punching statistics of wavelength bands; Plotting spectral information; Printing and punching identification record information.

Further upgrades to the LARSPEC system to improve user capabilities are continually in progress. Planned upgrades for 1981 include: the addition of software to recalibrate data on certain FSS data tapes and possibly changing the current method of storing the description of wavelength data in general, inclusion of the capability to plot response vs. wavelength data which have differing wavelength increments on a single graph, the implementation of data search software to enable a user to quickly locate all data in a particular parameter category, updating of the LARSPEC User's Manual to reflect recent changes, the division of the three LARSPEC processors into modules which would only be loaded into the main core as needed and the implementation of a band averaging capability which would allow the user to apply a weighting function for averaging specified wavelength bands.

2. Software Utility Reports

In order to facilitate and manage operations involving multiple sites, LARS has written a few specialized utilities. Tape transfer software has been written to pass the contents of tapes to and from LARS and remote sites with appropriately equipped Data 100 terminals. The User Group Accounting System was implemented to allow JSC administrators to flexibly define sub-account (User Groups) and monitor resource consumption for User Groups, Branches and in aggregate. The Resource Request System was initiated to provide a "paper trail" for all computer resource requests (ids, disks, tapes, etc.) and to keep track of the status of all requests. This system replaced phone calls as the primary means to request directory changes.

2.1 Tape Transfer

Operation of the tape transfer software (TAPTRAN) has been running smoothly. Software modifications were made to refine messages and information to the user. In August, 1980, the capability was expanded to include remote ports other than JSC.

2.2 Accounting

The JSC user group accounting system was re-written in January. The new system is more flexible and easier to maintain. A SAS readable accounting file was added to the system in August. In order to monitor disk allocation, a disk accounting program was produced in May and reports are generated monthly.

2.3 Resource Request System

The resource request system was completely restructured to allow users to view request forms. The Resource Request System is designed to enable a remote site coordinator to request resources of the LARS computer facility. User ids may be established or changed, tapes may be purchased, and ring-in authorization may be established for tapes through the resource request system. All of the work involved with this system is to be done online and will likewise be reported to the appropriate personnel through online methods.

3. Recommendations

The implementation of a two-node network will require more complex data transfer and management tasks if users are to successfully utilize the capabilites available throughout the network. As more research sites gain access to the system additional complexities will result. The substantial benefits of sharing a data processing environment will be realized if the tools exist to manage usage, quickly react to user needs, and provide users access to the data and software they need in a timely and easy fashion.

3.1 Establish a Reasonable Division of Software between Systems

In order to have compatible and usable systems some software such as compilers, editors, basic utilites, etc. should be common to both JSC and LARS. For the sake of efficiency and cost, the two systems and the people required to run and maintain them should not be totally redundant. Global roles should be established for the two systems and a reasonable division of software and maintenance responsibilities should be established.

For example, since LARS is responsible for a major portion of the field data base, it may make sense for LARS to be the only system where LARSPEC and the field data library are housed. SAS and IMSL are of very general use and should be available on both computer systems. The installation of these packages on both systems could be performed by one group--as could consulting support for them. The installation of a large data base management system requiring a large number of specialized experts might better be accomplished at JSC. While one group might be given global operational responsibility for specific operating systems (VM, CMS, VS1), both sites should have people capable of installing and upgrading these systems.

3.2 Develop New Procedure for Allocating Computer Resources

As a matter of practice, a single user will find it easiest to perform each of his major tasks using only one computer system. There are instances when users will have separate tasks, one of which will be done on the system at Purdue, the other at JSC. It will be of great benefit to such users if similar or identical procedures are used for requesting computer resources such as tapes, ids and disk space; and if similar accounting procedures and allocation rules apply. Because of the rate structure required by the charter for LARS systems services and because charging policies are often strongly related to facility management philosophies, it may prove to be impractical or impossible to adopt similar accounting procedures for the transfer of funds. This fact does not prevent the creation of artificial allocation and accounting variables for the measurement of resource consumption on both systems. It is recommended that a single individual at each research site enter all computer id, tape and disk requests for both the JSC and LARS systems from information provided on a single user form. Figure C-3 is a candidate for such a form. It is also suggested that a single accounting system be developed for user group accounting on both systems using a set of mutually agreed to measurement units.

3.3 Develop Expanded Data Transfer Capabilities

The Tape Transfer software which is currently available is highly specialized, allowing transfers of tape data to and from appropriately equipped Data 100 terminals and LARS. When JSC installs its computer system, software should be developed to allow direct transfers of data between both tape and disk media on each system. RSCS Networking provides the framework for several approaches to user oriented file transfer capabilities. File transfer code is greatly required for users to be able to move personal data sets between systems.

3.4 Develop a Software Library Policy

CMS provides a library oriented file management system which may be exploited for software transfer and configuring virtual machines to have specific software resources available. The concept is implemented by placing routines which complement each other on a single mini-disk (library) and providing general read-access to the disk to the user community. Write-access is reserved to those ids used to maintain the library. A single CMS virtual machine has the capability to concurrently access in excess of 20 library disks.

It is recommended that all software used by the JSC research community be classed as either personal software, part of an unsupported program library, or part of a supported program library (See Figure C-4). Software should be placed in a supported program library to provide a reliable and stable software environment insuring the availability of basic capabilities and/or to insure the integrity of major applications tests. Supported software should have substantial test procedures available to insure its validity. These test procedures should be exercised after each major system upgrade. A supported software library should be under configuration and change control, and errors encountered in code should be fixed as soon as possible.

USERID ACTION FORM

1. ( ) NEW ( ) RENEWAL ( ) CHANGE ( ) DELETE ID
2. DATE OF REQUEST: ____________________
3. REQUESTOR NAME: ____________________
4. COMPUTER SYSTEM ( ) EODL ( ) LARS ( ) BOTH
5. ID ASSIGNED (8 CHARACTERS OR LESS): ____________________
6. PASSWORD REQUESTED (8 CHARACTERS OR LESS): ____________________
7. NUMBER OF CPU HOURS ALLOWED (OPTIONAL): ____________________
8. SPECIAL FACILITIES:
   A. DISK SIZE (MEG BYTES): ____________________
   B. MAX CORE SIZE (960K DEFAULT): ____________________
   C. READ PASSWORD: ____________________
   D. WRITE PASSWORD (OPTIONAL): ____________________
   E. SPECIAL DISK LINKS (OPTIONAL): ____________________
9. DESCRIPTION OF WORK TO BE DONE (OPTIONAL): ____________________
10. PERSON(S) WHO WILL BE USING THIS ID: ____________________
11. COMMENTS: ____________________

____
12. REQUESTOR PHONE NUMBER: ____________________ EXT.: __________

----------OPERATIONS USE ONLY----------

13. ID ASSIGNED: ____________________
14. ORGANIZATION (IPM, LEC, NASA, ETC.): ____________________
15. PROJECT BRANCH (SF3, SF4, SF5, ETC.): ____________________
16. USER GROUP (FIELD RESEARCH, SUPPORTING RESEARCH, ETC.): __________

____
17. APPROVALS:

X ____________________
NASA TECH. MONITOR
(REQUIRED FOR NON-NASA PERSONNEL)

X ____________________
BRANCH CHIEF OR
BRANCH LARS MANAGER

X ____________________
SF6 APPROVAL

Figure C-3. Strawman Userid Action Form for the EODSN Network

	UNSUPPORTED LIBRARIES	SUPPORTED LIBRARIES
PURPOSE	TO PROVIDE A WIDE RANGE OF FLEXIBLE NEW CAPABILITIES TO THE RESEARCH COMMUNITY FOR EXPERIMENTAL USE	TO PROVIDE A RELIABLE AND STABLE SOFTWARE ENVIRONMENT, INSURING THE AVAILABILITY OF BASIC CAPABILITIES AND THE INTEGRITY OF MAJOR APPLICATIONS TESTS
DOCUMENTATION	REQUIRES SOME DOCUMENTATION	MUST BE FULLY DOCUMENTED
CONTROL	UNDER CONTROL OF DEVELOPER	UNDER CONFIGURATION AND CHANGE CONTROL
ERROR CORRECTION	ERRORS DOCUMENTED, FIXED AT DEVELOPERS DISCRETION	ERROR DISCOVERING AUTOMATICALLY INITIATES CORRECTED PROCEDURES UNDER CHANGE CONTROL
TEST PROCEDURES	EXAMPLE RUN AVAILABLE	SUBSTANTIAL TEST PROCEDURES EXERCISED AT EACH SYSTEM CHANGE TO INSURE VALIDITY

Figure C-4. Suggested Software Libraries Organization

All software in a supported software library should be fully documented and documentation should be distributed to each major research site utilizing the network. The operating system, program products such as SAS and IMSL, major operational capabilities such as registration, and major applications systems used for pilot experiments are examples of software systems which should be supported program libraries.

Unsupported program libraries should be composed of software systems which have wide or general interest but which are not essential, time-critical elements of a major NASA program. Routines and systems should be placed in unsupported program libraries in order to expose the research community to a wide range of flexible capabilities for experimental use. Sufficient documentation should be provided for exploratory program use. Rather than being under change control, unsupported libraries should be under the control of the developer or implementor. When errors are uncovered in unsupported software they should be reported to the developer who will document the existence of the errors on the unsupported library disk. Errors will be fixed at the discretion of the developer. A simple input and output example should be provided on the unsupported library disk for purposes of instruction and as a minimal test procedure. Software systems which are under development (such as LARSPEC or future pilot test systems), initial implementations of systems such as VICAR or OLPARS, new programming languages, and user developed utility routines of general interest should be placed in unsupported libraries.

A detailed set of software library standards and procedures should be established and provided to all groups who might become responsible for either supported or unsupported program libraries. JSC should identify a software librarian whose duty would be to maintain a current list of libraries and their capabilities. This librarian would also have the responsibility for insuring that all supported and unsupported libraries meet the standards established. Information about changes to both supported and unsupported libraries should be published in a research community newsletter.

D. DATA BASE SUPPORT

by Luke Kraemer

To support the research needs of the SR&T community over the past year, LARS has acquired additional data sets for the RT&E Data Base. Improving the existing data base as well as implementing new capabilities have also highlighted this past fiscal year. LARS has also developed an economical method for inverting large MET data sets and has begun processing some of these sets. In preparation for the installation of the EOD computer in Building 17, we have re-coded existing software so that it will be compatable on both the EOD and LARS systems. Data base and data base software transfer activities will begin soon. Figure D-1 summarizes the data sets available on the LARS system.

1. RT&E Data Base

Most of the work done on the RT&E Data Base this year was devoted to adding new data sets to the library, adding new capabilities to existing software, and modifying software to work on both systems. Other tasks involved re-organizing both the disk and the tape libraries and writing some new data base software.

1.1 Receive, Verify, and Enter Data Sets

The following data sets were received at LARS for inclusion into the RT&E Data Base:

Crop Year 1978 Corn/Soybean
Crop Year 1978 Small Grains
Crop Year 1978 Ground Truth
Crop Year 1979 US & Foreign

The addition of these data sets has enabled LARS users access to approximately 60,000 LACIE sized images (Figure D-2). Before this data is released to the user community though, the data which is stored on high density magnetic tapes must be assigned tape slot locations

DATA AVAILABLE ON THE LARS SYSTEM

(ENTRIES)

SPECTROMETER/ RADIOMETER	METEOROLOGICAL	MULTISPECTRAL. SCANNER	GROUND TRUTH	OTHER
-FIELD MEASUREMENTS DATA BASE (160,000) SIX YEARS OF DATA COLLECTED BY FSAS, FSS EXOTECH MODEL 20 C&D EXOTECH MODEL 100 OVER 10 TEST SITES	-DAILY/MONTHLY SUMMARIES OCT. '77-PRESENT -HISTORICAL SYNOPTIC JAN. '74-DEC. '75	-LACIE PHASE I (2522) -LACIE PHASE II (10595) -LACIE PHASE III (20525) -CROP YEAR 78 (15723) -CROP YEAR 79 (8908) -LARSYS -AIRCRAFT & LANDSAT (3800)	-LACIE PHASE II (49) -LACIE PHASE III (123) -CROP YEAR 78 (169)	-U.S. COUNTY BOUNDARIES -DOT DATA BASE
PLANNED: MULTIBAND RADIOMETER	PLANNED: -HISTORICAL SYNOPTIC JAN. '66-SEPT. '77 -DATSAVE (AIR FORCE) 1963-1977 -CO-OP 1878-1978	PLANNED: -RADAR/MICROWAVE -CROP YEAR 80 IMAGE		

Figure D-1. Data Available on the LARS System

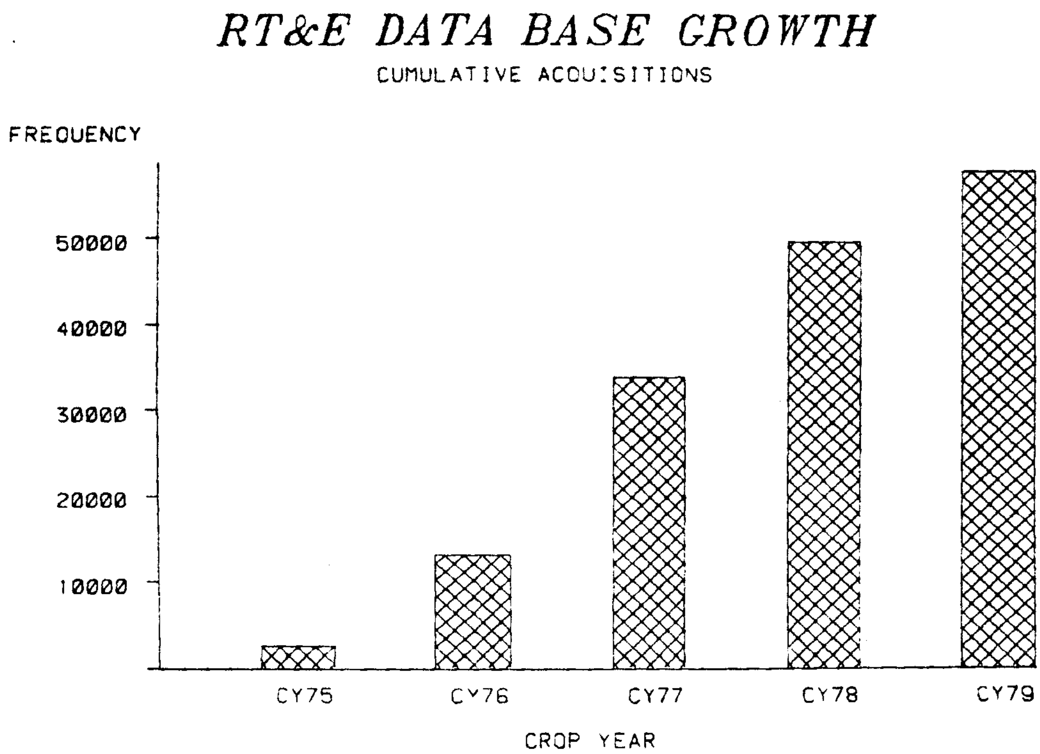


Figure D-2. Historical Growth of RT&E Data Base in the LARS library. Then each tape is tested for its format and readability. All tapes that fail the tape tests must be regenerated. Fortunately, the failure rate has dropped significantly over previous years due to the improved tape testing procedures developed at Building 17. After a complete and correct data set has been received both the RT&E Data Base and the Subset Data Base must be updated to include the new information. The data base update is performed during a night time shift thereby affecting users for only one midnight to 8 am period. The availability of the new data set is then announced in both SRTNEWS and SCANLINES.

In the case of the Crop Year 1979 data, LARS was required to do some additional processing of the image data. During the tape verification procedure, a number of erroneous image acquisition dates were identified. After notifying NASA of this problem LARS then created new tapes with the proper dates. A copy of these tapes were shipped to JSC for their data base.

1.2 Software Improvements

As more users started to access the RT&E Data Base, additional features were requested for incorporation into the system. In most cases, only minor modifications were required, however, some major updates and new coding were also asked for. Major tasks included writing a query subroutine SEGALL which will return all disk directory stored information for an acquisition. A program to report all acquisition dates for a crop year was also written. This program was used in identifying the incorrect 1979 data. New features added to the SUBSET system include the ability to query data by its JSC tape number and storage location (JSC, LARS, or BOTH), and to query data for any existing ground truth observations.

A major task started during the final quarter of this fiscal year involved identifying and modifying all RT&E Data Base software that is to operate on both computer systems. The software is required to identify which system the user is logged into without any user interaction. The software on both systems also has to be identical, therefore, an identification file will be present in both data base systems which will signal to the software which machine is being used. Routines that locate data and mount tapes will query this file and then proceed processing based on the initial system query.

An additional program which can only be used on a Tektronix terminal is STATESET. This program enables a user to perform a SUBSET query and then displays the results of the query as a map on the screen of the device. For every acquisition that meets the conditions specified, a rectangle is placed on the map corresponding to the center latitude and longitude of the image. Figure D-3 is a display of all 1979 segments in Iowa.

1979 SEGMENTS IN IOWA

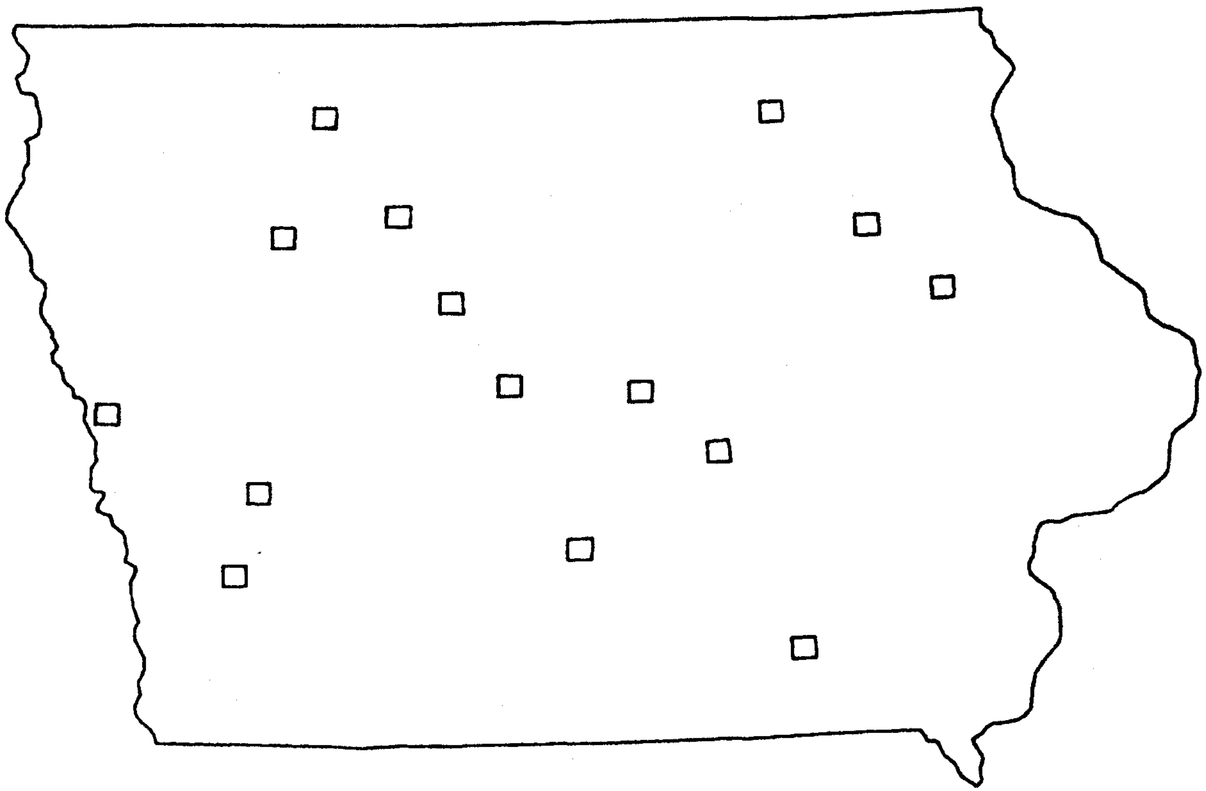


Figure D-3. Geographical Representation of 1979 Segments in Iowa

1.3 Data Base Re-organization

During the second quarter all crop year 1977 tapes with duplicate imagery elsewhere in the library were identified. As instructed by NASA, these tapes were removed from the data base. Portions of the tape data base were then moved to fill in the resulting free space. All users were notified one month in advance.

In order to greatly reduce the total CPU time required to update the disk data base, a new file called the Segment Index (Appendix I, Pg. 6) was built into the system. This file serves as a fast entry into the data base and also allows new segments to be added to the tail of the Segment List rather than inserting a new entry into a sorted list. Maintenance of a sorted list becmes expensive with a lengthy file of segment entries. This new Segment Index was incorporated into the system while at the same time RT&E users were not required to modify any existing applications software. The results of this modified design indicates that user queries are slightly faster and that data base updates are significantly faster.

New data items now stored in the disk data base include the JSC tape and file that an image is stored on, the archive tape and file where that same image is stored, a flag which indicates which sites the data is stored at, and another field which indicates the crop year an image represents (Appendix I, Pg. 7 through 17). New items that can be queried by SUBSET include the JSC tape number, the data site location, and the ground truth availability flag.

2. Dot Data Base

The LARS Dot Inventory System is a set of programs which aids analysts in the digitization of ground truth information. This system combines the basic concepts from Tim White's original Dot System, input from JSC analysts and programmers, and the programming and data base expertise of the LARS support team. A detailed user and programmer manual was written and a complete course detailing the system was presented at JSC. At NASA's request, some modifications were made to the Dot System. Two Dot files were created at JSC and entered into the data base as a test of the system.

2.1 Design and Implement Dot Data Base

During the first quarter LARS investigated different approaches in the design of a Dot Ground Inventory System. After studying Tim White's first Dot System and interviewing JSC analysts and programmers, a scheme was developed and a brief description was presented at the first quarterly review. The system is broken into two modes - the user mode and the data manager mode. In the user mode, the analyst has complete freedom to query and extract information from the data base. The analyst then creates and/or edits a dot file which is stored on his disk. When the file is complete and correct it is transferred to the data base manager. The manager verifies that the file is valid and enters it into the data base.

Programming began in March and a more detailed explanation of the system was presented at JSC during the April visiting consultant trip. Programming was completed in May and the system was tested at LARS. Following the training course in June, modifications were requested. Some of the major modifications requested included adding additional identification fields in the header record, providing the capability for multiple crop names lists, and accepting different grid patterns.

2.2 Documentation and Training

As the system was being coded, all routines were thoroughly documented. A complete Dot System User's Manual was also written and entered into the LARS computer in SCRIPT. To obtain a copy of the manual log onto the LARS computer and enter the following commands:

```
GETDISK JSCDISK 19A
DOTHELP
```

The manual will then be printed out on your line printer (Appendix I). As upgrades are made to the system, the manual is also updated and notification of any major changes are reported in SRTNEWS and SCANLINES.

An intensive two day course on the Dot System was presented at JSC in June. The course was broken into two sections - a user/analyst section and a programmers section. The user/analyst section consisted of an introduction to the system, a detailed description of the system, a question and answer period, and then a hands-on exercise. Topics covered in the programmer's class included an introduction to the system, a brief review of the user software, a detailed discussion of the physical and logical layout of the data base, an introduction to all callable routines, and a question and answer period.

3. MET Weather Data Base

NASA has archived a large collection of historic MET synoptic data. This data covers world weather readings from 1966 to 1975. The data set is not organized in a very convenient manner and LARS has been requested to convert this data to a specified format. A very economical approach was devised to reorganize the weather reports and processing has begun on the data. In addition to the historic data, LARS also receives current daily MET data from NOAA.

3.1 Design and Implement Algorithm

In March the LARS 3A Computer Support staff was requested to investigate methods of re-sorting large weather data sets. A small sample set was sent to Purdue for testing. The data, which is stored on tape, is sorted on the date and time keys. Each one-year collection of reports requires 24 magnetic tapes; therefore, to query one MET reporting station for an entire year, one would have to search every record on all 24 tapes. There are approximately 11 million reports per year. The desired sort keys in order are - world region block number, reporting station number, date, and time.

During the LARS visiting consultant trip in April a meeting was held with representatives from NOAA, NASA and LEMSCO. Results of the one month test case were discussed and plans for future testing were set forth. Requirements for the next test included creating data tapes in the desired order, writing the tapes in a machine independent format, building a directory for the data base, and, most importantly, writing a fast and inexpensive algorithm. The test case was decided to be 1975. Processing of this data set was suspended until sufficient funding was secured. Work continued in August and the optimal algorithm was completed the following month. The final reorganization would require two steps - a sort and then a merge. The sorting stage was started in October and required three weeks. Due to the excessive demands of the algorithm on the CPU, sorting could only be run during the third shift. The merging phase was started in November and required only two days to complete.

The designing of a disk directory and data accessing system has begun and should be completed by December. The system will have some query capabilities as well as routines that will mount and position data tapes and read desired weather reports. A meeting will be called for interested NASA, NOAA, and LEMSCO personnel concerning the results of this large scale test.

3.2 Receive Data and Verify

Complete synoptic yearly data sets require 24 storage tapes or two tapes per month. When a data set arrives at LARS the first operation performed is a complete tape copy of the set. Due to the age and the poor quality of the tapes, extreme care must be taken during this step.

In a majority of the tapes, the plastic inner hub ring has either been cracked or partially broken off. If too much of the hub has been broken off then the tape is unreadable. Fortunately, all the tapes have been readable so far. The next step in verification is to read each monthly group and generate a listing of station reporting frequencies. Also, a partial data verification check is performed with this program. After a monthly group passes all tape checks, it is ready for sorting. The 1975 data set was received and verified in May. The 1974 data set was received in November and verification has begun.

In addition to receiving historic weather data, current daily weather summaries are also shipped to LARS from the National Climatic Center. The information is sent on magnetic tape at approximately two month intervals. When a shipment arrives, the tapes are checked for readability and format and then labelled and placed in the LARS library. Researchers at JSC are then notified.

4. Data Base Documentation and Training

Documentation and training is an integral component of the LARS computer support task. Documentation is prepared for new products and presented at regular intervals at JSC.

4.1 Documentation

All user callable data base routines are fully documented internally. User documentation also exists in the form of LARS abstracts. These abstracts have been entered onto the system in SCRIPT and will be available to all users in the near future. A JSCDISK will be initialized and will hold this documentation. The Dot Systems User Manual will also be stored in this library. The Dot Manual is currently on the JSCDISK 19A and is available to all users.

Documentation concerning the transfer of all data base materials is currently being prepared. A document listing all data base routines, their dependents, and core sizes has already been delivered. Documentation for the FSD transfer and testing team will begin soon.

4.2 Training

LARS visiting consultants travel to JSC at regular intervals. Typically during these trips, mini-courses on existing and new data base products are presented. During the LARS CMS Short Course in December an entire module was devoted to the RT&E Data Base. In June an intensive two day course was presented on the Dot Ground Inventory System. This course consisted of lectures, question and answer sessions, and a hands-on exercise. A tape/slide mini-course on the data base was also prepared in July. This module was distributed along with other CMS information modules to user groups on the LARS system.

In preparation for the transfer of data base routines from LARS to JSC, a course outline for the training of operations staff at JSC has been delivered. This course will introduce the data base system and then have a training period detailing the maintenance of the system. This course will be presented at the LARS installation.

5. Plans and Recommendations

5.1 RT&E and Dot Data Bases

* Add new Image, Ground Truth, and Dot Data Sets
* Develop plan for maintaining identical data bases on both systems
* Continue programming support
* Link to Weather Data Base
* Transfer Data Base to JSC Installation
* Determine political locations of Segments
* Correct erroneous image header records
* Incorporate Field Observations Data

As future data sets are completed at JSC, copies will be generated and shipped to LARS. Following verification, these data sets will be included in their proper data bases. We recommend that identical data bases be maintained on both systems. This plan would insure both flexibility for users and reliability for the data base administrators. A procedure for updating both system's data bases will have to be developed as soon as the networking plans have been finalized. Programming support will continue as new routines and data base capabilities are requested. We recommend that links between the RT&E Data Base and the MET Data Base be developed and maintained. These links should prove to be valuable to researchers who need to analyze both image and weather data. When notified by NASA, the RT&E Data Base and all its associated software will be transferred to Houston. When this data base system is installed, a LARS computer support person should be present to consult and assist in this operation. LARS also recommends that the RT&E Data Base capabilities be enhanced and that data quality be insured. Two areas for immediate improvement would be inclusion of political boundaries and incorporation of the Field Observations Data Base. Data items known to be in error are sun elevation and azimuth, satellite identifier, and scene frame number.

5.2 Weather Data Base

* Complete query system
* Process additional yearly data
* Create daily and monthly summaries from synoptic data
* Develop basic report generators
* Incorporate Tektronix line drawing capabilities
* Process new data sets as requested

Plans for an on-line query system have been completed and implementation will begin soon. Users will have the capability to query the data base for block and station number, date, and latitude and longitude. The query function will return the tape and file location of the desired data. As additional yearly synoptic data sets are received and processed, the disk based system will be updated. Daily and monthly summaries can be generated from this synoptic data. In many applications these summaries are more valuable than the intensive synoptic weather reports. Location of the summaries will also be stored in the disk directory. Some basic report generators that will display information concerning a station or group of stations should also be developed. These reports would aid in the selection of MET stations. We also recommend that the line drawing capabilities of the Tektronix terminals be utilized in searching the location of stations as well as displaying contour maps and overlays. Finally, new weather data sets are scheduled for receipt during this year. LARS will verify and perform any necessary processing on the data tapes. The disk directory data base will be modified to accept information on the new data versions.

5.3 Documentation & Training

* Complete software documentation for transfer
* Transfer user documentation
* Train JSC personnel in Data Base usage and maintenance
* Contribute articles to SCANLINES and SRTNEWS
* Continue visiting consultant trips

In preparation for the transfer of the LARS RT&E Data Base and associated software to the new EOD computer, all programs and routines will be documented. Documentation will be in three forms: internal (program comments), operational, and user. Internal documentation will consist of the program function, variable definitions, and explanation of coding. Operational documentation will outline operating procedures for data base maintenance programs. This documentation is intended for the data base manager only. User documentation will exist for all user callable programs and routines. This documentation will be available to all users and will be maintained on an accessible disk. JSC personnel designated as members of the data base maintenance team will be trained in the usage and maintenance of the RT&E Data Base. Training should take place at both installations. As new capabilities are implemented and new data sets are received, LARS will submit articles to SCANLINES and SRTNEWS. LARS will also present seminars on a regular basis to the JSC community on assorted RT&E Data Base topics.

E. CONSULTING AND TRAINING

by S. Kay Hunt

This section deals with the personnel services portion of the Computer Processing Support Task. Training provides the background users require as a prerequisite to system use; consulting provides the expertise which helps to make the users more efficient and effective.

1. Visiting Consultant Reports

Problems encountered by remote users are frequently hard to fully understand or appreciate when described over the phone, especially when they require an understanding of the exact sequence of events, operating condition, etc. to diagnose. The immediate availability of an expert to answer questions and suggest approaches to software design and implementation is valuable. It is also extremely valuable to gain insight into a remote user's operating environment and resource needs. For these reasons LARS supplies visiting consultants to JSC for three days to a week once every two or three months as requested. Visiting Consultant trips also afford those responsible for certain design or development tasks the opportunity to discuss these tasks with interested parties at JSC.

Carol Jobusch served as visiting consultant in February, 1980. She was able to provide general user consultation as well as present a seminar on Release 6 of CMS, provide SAS, SAS/DBMS consulting and also discuss weather data needs.

Luke Kraemer was at JSC in April 1980 to provide general consulting on problems encountered by LARS computer users, to discuss weather data, and to review procedures for requesting computer resources.

Pete Jobusch, Luke Kraemer, and Tom Wilson travelled to JSC in June 1980 to consult with JSC personnel. A portion of the trip was spent in presenting some sections of the LARS CMS short course. Pete made ERSYS implementation recommendations to IBM, Luke presented the Dot Ground Inventory System and Tom briefed users on the new batch system design.

Luke Kraemer visited JSC in November 1980 to discuss weather reformating results and transfer of the data base and to provide general user consulting.

2. CMS Short Course Report

During the week of December 10-14, 1979 a CMS short course was presented at JSC. Five instructors were available during various portions of the week: Larry Biehl, Luke Kraemer, Peter Jobusch, Susan Schwingendorf, and Carol Jobusch. For this course, the topics were divided into 19 one hour modules, with prerequisites defined for each module. There were modules for the beginner, intermediate and advanced user. Appendix II contains the course outline and schedule. A similar course was presented to ERIM during May of 1980 in preparation for ERIM's coding of the Corn/Soybeans analysis software.

3. Tape Slide Module

Tape slide modules of each of the CMS short course presentations were made available to the users at JSC. The units varied in length from 30 to 60 minutes and are intended for individuals or small groups of people who need to use the Purdue/LARS computer before the next major training course is organized. The mateial presented on each cassette was taken from CMS courses given at JSC, ERIM and LARS during the past year. The following tape slide units are available to computer users:

- Introduction to VM370
- CMS I
- Virtual Machine Concepts
- CMS II
- EDIT I
- EXEC I
- Programming I
- ProgrammingII
- EDIT II
- EXEC II
- CP Commands
- CMS III
- RT&E Data Bases
- IMSL

Future modules are being planned for SAS, SCRIPT, and BATCH.

4. DBMS, Operating System Review

During the December 1979 through February 1980 time frame personnel from LARS, NASA, UCB and ERIM participated in a series of meetings at JSC to critique proposals being made by IBM/FSD for the selection of the primary operating system and data base manager for EODLS/ERSYS. During these meetings the desirability of using VM/370 as the primary operating system vis'a'vis OS/VS1 was stressed, and the feasibility was shown through demonstrating the availability of support for this environment by several major data base management systems. It was recommended to JSC that a DBMS system be selected which could support networked or relational file structures. Based on discussions with several DBMS suppliers several Data Base Management Systems were identified as being able to serve CMS users from a VS1 Data Management virtual machine.

5. ERSYS

Personnel from LARS travelled to JSC in December, January and April to attend Design Reviews of the ERSYS System as presented by IBM/FSD. As a result of these meetings and of presentation materials from other reviews, LARS personnel submitted 10 Review Item Disposition (RID) Forms. LARS also hosted three visits by IBM personnel to directly review and consult on ERSYS design questions, and two visits from ERIM personnel to review and provide input on ERIM's design of a Corn/Soybeans Analysis System.

LARS is hopeful that the ERSYS system will provide a truly modular system framework, suitable as a development aid for new software analysis techniques by all members of the research community.

6. SRTNEWS

A feature available on the LARS computer which enhances communication between users at JSC and LARS is the SRTNEWS facility. This is a news facility whereby a user may request all current news as previously input to be printed at the user terminal or on the line printer for review. During the past year 36 news articles were disseminated through SRTNEWS. Newsworthy information may be submitted to Dee Dee Dexter at LARS or Jimmy Gilbert at JSC.

7. SCANLINES

SCANLINES is a means to disseminate information among users. It is a LARS news bulletin produced and printed on a monthly basis and containing current news items of interest to a wide community of users. Over 100 articles relating to the use of the computer, new software packages, new features and capabilities were published during the last year.

F. RECOMMENDATIONS

by J. L. Kast

1. Plan and Implement a two-node Earth Observation Data Systems Network

To meet the expanding computational needs of AgRISTARS and to prepare for the Renewable Resource Inventory and Basic Research Programs, JSC is pursuing the acquisition of a computer system. This system is not anticipated to have sufficient capacity to supply the computational needs of JSC's entire research community, including the universities and ERIM. The computer JSC acquires should be joined with the computer at LARS (which is currently the primary computing facility used by JSC and LARS for remote sensing research) to provide a data processing network for the Earth Observations Research and Development Community (See Figure F-1). Development of such a network would:

* provide a significant portion of the capacity needed for the AgRISTARS, Renewable Resource Inventory and Basic Research Programs,

* allow the benefits which Purdue and JSC have experienced by sharing data processing environments to be extended to additional elements in Earth Observations Research and Applications Programs,

* capitalize on the experience and equipment in place at Purdue, while developing a similar facility under direct NASA control,

* make possible the development of a highly flexible research facility and system testbed at one network node and a highly controlled system integration and test facility at the other node, while maintaining operating system compatibility on both nodes,

* make possible access to software developed on either machine,

* allow overload computing to be shared by both machines,

* provide a backup capability to both systems in the event of machine failure at either site.

Before such a network can be fully developed, a number of issues must be addressed. A task force consisting of a Planning Group and a Working Group should be established for the purpose of strategically and technically solving the procedural, administrative and technical problems associated with the implementation of the two-node network.

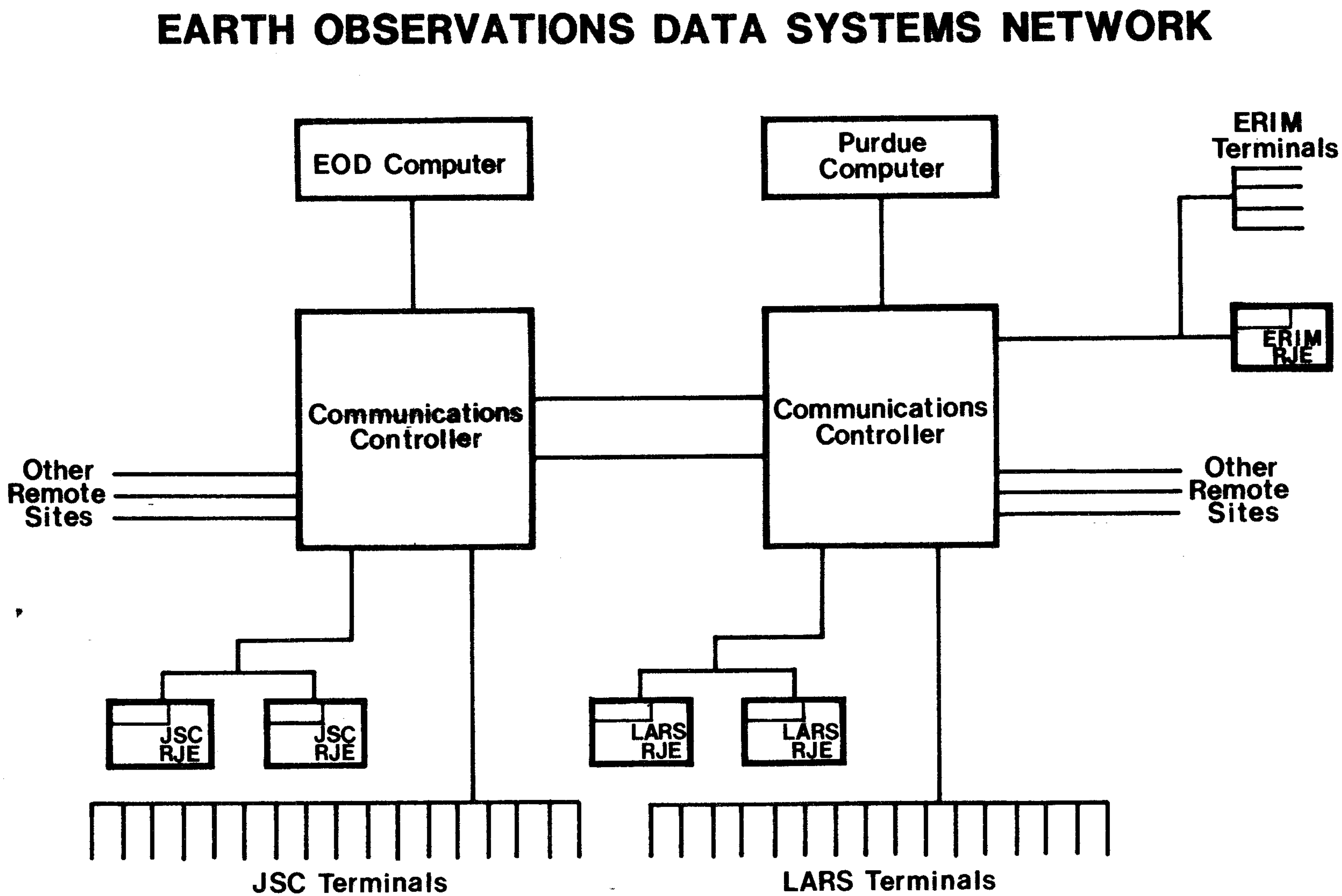


Figure F-1. Earth Observations Data Systems Network

The recommendations of these groups should be presented to NASA management as a long-term plan. Upon approval by NASA, recommendations of the task force should be implemented by the appropriate elements within the research community.

Access to the network should be provided to all contractors expending more than $200,000 per year and to contractors whose work needs to be immediately available for testing, inclusion in analysis systems, or for exchange with other research groups. In addition to reducing costs through less software development redundancy, elimination of software conversion costs, and the reduction in the number of computer facilities that NASA dollars must support, sharing of a computational environment by all major contractors provides for a long-term increase in productivity by reducing the time needed for technique transfers, making a full range of analysis capabilities available to all research sites, providing a mechanism for research-community-wide communication, and eliminating system differences as deterents to understanding newly developed software techniques.

2. Establish a Research Community Baseline System

New technique development and technique exchange could be further speeded if a modular, baseline research system framework were developed which would accomodate most remote sensing analysis software. Such a system structure should allow analysis system components developed at different sites to be easily compared or combined into a hybird analysis system without extensive reprogramming. The development of such a system structure requires substantial, thorough documentation and communication if it is to be successful. It is unclear at this point whether or not ERSYS is to serve as a baseline system. The programming conventions and systems structure of ERSYS need to be documented and understood by all groups who are developing candidate techniques for Exploratory Experiments and Pilot Tests. Until this happens, software developed by different contractors will be incompatible and difficult to compare. It also will be necessary to document the procedure for establishing developmental systems which utilize the framework of a baseline system.

3. Define, Implement and Enforce Reasonable Standards

In order to expedite technology transfer within the research community and between the research and applications communities, certain programming, documentation and software delivery standards should be established. One benefit of a modular baseline and analysis system is that its framework will allow the addition or replacement of analysis processors without requiring major system rewrites. Therefore, if all research sites would deliver new techniques in the form of software compatible with the modular system, virtually no additional programming would be required to conduct tests comparing new techniques with each other and/or with the baseline system. Making use of common programming conventions, documentation and delivery standards would greatly enhance the ability of researchers at various sites to understand and utilize the software developed at other sites. JSC should allocate funds and assign members of the research community to a task force responsible for the creation of such standards. Standards would be beneficial, even if a shared computational environment were unavailable.

The task force on standards should include in its effort the consideration of the following topics:

* an identification of those programming languages and software packages which will be acceptable for purposes of software delivery. Use of new languages and special packages are acceptable for concept research, but pilot experiments should rely on thoroughly tested systems.

* recommendations of minimal coding, documentation and test procedure requirements for software delivery.

* recommendation of methods of altering and upgrading the standards utilized.

* means to document and enforce standards.

* criteria for placing newly developed routines and/or software systems under configuration control.

* the development of a research community unsupported software library for software not under configuration control.

After the standards and procedures have been identified, an individual at JSC should be assigned the responsibility of helping technique developers conform with the standards and for disseminating information to potential users of newly developed capabilities.

4. Investigate ways to Improve Systems Communications

For a number of geographically dispersed research and development organizations to function effectively as informed components of a well coordinated program, development of a wide range of communication capabilities is essential. Since the data processing network will be utilized by virtually every element in the research community, it is especially important that means for good communication of new capabilities, procedures, standards, requirements and hardware be established. In addition to phone calls, quarterly technical interchanges and special trips, it is recommended that JSC and its research community:

* Publish in a bi-weekly or monthly newsletter information about and examples of new system capabilities, operations schedules, frequently encountered problems, training course contents and schedules, significant experiment results, and other information of general interest. JSC may elect to develop its own newsletter or ask LARS to expand SCANLINES (which is already distributed to most user organizations and which currently carries the suggested news about the LARS computer system). Whichever aproach is selected, its success will depend on the willingness of those responsible for significant projects to write a short article outlining what has been done. Quite frequently motivation from a "site editor" would greatly enhance the quality and amount of material exchanged through this medium.

* Establish and maintain a Systems News file accessible to all users which documents very recent system upgrades and alterations. SRTNEWS on the LARS system is a example of such a news file. One individual at each site should be responsible for collecting and placing news articles on the system. If a "site editor" is established, he would be a good choice for maintenance of a News file.

* Systems and utilities under configuration control as well as routines which are candidates for general usage should be supported with online documentation in the form of "Help" files. Appendix III provides an example of a Help file on the LARS-supplied MAIL command.

* Implement online memo sending (MAIL) and problem reporting (TROUBLE) systems as a means for user-to-user and user-to-supplier communication. Both the MAIL and TROUBLE commands must be modified to work in the two system network.

* All supported software (where supported software is defined as software under configuration control) should minimally be supported by "Help" files, User's Manuals and Systems Manuals. At least one copy of each manual should be available in the terminal area at each research site.

* For major software packages and basic operating system capabilities, tape/slide modules or video tapes should be developed and distributed to each major research site. These materials should be sufficient to introduce new users to a basic capability and should be applied to self-study courses for new users. The tape/slide modules developed as an introduction to the LARS system and a selected subset of its systems software, user utilities, and software packages serve as examples of the kind of materials which should be developed.

* A modular ensemble of training courses for users of the network should be formulated, including:

1 Access and use of the computer network,

2 Access, use and augmentation of the baseline analysis system,

3 Use of the special network utilities (MAIL, News files, Batch, Data Bases, file transfers, unsupported program library, etc.),

4 Programming conventions for software (supported languages, baseline system compatibility, universal format capability, commenting practices, transferability, etc.),

5 Documentation standards,

6 Standard algorithm evaluation and test procedures,

7 Procedures for requesting data, and

8 Others, as needed.

The CMS short course which has been presented annually by LARS at JSC is a first step towards developing courses for number 1 and number 3 above.

5. Establish Mechanisms to Secure User Input and Feedback

As research needs change, more advanced equipment and software systems become available, and users come to more fully know the capabilities and limitations of the tools available, the data processing system should evolve. One severe problem which should be avoided at all costs is the isolation of the system's users from those providing the data system's capabilities. When tunnel vision occurs, a great deal of waste, dissatisfaction and frustration are generated. Common symptoms of this problem are claims by the user community that the systems people are unresponsive, waste time on minor problems, ignore user needs, are not preparing adequately for upcoming experiments and don't understand their jobs. To this the systems people respond that the user groups are not providing adequate requirements with sufficient lead-times, expect miracles, don't know what they want or need, and that they (the systems group) have met all requirements in a timely manner and gone to extremes to be responsive. Both groups are frequently right about the other: the underlying problem is a mutual lack of understanding. By himself a system user is frequently inadequate to fully document his systems requirements for a task because he is not used to thinking in dimensions of system resources and certainly limited in his ability to wisely select from alternative systems approaches. On the other hand, the systems person doesn't even know what the problems and projects are about.

The only way to avoid this problem is for users and systems suppliers to talk directly with each other when requirements are being generated and to stay in contact through system implementation, operation and upgrades. At LARS, several approaches to this problem are pursued. From time to time information exchange sessions are held when plans are reviewed in detail and hypothetical scenarios discussed in joint user/supplier groups. Suggestions are actively solicited by system designers during these meetings.

A second approach to the problem involves consultants from the computer facility being stationed at the Flex1 terminal area and visiting consultants periodically sent to remote sites to review user problems and plans. User problems are then reviewed in bimonthly meetings of the LARS System Services staff and personnel are assigned to any significant problems which remain unresolved.

Perhaps the most effective approach to user/systems communication at LARS has been the development of a Systems Services Representative program. In addition to their systems maintenance and development responsibilities, most computer specialists at LARS are assigned as representatives to one or two LARS projects. As a representative, the computer specialist is to establish an agency relationship between the LARS systems services and the major research project. This means the System Services representative is to serve as an advocate for a project within System Services, helping the project's data processing needs to be adequately communicated to the right people. He also represents the System Services to the project.

As an advocate for the project, he should:

- Know the project's objectives, major milestones, current status, needs and problems.

- Know the people involved in the project and their general project responsibilities.

- Know what the project budget is.

- Know what system services the project is expected to consume by when.

- Help make those who provide services to the project aware of the services needed and the schedule.

- See that problems the project encounters with System Services are solved in the quickest manner with minimal project impact.

- Make appropriate System Services Managers aware of serious user criticism or concerns.

As a representative of System Services, he should:

- Make projects aware of all efforts made to solve their problems.

- Never publicly malign individuals, groups, or projects of System Services.

- Help project leaders and users become aware of products, procedures, resources and capabilities that may benefit their project.

- Help prepare realistic Services Budgets for future projects.

- Stay in contact with the project manager (at least monthly).

- Make sure the Services budget is spent effectively and not grossly overspent.

- Know and report plans for future projects.

- Help identify other potential projects and investigators.

- Seek help and advice in procuring new clients.

- Seek help and advice in new product needs.

This program has been very successful because each project manager has a good contact in System Services who understands his problems, because systems people gain insight into what remote sensing is about and into research and development project problem, and because the representative can more frequently present project needs to his peers more effectively than the project manager can. A similar arrangement is recommended for the user groups and research projects of the entire JSC user community.

At sites without a computer systems group, site experts should be identified to handle systems related problems in cooperation with the JSC and LARS systems groups. The site expert should be a technical specialist, familiar with computer programming as well as the technical aspects of typical remote sensing data processing software. At most sites it would probably be convenient for this person to double as a programmer. Site experts will need to spend two or more one-week periods a year becoming intimately familiar with the processing network, new software utilities which have been added, and in reviewing problems, etc. with his peers at other sites.

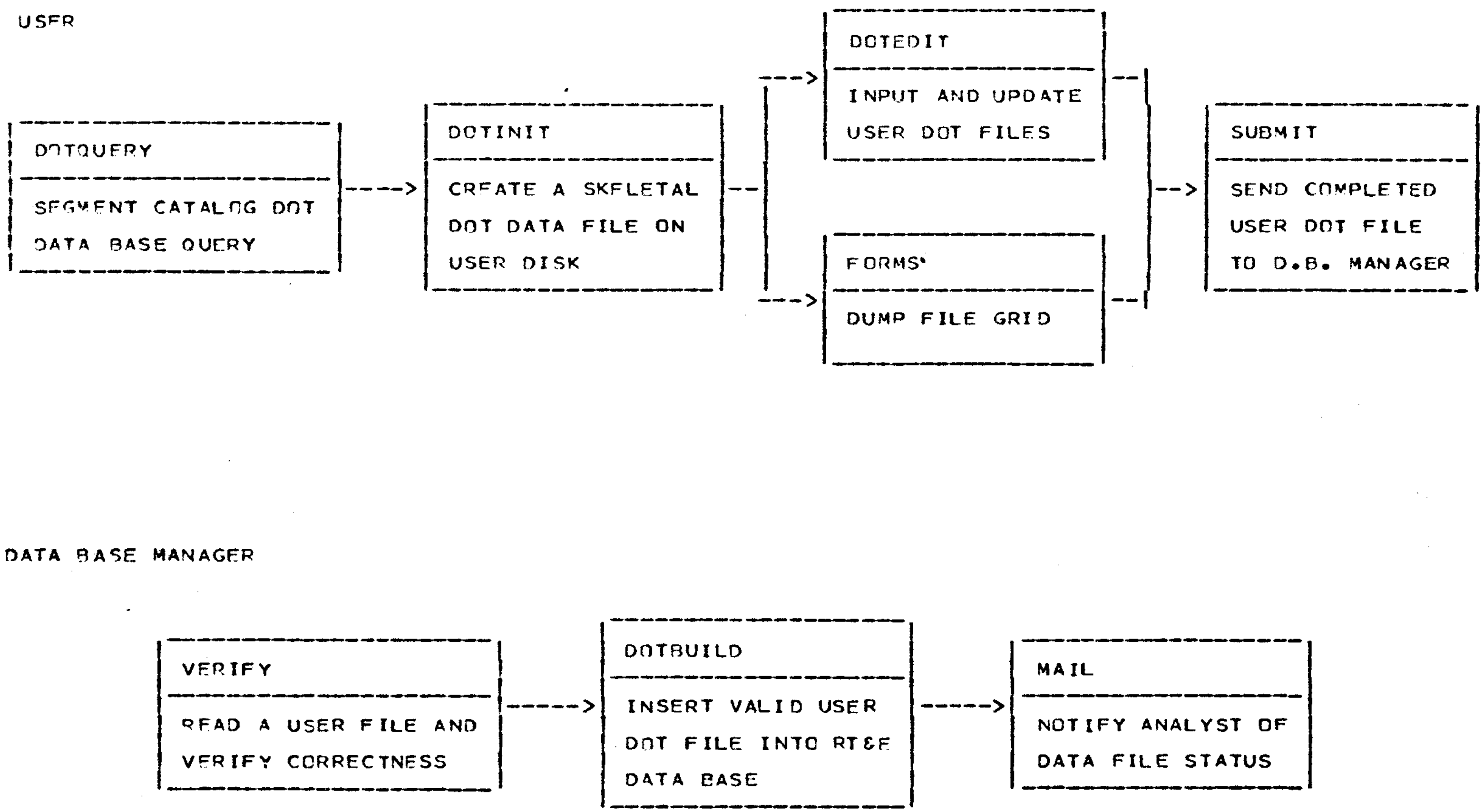
Each local site should also identify a person to serve as a computational resources manager. This person will be responsible for interfacing with the network in order to secure and maintain the computational resources necessary to support users at his local site. Depending upon the amount of activity at the local site, the resources manager and the site expert may be the same person.

Appendix I

LARS DOT GROUND INVENTORY

SYSTEM USER'S MANUAL

DOT INVENTORY SYSTEM FLOW



DOT INVENTORY SYSTEM FLOW

THE LARS DOT INVENTORY SYSTEM IS A SET OF PROGRAMS WHICH AIDS ANALYSTS IN THE DIGITIZATION OF GROUND TRUTH INFORMATION. THE SYSTEM IS BROKEN DOWN INTO TWO PHASES; THE DATA ENTRY OR 'USER' PHASE AND THE SEGMENT CATALOG UPDATE OR 'DATA BASE MANAGER' PHASE. THE USER PHASE IS AVAILABLE TO VALID JSC USERID'S ON THE LARS COMPUTER SYSTEM; WHEREAS, THE DATA BASE MANAGER PHASE WILL ONLY BE AVAILABLE TO THE RT&E DATA BASE MANAGEMENT TEAM. MANAGEMENT OF THIS DATA BASE IS CURRENTLY BEING PROVIDED BY LARS PERSONNEL.

ENTRY INTO THE THE DOT SYSTEM IS INITIATED BY THE DOT ANALYST. AFTER LOGGING INTO THE LARS COMPUTER AND GAINING ACCESS TO THE DATA BASE DISK, THE ANALYST SHOULD THEN QUERY (DOTQUERY) THE DATA BASE FOR THE SEGMENT NUMBER AND CROP YEAR DESIRED. NEXT, THE USER MUST CREATE A DOT FILE ON HIS DISK BY RUNNING THE 'DOTINIT' PROGRAM. IF THE SEGMENT OF INTEREST IS PRESENT IN THE DATA BASE, THEN A COPY OF THAT FILE CAN BE STORED ON THE USER'S DISK USING DOTINIT. IF AN ENTIRELY NEW DOT DATA FILE IS REQUIRED, THEN A SKELETAL FILE CAN BE SAVED ON THE USER DISK BY ALSO USING DOTINIT. THIS CMS DISK FILE IS ACTUALLY A CONCATENATION OF THREE LOGICAL RECORDS: A DOT LABEL TABLE RECORD, A GRID POINTS LIST RECORD, AND AN OFFGRID LIST RECORD. A FORMAT DESCRIPTION OF THESE RECORDS ARE IN ATTACHMENT A. NOTE THAT MORE THAN ONE FILE MAY BE STORED ON A USER DISK AND THAT MORE THAN ONE FILE WITH THE SAME SEGMENT NUMBER AND CROP YEAR MAY BE PRESENT IN THE RT&E DATA BASE.

ONCE THE ANALYST HAS CREATED HIS PERSONAL DOT DATA FILE ON HIS DISK, THEN HE CAN ENTER OR ALTER GROUND TRUTH INFORMATION USING 'DOTEDIT'. THIS EDITOR IS SIMILAR IN NATURE TO THE CMS EDITOR BUT ALSO PROVIDES SOME NEW SPECIALIZED COMMANDS FOR THE DOT FILES. AFTER A DOT FILE HAS BEEN MODIFIED, A DATA FILE DISPLAY ROUTINE (FORMS) CAN BE RUN. THIS PROGRAM WILL DUMP, IN A USER READABLE FORMAT, A USER SPECIFIED DOT FILE ON A LINE PRINTER.

UPON COMPLETION OF THE ITERATIVE EDITING AND PRINTING PROCESS, THE ANALYST DOT DATA FILE IS THEN READY FOR INCLUSION IN THE RT&E DATA BASE IF SO DESIRED. SOFTWARE TO RETRIEVE DOT FILES FROM EITHER THE DATA BASE OR USER DISK IS AVAILABLE. IF THE NEW FILE IS TO BE ENTERED INTO THE SEGMENT

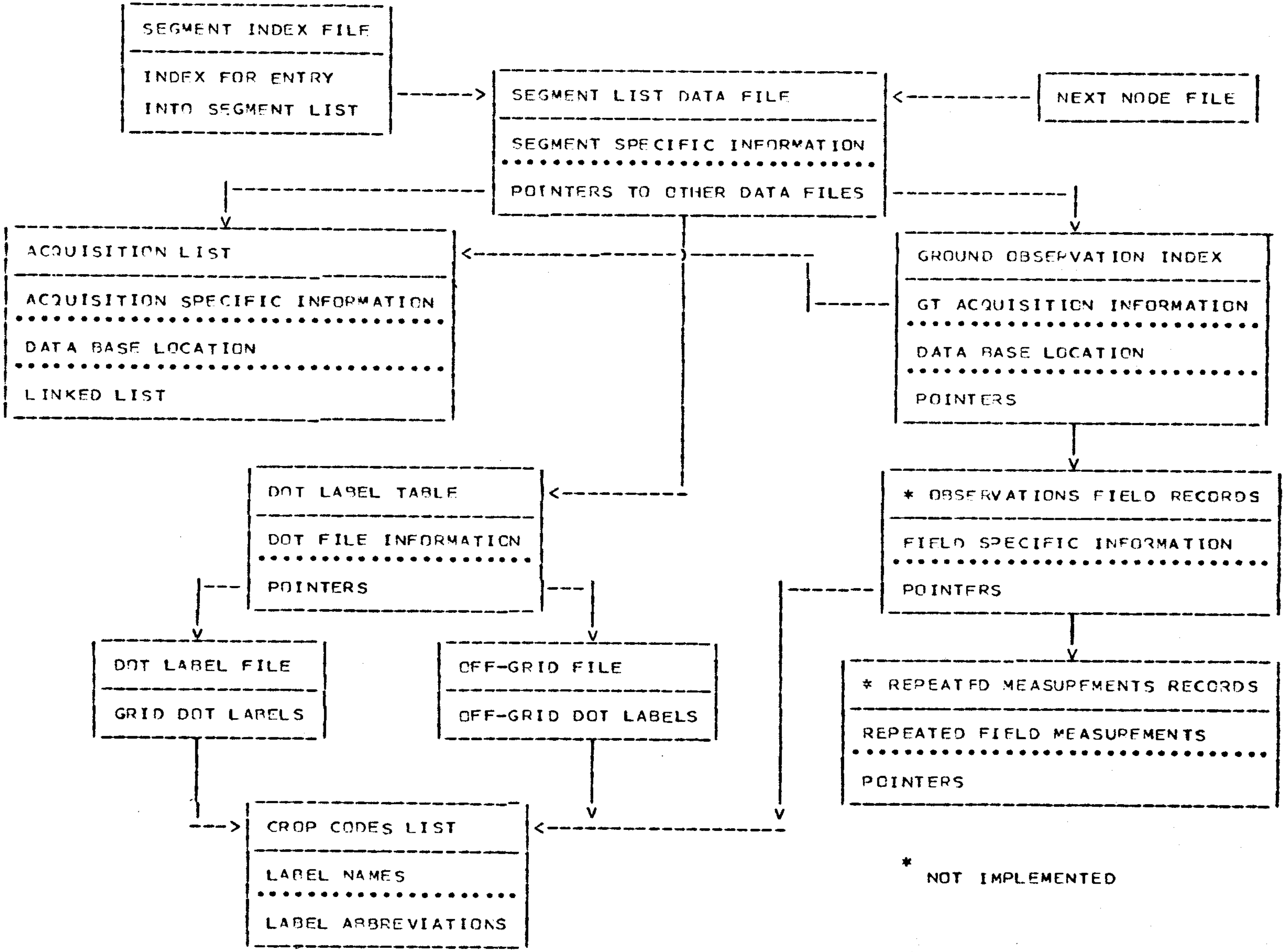
CATALOG. THEN THE DATA MUST BE TRANSMITTED TO THE DATA MANAGER WITH THE 'SUBMIT' ROUTINE. THE NEW DOT DATA FILE HAS NOW ENTERED THE DATA BASE MANAGER PHASE.

THE DATA BASE MANAGER WILL LOG INTO THE LARS COMPUTER ON A SCHEDULED BASIS. WHILE LOGGED IN, THE MANAGER WILL DETERMINE IF ANY NEW DOT DATA FILES HAVE BEEN TRANSMITTED. IF FILES HAVE BEEN SENT, THEN THE DATA SET WILL PASS THROUGH A FORMAT AND DATA VERIFICATION PROCESSOR (VERIFY). IF A FILE IS FOUND TO BE INVALID, THEN THE ORIGINATOR OF THE FILE WILL BE NOTIFIED VIA THE LARS 'MAIL' SYSTEM. IF THE FILE PASSES THE VERIFICATION TEST, THEN IT WILL BE INSERTED INTO THE DATA BASE WITH THE 'DOTBUILD' PROGRAM. THE ANALYST WILL THEN BE NOTIFIED BY LARS MAIL OF THE SUCCESSFUL INSERTION. IF FOR SOME REASON A DOT FILE MUST BE DELETED FROM THE RT&E DATA BASE, THEN THE DATA BASE MANAGER MUST BE NOTIFIED OF THIS REQUEST IN PERSON, BY PHONE, OR IN WRITING. THE CURRENT MANAGER IS LUKE KRAEMER, LARS, 1220 POTTER DRIVE, WEST LAFAYETTE, IN 47906, (317)-749-2052.

DOT GROUND INVENTORY DATA BASE SYSTEM

MODIFICATION OF THE RT&E DATA BASE TO ACCOMMODATE LABELED DOT INFORMATION.

RT&F DATA BASE



FORMAT OF NEXTNODE FILE

ENTRY	FORMAT	BYTES
POINTER TO NEXT FREE SEGMENT	I*4	1 - 2
POINTER TO THE END OF SEGMENT LIST	I*4	5 - 8
POINTER TO THE FIRST FREE ACQUISITION	I*4	9 - 12
POINTER TO THE END OF ACQUISITION LIST	I*4	13 - 16
POINTER TO THE FIRST FREE DOT HEADER	I*4	17 - 20
POINTER TO END OF DOT HEADER LIST	I*4	21 - 24
POINTER TO FIRST FREE GRID RECORD	I*4	25 - 28
POINTER TO END OF GRID LIST	I*4	29 - 32
POINTER TO FIRST FREE OFFGRID RECORD	I*4	33 - 36
POINTER TO END OF OFFGRID LIST	I*4	37 - 40
NUMBER OF CROP NAMES LISTS IN USE	I*4	41 - 44
POINTER TO THE FIRST GROUND OBSERVATION ENTRY	I*4	45 - 48
POINTER TO THE END OF THE GROUND OBSERVATION LIST	I*4	49 - 52
UNUSED		53 - 80

SEGMENT INDEX FORMAT

ENTRY	FORMAT	BYTES
POINTER TO FILE FOR SEGMENT 1	I*2	1 - 2
POINTER TO FILE FOR SEGMENT 2	I*2	3 - 4
POINTER TO FILE FOR SEGMENT 3	I*2	5 - 6
POINTER TO FILE FOR SEGMENT 4	I*2	7 - 8
.		
.		
.		
.		
.		
.		
POINTER TO FILE FOR SEGMENT 9998	I*2	19995 - 19996
POINTER TO FILE FOR SEGMENT 9999	I*2	19997 - 19998
POINTER TO FILE FOR SEGMENT 10000	I*2	19999 - 20000

SEGMENT LIST

ENTRY		FORMAT	BYTES
DATE SEGMENT INITIATED (YYDDD)		I*4	1 - 4
SEGMENT DESCRIPTOR (COUNTY,STATE,OTHER)		8A*4	5 - 36
ACQUISITION LIST POINTER:	FIRST ENTRY	I*4	37 - 40
	LAST ENTRY	I*4	41 - 44
LABEL INDEX POINTER :	FIRST ENTRY	I*4	45 - 48
	LAST ENTRY	I*4	49 - 52
GROUND OBSERVATION INDEX:	FIRST ENTRY	I*4	53 - 56
	LAST ENTRY	I*4	57 - 60
SEGMENT NUMBER		I*2	61 - 62
SEGMENT CENTER LATITUDE (MINUTES NORTH)		I*2	63 - 64
SEGMENT LONGITUDE (MINUTES EAST)		I*2	65 - 66
COUNTRY (NUMBER CODED)		I*2	67 - 68
STATE (NUMBER CODED)		I*2	69 - 70
COUNTY (NUMBER CODED)		I*2	71 - 72
AGRO-PHYSICAL UNIT		I*2	73 - 74
CROP REPORTING DISTRICT		I*2	75 - 76
PREVIOUS SEGMENT ENTRY		I*2	77 - 78
NEXT SEGMENT ENTRY		I*2	79 - 80
CROP YEAR		L*1	81
UNUSED			82 - 100

ACQUISITION LIST

ENTRY	FORMAT	BYTES
SENSOR SYSTEM	A*8	1 - 8
PREVIOUS ACQUISITION	I*4	9 - 12
NEXT ACQUISITION	I*4	13 - 16
ORBIT NUMBER	I*4	17 - 20
SCENE FRAME ID (SFI)	I*4	21 - 24
REFERENCE SFI	I*4	25 - 28
REFERENCE SFI FOR GROUND OBSERVATIONS	I*4	29 - 32
DATE DATA COLLECTED (YYDDD)	I*4	33 - 36
TIME DATA COLLECTED (GMT)	I*4	37 - 40
DATE ENTERED IN DATABASE (YYDDD)	I*4	41 - 44
GODDARD PROCESSING DATE	I*4	45 - 48
DATE OF UNLOAD TAPE	I*4	49 - 52
PEAK SHARPNESS	R*4	53 - 56
NORMALIZED PEAK TO BACKGROUND RATIO	R*4	57 - 60
SEGMENT NUMBER	I*2	61 - 62
SUN ELEVATION (MIN.)	I*2	63 - 64
SUN AZIMUTH (MIN.)	I*2	65 - 66
TAPE NUMBER	I*2	67 - 68
LINES OF DATA	I*2	69 - 70
COLUMNS OF DATA	I*2	71 - 72
PFC BIAS FOR CHANNEL 1	I*2	73 - 74
PFC BIAS FOR CHANNEL 2	I*2	75 - 76
PFC BIAS FOR CHANNEL 3	I*2	77 - 78
PFC BIAS FOR CHANNEL 4	I*2	79 - 80
PFC GAIN FOR CHANNEL 1	I*2	81 - 82
PFC GAIN FOR CHANNEL 2	I*2	83 - 84
PFC GAIN FOR CHANNEL 3	I*2	85 - 86
PFC GAIN FOR CHANNEL 4	I*2	87 - 88
CLOUD COVER	L*1	89
PROCESSING FLAG	L*1	90
GREENNESS OF SOIL LINE	L*1	91
XSTAR HAZE PARAMTER	L*1	92
FILE NUMBER	L*1	93
FIRST CHANNEL	L*1	94
LAST CHANNEL (NC)	L*1	95
SITE	L*1	96
LANDSAT NUMBER	L*1	97
DATA CLASSIFICATION	L*1	98
ARCHIVE FILE NUMBER	L*1	99
JSC FILE NUMBER	L*1	100
JSC TAPE NUMBER	I*4	101 - 104
ARCHIVE TAPE NUMBER	I*4	105 - 108
UNUSED		109 - 120

DOT LABEL TABLE

ENTRY	FORMAT	BYTES
PREVIOUS LABEL ENTRY	I*4	1 - 4
NEXT LABEL ENTRY	I*4	5 - 8
SEGMENT NUMBER	I*2	9 - 10
CROP YEAR	I*2	11 - 12
DATE ENTERED DATA BASE (YYDDDD)	I*4	13 - 16
DATE OF LABELING (YYDDD)	I*4	17 - 20
ANALYST NAME	6A*4	21 - 44
ANALYST IDENTIFIER	I*4	45 - 48
ACQUISITION USED IN LABELING		
DATE 1 (YYDDD)	I*4	49 - 52
•		
•		
•		
DATE 6 (YYDDD)	I*4	69 - 72
SPECIAL CODES USED		
CODE S0	4A*4	73 - 88
CODE S1	4A*4	89 - 104
•		
•		
•		
CODE S9	4A*4	217 - 232
LABELING CONVENTION	I*4	233 - 236
DOT LABEL FILE ADDRESS	I*4	237 - 240
OFF-GRID FILE ADDRESS	I*4	241 - 244
NUMBER OF OFF-GRID POINTS	I*2	245 - 246
DOT LABEL TYPE	I*2	247 - 248
DATE AIRCRAFT IMAGERY ACQUIRED (YYDDD)	I*4	249 - 252
DATE INVENTORY MADE OR RECEIVED AT JSC	I*4	253 - 256
FILE NUMBER	I*2	257 - 258
DATA INPUT TYPE	A*2	259 - 260
EXPERIMENT NUMBER	I*4	261 - 264
USER COMMENTS	72A*1	265 - 336
DATA ENTRY CLERK NAME	6A*4	337 - 360
CROP CODE LIST IDENTIFIER	A*1	361
GRID NUMBER 1	A*1	362
GRID NUMBER 2	A*1	363
UNUSED		364 - 400

GRID POINTS LIST

ENTRY		FORMAT	BYTES
ROW 1 COL 1	LABELS	8L*1	1 - 8
ROW 1 COL 2	LABELS	8L*1	9 - 16
⋮			
ROW 1 COL 19	LABELS	8L*1	144 - 152
ROW 2 COL 1	LABELS	8L*1	153 - 200
⋮			
ROW 22 COL 19	LABELS	8L*1	3336 - 3344

NOTE: IN FILES WITH THE BIGDOT FORMAT, THE EVEN NUMBERED ROWS CONTAIN THE BIGDOT LABELS, AND THE ODD NUMBERED ROWS, THE SMALL DOT LABELS

OFFGRID LIST

ENTRY	FORMAT	BYTES
X COORDINATE 1	L*1	1
Y COORDINATE 1	L*1	2
UNUSED		3 - 4
LABELS FOR OFFGRID POINT 1 (SMALL DOT)	8L*1	5 - 12
LABELS FOR OFFGRID POINT 1 (BIGDOT)	8L*1	13 - 20
X COORDINATE 2	L*1	21
Y COORDINATE 2	L*1	22
UNUSED		23 - 24
LABELS FOR OFFGRID POINT 2 (SMALL DOT)	8L*1	25 - 32
LABELS FOR OFFGRID POINT 2 (BIGDOT)	8L*1	33 - 40
•		
•		
•		
X COORDINATE 200	L*1	3981
Y COORDINATE 200	L*1	3982
UNUSED		3983 - 3984
LABELS FOR OFFGRID POINT 200 (SMALL DOT)	8L*1	3985 - 3992
LABELS FOR OFFGRID POINT 200 (BIGDOT)	8L*1	3992 - 4000

CROP CODES LIST

ENTRY	FORMAT	BYTES
RECORD 1		
NUMBER OF ENTRIES	I*3	1 - 3
RECORD 2		
CROP NAME 1	4A*4	1 - 16
CROP CODE 1	A*2	17 - 18
RECORD 3		
CROP NAME 2	4A*4	1 - 16
CROP CODE 2	A*2	17 - 18
•		
•		
•		
•		
RECORD N+1		
CROP NAME N	4A*4	1 - 16
CROP CODE N	A*2	17 - 18

GROUND OBSERVATION LIST

ENTRY	FORMAT	BYTES
PREVIOUS GROUND TRUTH ENTRY	I*4	1 - 4
NEXT GROUND TRUTH ENTRY	I*4	5 - 8
SEGMENT NUMBER	I*2	9 - 10
NUMBER OF FIELDS MONITORED FOR AGRO-NOMIC DATA	I*2	11 - 12
DATA OF INITIAL GT RECORD (YYDDD)	I*4	13 - 16
DATE OF GT REFERENCE (YYDDD)	I*4	17 - 20
POINTER TO ACQUISITION LIST FOR FIRST W TO W GT	I*4	21 - 24
POINTER TO ACQUISITION LIST FOR LAST W TO W GT	I*4	25 - 28
POINTER TO FIRST MONITORED FIELD	I*4	29 - 32
POINTER TO LAST MONITORED FIELD	I*4	33 - 36
UNUSED		37 - 50

OBSERVATION FIELD LIST

ENTRY	FORMAT	BYTES
PREVIOUS FIELD MONITORED	I*4	1 - 4
NEXT FIELD MONITORED	I*4	5 - 8
SEGMENT NUMBER	I*2	9 - 10
FIELD NUMBER	I*2	11 - 12
FIELD IDENTIFIER	A*8	13 - 20
CROP NAMES ENTRY	I*2	21 - 22
CROP STATUS ENTRY	I*2	23 - 24
DATE PLANTED (YYDDD)	I*2	25 - 26
NITROGEN FERTILIZATION	I*2	27 - 28
ROW WIDTH (METERS)	R*4	29 - 32
POINTER TO FIRST OF REPEATED MEASURES	I*4	33 - 36
POINTER TO LAST OF REPEATED MEASURES	I*4	37 - 40
NUMBER OF ARCS (NARC)	I*2	41 - 42
LINE COORDINATE 1	I*2	43 - 44
COLUMN COORDINATE 1	I*2	45 - 46
LINE COORDINATE 2	I*2	47 - 48
COLUMN COORDIANTE 2	I*2	49 - 50
• • • •		
LINE COORDINATE NARC	I*2	39+NARC*4 - 40+NARC*4
COLUMN COORDINATE NARC	I*2	41+NARC*4 - 42+NARC*4

REPEATED MEASURE LIST

ENTRY	FORMAT	BYTES
PREVIOUS MEASUREMENT	I*4	1 - 4
NEXT MEASUREMENT	I*4	5 - 8
SEGMENT NUBER	I*2	9 - 10
FIELD NUMBER	I*2	11 - 12
DATE MEASURED (YYDDD)	I*4	13 - 16
MATURITY	L*1	17
% OF GROUND COVER	L*1	18
% OF GREEN LEAVES	L*1	19
CONDITION	L*1	20

DOT GROUND INVENTORY DATA BASE SYSTEM

SOFTWARE FOR SUBMISSION OF LABELED DOT FILES FOR INPUT INTO THE DATA BASE.

INTRODUCTION TO THE DOT GROUND INVENTORY DATA BASE SYSTEM

THE FOLLOWING IS A SERIES OF DOCUMENTATION FOR THE DOT DATA BASE. IT IS LOCATED ON THE JSCDISK 19A, WHICH THE USER SHOULD HAVE ACCESSED (IN READ-ONLY MODE) PRIOR TO THE EXECUTION OF ANY OF THE ROUTINES DESCRIBED BELOW. THIS DOCUMENTATION CAN BE PRINTED BY TYPING 'DOTHELP'. TO DISPLAY THIS DOCUMENTATION ON THE TERMINAL, TYPE 'DOTHELP TERM'.

DESCRIPTION OF DOTINIT ;

DOTINIT IS USED TO INITALIZE A USER'S DISK WITH A DOT FILE FOR EDITING. THE DOT FILE CAN EITHER BE ONE IN THE RT&E DATA BASE OR A NEW DOT FILE. THE PROGRAM PROMPTS THE USER FOR THE FILENAME AND FILETYPE THE DOT FILE IS TO BE LOADED UNDER. THE FILEMODE IS DEFAULTED TO 'A'. IF THE USER NEEDS A DIFFERENT FILEMODE, HE MAY ENTER IT AFTER THE FILETYPE.

EXAMPLE OF OPERATION ;

SOURCE	RESPONSE
USER:	DOTINIT
PROGRAM:	INITIALIZATION OF USER'S DISK WITH A NEW EMPTY DOT FILE OR A DOT FILE IN THE RT&E DATA BASE. INPUT THE FILENAME, FILETYPE, <FILEMODE> THAT THE DOT FILE IS TO BE CALLED. THE FILEMODE IS DEFAULTED TO 'A'.
USER:	DOT FILE
PROGRAM:	IF DOT FILE IS IN THE RT&E DATA BASE TYPE 'OLD'. IF THE DOT FILE IS NOT IN THE RT&E DATA BASE TYPE 'NEW'. TYPE 'STOP' TO STOP DOTINIT.
USER:	NEW
PROGRAM:	IF THE FILE BEING CREATED IS A SMALL DOT FILE, TYPE '1' IF THE FILE BEING CREATED IS A BIG DOT FILE, TYPE '2'
USER:	1
PROGRAM:	THE FILE IS INITIALIZED.

RESULT: THE USER HAS CREATED A SMALL DOT FILE ON HIS DISK CALLED 'DOT FILE'.

THE FILENAME, FILETYPE, AND OPTIONALLY FILEMODE ARE ENTERED AT THE TIME DOTINIT IS STARTED.

EXAMPLE ;

USER:	DOTINIT SEG800 DOT
PROGRAM:	IF THE DOT FILE IS IN THE RT&E DATA BASE TYPE 'OLD' IF THE DOT FILE IS NOT IN THE RT&E DATA BASE TYPE 'NEW'. TYPE 'STOP' TO STOP DOTINIT.
USER:	OLD
PROGRAM:	WHAT IS THE SEGMENT NUMBER OF THE FILE? (1 - 9999)
USER:	800
PROGRAM:	DO YOU REQUIRE A QUERY OF THAT SEGMENT NUMBER? (YES/NO)
USER:	NO
PROGRAM:	WHAT IS THE CROP YEAR OF THE DOT FILE? (75 - 99)
USER:	77
PROGRAM:	WHAT IS THE FILE SEQUENCE NUMBER OF THE DOT FILE? (INPUT A '0' TO STOP DOTINIT.)

NOTE: THE PROGRAM EXPECTS A INTEGER 0 TO 64. THIS IS FOUND FROM A QUERY OF THE DATA BASE USING 'DOTQUERY'.

USER: 3

IF THE PROGRAM FOUND THE DOT FILE REQUESTED, THEN THAT FILE IS LOADED UNDER THE NAME 'SEG900 DOT'. IF THE DOT FILE IS NOT FOUND, THEN AN ERROR MESSAGE IS DISPLAYED AT THE TERMINAL.

DOTEDIT
PROGRAM TO EDIT DOT FILES

DESCRIPTION: DOTEDIT IS THE DOT FILE EDITOR. IT ALLOWS THE USER TO INTERACTIVELY DEFINE GEOGRAPHICAL AND TEMPORAL INFORMATION IN A HEADER, AND ENTER AND/OR CHANGE DOT LABELS. THE EDITOR CONSISTS OF THREE FUNCTIONALLY SEPARATED SECTIONS: ONE FOR EDITING HEADER INFORMATION, ONE FOR EDITING GRID POINTS, AND ONE FOR EDITING OFFGRID POINTS. WHAT FOLLOWS IS A DETAILED DESCRIPTION OF EACH OF THE COMMANDS FOR THE DOT EDITOR.

THE COMMANDS ARE CLASSIFIED INTO FOUR GROUPS, DEPENDING UPON WHICH EDIT SECTION A COMMAND MAY BE USED IN. 'GLOBAL' COMMANDS ARE COMMANDS WHICH MAY BE EXECUTED IN ANY OF THE THREE SECTIONS. THE OTHER COMMANDS ARE SPECIFIED AS 'HEADER', 'LABEL', OR 'OFFGRID' FOR THE HEADER, LABEL, AND OFFGRID SECTIONS OF THE EDITOR RESPECTIVELY. IN THE UPPER RIGHT HAND CORNER OF EACH PAGE DESCRIBING COMMANDS, THE CLASSIFICATION OF THE COMMAND BEING DESCRIBED IS PRINTED. IT IS POSSIBLE FOR ONE COMMAND TO HAVE SEVERAL DIFFERENT CLASSIFICATIONS. THIS MEANS THAT FOR EACH EDITOR SECTION WHICH ALLOWS THE COMMAND, THE COMMAND MAY HAVE A DIFFERENT SYNTAX AND EFFECT ON THE DOT FILE BEING EDITED. IN ALL OF THE SECTIONS IT IS POSSIBLE TO ENTER AN INPUT MODE IN WHICH THE USER ENTERS CONTINUOUS STREAMS OF DATA. WHILE IN INPUT MODE, IT IS NOT PERMITTED TO ENTER ANY COMMANDS, ONLY DATA.

HEADER COMMAND

DESCRIPTION: THIS COMMAND ALLOWS THE USER TO CHANGE AND INPUT THE HEADER INFORMATION ASSOCIATED WITH HIS DOT FILE. THE FOLLOWING ARE THE COMMANDS WHICH ARE ALLOWED WHILE THE USER IS IN THE HEADER SECTION OF THE EDITOR:

TYPE CHANGE INPUT.

FOR FURTHER INFORMATION, SEE THE DOCUMENTATION FOR THE INDIVIDUAL COMMANDS (HEADER APPLICATION).

FORMAT: HEADER

NOTES: THIS SUBSECTION OF THE EDITOR IS THE FIRST ONE ENTERED WHEN EDITING BEGINS. IN ANY OTHER SUBSECTION, THE EDITOR IMMEDIATELY TRANSFERS CONTROL TO THE HEADER SECTION. IF THE USER IS ALREADY IN THE HEADER SECTION AND TYPES THIS COMMAND, THERE IS NO CHANGE. ALL THE LABELS AND INFORMATION ENTERED PREVIOUSLY ARE KEPT IN MEMORY, IT IS NEITHER ERASED NOR WRITTEN OUT TO THE FILE. THE 'HEADER' COMMAND CAN BE ENTERED AT ANY TIME EXCEPT WHILE IN INPUT MODE.

EXAMPLES: HEADER

LABEL COMMAND

DESCRIPTION: THIS COMMAND ALLOWS THE USER TO ENTER AND CHANGE LABELS FOR THE GRID POINTS ASSOCIATED WITH THE FILE BEING EDITED. THE FOLLOWING ARE COMMANDS WHICH ARE ALLOWED IN THE LABEL SECTION OF THE EDITOR:

INPUT	TYPE	CHANGE	HELP
SET	ADD	REM	UP
DOWN	LEFT	RIGHT	FIRST
LAST	TOP	BOTTOM	HOME
PTR	CODES	HEADER	LABEL
OFFGRID	SAVE	QUIT.	

FOR FURTHER INFORMATION SEE THE DOCUMENTATION FOR THE INDIVIDUAL COMMANDS.

FORMAT: LABEL

NOTES: THIS SECTION OF THE EDITOR ALLOWS THE USER TO CHANGE AND ENTER LABELS FOR THE GRID POINTS. IT MAY BE TYPED AT ANY TIME EXCEPT DURING INPUT MODE. AND CONTROL IS TRANSFERRED TO THE LABEL SECTION. IF THE USER IS ALREADY IN THE LABEL SECTION, NO CHANGE OCCURS.

LABELS ARE ENTERED IN THE FOLLOWING FASHION. FIRST ALL THE SMALL DOT LABELS, FOLLOWED BY A SLASH, AND THEN ALL THE BIGDOT LABELS. THE LABELS CONSIST OF A TWO CHARACTER CODE (THE CODES MAY BE DISPLAYED BY TYPING THE 'CODES' COMMAND WHILE IN THE 'LABEL' SECTION OF THE EDITOR.) IF THERE IS ONLY ONE CHARACTER IN THE CODE, IT MUST BE FOLLOWED BY A SPACE, SO THAT THE LENGTH IS TWO CHARACTERS. ALL THE CODES FOLLOW EACH OTHER WITH NO SPACES BETWEEN THEMSELVES, EXCEPT FOR THOSE SPACES REQUIRED TO MAKE THE LENGTH OF THE CODE TWO CHARACTERS. THERE IS NO SPACE BEFORE OR AFTER THE SLASH SEPARATING SMALL DOTS FROM BIG DOTS. IF THERE ARE NO BIG DOTS, THE SLASH NEED NOT BE SPECIFIED, BUT IT MUST OCCUR IF THERE ARE BIGDOTS AND NO SMALL DOTS. FOR EXAMPLES OF THIS, SEE THE SAMPLE TERMINAL SESSION, AND THE EXAMPLES OF SOME OF THE COMMANDS.

DURING INPUT MODE (SEE THE DOCUMENTATION FOR 'INPUT') THE LABELS FOR A DOT ARE INPUT AS DESCRIBED ABOVE, A COMMA IS USED TO SEPARATE THE LABELS FOR ONE DOT FROM THE LABELS FOR THE NEXT DOT. IF LABELS MUST APPEAR ON A COMMAND LINE --- SUCH AS IN THE 'SET' COMMAND OR THE 'CHANGE' COMMAND --- THEN THEY MUST BE PRECEDED BY AN EQUALS SIGN ('=') TO INDICATE TO THE EDITOR THAT LABELS, AND NOT ANOTHER OPTION FOR THE COMMAND, FOLLOW. THERE SHOULD NOT BE ANY SPACES AFTER THE EQUALS SIGN.

IF THE FILE IS A SMALL DOT FILE, AND BIGDOT LABELS ARE ENTERED, THEN ALL THE LABELS AFTER THE SLASH ARE IGNORED. NO WARNING MESSAGE IS GIVEN.

A POINTER IS KEPT WHICH KEEPS TRACK OF THE ROW AND COLUMN CURRENTLY UNDER CONSIDERATION. WHEN LABEL IS FIRST ENTERED, IT IS SET TO THE HOME DOT IN THE UPPER LEFT HAND CORNER OF THE GRID (AT ROW 1 COLUMN 1). WHENEVER A COMMAND IS EXECUTED, ITS EFFECT IS ALWAYS REFERENCED TO THE DOT WHICH IS POINTED TO BY THE POINTER. FOR EXAMPLE, THE 'TYPE' COMMAND STARTS TYPING LINES BEGINNING WITH THE ROW HELD BY THE POINTER, AND THE 'INPUT' COMMAND STARTS AT THE DOT WITH THE ROW AND COLUMN OF THE POINTER.

FINALLY, THERE ARE TWO METHODS OF VIEWING, AND CHANGING THE GRID. BY DEFAULT, AS SOON AS THE LABEL SECTION IS ENTERED, PROCESSING IS DONE BY ROWS. STARTING AT ROW 1, CHANGES AND PROCESSING IS DONE FOR ALL THE COLUMNS IN THE ROW, AND THEN THE ROW IS INCREMENTED. IN ADDITION, THE 'TYPE' COMMAND TYPES OUT ROWS OF DATA. THERE IS A SECOND OPTION AVAILABLE, IF THE USER DESIRES. BY CHANGING THE MODE TO COLUMNS, ALL PROCESSING IS DONE FOR A GIVEN COLUMN FIRST. IN THIS SECOND ALTERNATE MODE, THE

TYPE COMMAND WOULD TYPE OUT COLUMNS OF DATA INSTEAD OF ROWS OF DATA.

EXAMPLES: LABEL

OFFGRID COMMAND

DESCRIPTION: THIS COMMAND ALLOWS THE USER TO ENTER AND CHANGE LABELS FOR THE OFFGRID POINTS ASSOCIATED WITH THE FILE BEING EDITED. THE FOLLOWING ARE COMMANDS WHICH ARE ALLOWED IN THE OFFGRID SECTION OF THE EDITOR:

INPUT	TYPE	CHANGE	HELP
SET	ADD	REM	UP
DOWN	TOP	BOTTOM	CODES
HEADER	LABEL	OFFGRID	SAVE
QUIT.	(XXX,YYY).		

FOR FURTHER INFORMATION SEE THE DOCUMENTATION FOR THE INDIVIDUAL COMMANDS.

FORMAT: OFFGRID

NOTES: THIS SECTION OF THE EDITOR ALLOWS THE USER TO CHANGE AND ENTER LABELS FOR THE OFFGRID POINTS. THE 'OFFGRID' COMMAND MAY BE ENTERED AT ANY TIME, EXCEPT DURING INPUT MODE, AND CONTROL IS TRANSFERRED IMMEDIATELY TO THE OFFGRID SECTION. IF THE USER IS ALREADY IN THE OFFGRID SECTION, NO CHANGE OCCURS.

THE COORDINATES CONSIST OF TWO VALUES SEPARATED BY COMMAS. THE FIRST COORDINATE IS FOR THE ROW, AND THE SECOND IS FOR THE COLUMN OF THE OFFGRID POINT. THE OFFGRID POINTS ARE KEPT IN A SORTED LIST, IN ORDER OF INCREASING ROW NUMBER. IF SEVERAL OFFGRID POINTS HAVE THE SAME ROW NUMBER, THEY ARE SUBSORTED USING THE COLUMN AS THE SORT KEY.

LABELS ARE ENTERED IN THE FOLLOWING FASHION. FIRST AN EQUALS SIGN ('='), THEN ALL THE SMALL DOT LABELS, FOLLOWED BY A SLASH, AND FINALLY ALL THE BIGDOT LABELS. THE LABELS CONSIST OF A TWO CHARACTER CODE (THE CODES MAY BE DISPLAYED BY TYPING THE 'CODES' COMMAND WHILE IN THE 'OFFGRID' SECTION OF THE EDITOR.) IF THERE IS ONLY ONE CHARACTER IN THE CODE, IT MUST BE FOLLOWED BY A SPACE, SO THAT THE LENGTH IS TWO CHARACTERS. ALL THE CODES FOLLOW EACH OTHER WITH NO SPACES BETWEEN THEMSELVES, EXCEPT FOR THOSE SPACES REQUIRED TO MAKE THE LENGTH OF THE CODE TWO CHARACTERS. THERE IS NO SPACE BEFORE OR AFTER THE SLASH SEPARATING SMALL DOTS FROM BIGDOTS. IF THERE ARE NO BIGDOTS, THE SLASH NEED NOT BE SPECIFIED, BUT IT MUST OCCUR IF THERE ARE BIGDOTS AND NO SMALL DOTS. FOR EXAMPLES OF THIS, SEE THE SAMPLE TERMINAL SESSION, AND THE EXAMPLES FOLLOWING THE DESCRIPTIONS OF THE OFFGRID COMMANDS.

DURING INPUT MODE (SEE THE DOCUMENTATION FOR 'INPUT') THE LABELS FOR AN OFFGRID POINT ARE INPUT AS DESCRIBED ABOVE. TWO OFFGRID POINT COORDINATES MUST PRECEDE THE LABELS, AND THE EQUALS SIGN IS STILL NECESSARY. IF THE FILE IS A SMALL DOT FILE, AND BIGDOT LABLS ARE ENTERED, ALL THE BIGDOT LABELS ARE IGNORED. NO WARNING MESSAGE IS GIVEN.

A POINTER IS KEPT WHICH KEEPS TRACK OF THE OFFGRID POINT CURRENTLY UNDER CONSIDERATION. WHEN 'OFFGRID' IS FIRST ENTERED, THE POINTER IS SET TO THE FIRST OFFGRID POINT. (INTERNALLY, THE OFFGRID POINTS ARE MAINTAINED IN SORTED ORDER.) WHENEVER A COMMAND IS EXECUTED, ITS EFFECT IS REFERENCED ACCORDING TO THE CURRENT SETTING OF THE POINTER. IT IS POSSIBLE ON SOME COMMANDS ('SET', 'ADD', 'CHANGE', 'TYPE', 'DEL', AND 'REM') TO SPECIFY COORDINATES BEFORE THE COMMAND, IN WHICH CASE THE POINTER IS SET TO THE OFFGRID POINT WITH THE COORDINATES. (AN ERROR OCCURS IF THE COORDINATES SPECIFIED ARE FOR A NONEXISTENT POINT.) IF THERE ARE NO OFFGRID POINTS IN THE FILE, THE POINTER IS SET TO A SPECIAL VALUE, ZERO, TO INDICATE THAT THERE ARE NO POINTS DEFINED. MOST COMMANDS WILL HAVE NO EFFECT WHEN THERE ARE NO OFFGRID POINTS.

EXAMPLE: OFFGRID

HELP COMMAND

DESCRIPTION: THE HELP COMMAND WILL LIST OUT TO THE USER ALL THE COMMANDS AND THE FIELDS THAT THEY CAN EDIT AT THAT TIME.

FORMAT: HELP

NOTES: NONE

EXAMPLES: HELP

SAVE COMMAND

DESCRIPTION: THE SAVE COMMAND WILL SAVE ALL FILES YOU HAVE CHANGED AND STOP EDITING.

FORMAT: SAVE

NOTES: ALL CHANGES MADE DURING THE EDITING SESSION AND ALL DATA INPUT DURING THE SESSION ARE SAVED. THIS COMMAND MAY BE ENTERED AT ANY TIME, EXCEPT DURING INPUT MODE.

EXAMPLES: SAVE

QUIT COMMAND

DESCRIPTION: THE QUIT COMMAND WILL STOP EDITING AND WILL NOT SAVE ANY CHANGES.

FORMAT: QUIT

NOTES: ALL CHANGES MADE DURING THE EDITING SESSION AND ALL DATA INPUT DURING THE SESSION WILL BE IGNORED. THIS COMMAND CAN BE ENTERED AT ANY TIME EXCEPT WHILE IN INPUT MODE.

EXAMPLES: QUIT

CODES COMMAND

DESCRIPTION: THIS TYPES OUT THE LIST OF VALID LABELS AND THEIR CORRESPONDING CROP NAMES.

FORMAT: CODES

NOTES: BOTH THE STANDARD PREDEFINED CODES (STORED IN THE CROP NAMES FILE) AND THE USER DEFINED SPECIAL CROP CODES ARE DISPLAYED.

EXAMPLES: >CODES

CROP	CODE	CROP	CODE
WINTER WHEAT	W	SPRING WHEAT	SW
GRASS	G	HAY	H
ALFALFA	A	PASTURE	P
CORN	C	SORGHUM	SR
SOYBEANS	SY	SUGARBEATS	SB
SAFFLOWER	SF	SUNFLOWER	SU
FLAX	FX	TREES	T
RYE	R	FALL BARLEY	BF
SPRING BARLEY	BS	NON-AG	X
FALL OATS	OF	SPRING OATS	OS
IDLE	I	IDLE STUBBLE	IS
IDLE FALLOW	IF	IDLE CROP COVER	IC
IDLE RESIDUE	IP	ROAD	RD
UNKNOWN	U	WATER	WA
CLOVER	CL	DURUM WHEAT	DW
COTTON	CN	RICE	RI
RHUBARB	S0	PEAS	S1
CABBAGES	S2	KUMQUATS	S3
PINEAPPLES	S4	HEMP	S5
POPPIES	S6	CUCUMBERS	S7
RADISHES	S8	RUBBER TREES	S9

INPUT COMMAND

DESCRIPTION: THE INPUT COMMAND WILL ALLOW THE USER TO INPUT THE INFORMATION FOR THE HEADER RECORD AS THE PROGRAM REQUESTS IT. IT WILL START AT THE FIELD SPECIFIED OR ELSE DEFAULT TO START AT SEGNUM IF THE FIELD IS NOT SPECIFIED.

FORMAT: INPUT (FIELD)

NOTES: THE FIELDS AVAILABLE ARE AS FOLLOWS :

HEADER FIELD	ABBREVIATION
SEGMENT NUMBER............................	SEGNUM
CROP YEAR.................................	CROPYR
DATE OF LABELING..........................	LAEDATE
ANALYST NAME..............................	NAME
ANALYST IDENTIFIER........................	ID
ACQUISITION USED IN LABELING..............	ACQ (I1)
LABELING CONVENTION.......................	LC
DATE AIRCRAFT IMAGERY ACQUIRED............	ACDATE
DATE INVENTORY MADE OR RECEIVED AT JSC....	JSCDATE
SPECIAL CROP CODES........................	SC (I1)
EXPERIMENT NUMBER.........................	EXPNUM
USER COMMENTS.............................	COMMENT
DATA INPUT TYPE...........................	DITYPE
DATA ENTRY CLERK NAME.....................	DENAME
GRID NUMBER 1.............................	GRID1
GRID NUMNBER 2............................	GRID2

EXAMPLES: INPUT CROPYR .....WILL START INPUT AT CROPYR

I SC 7 .....WILL START INPUT AT SPECIAL CROP CODE 7

TYPE COMMAND

DESCRIPTION: THE TYPE COMMAND WILL LIST OUT ALL THE VALUES IN THE HEADER RECORD.

FORMAT: TYPE

NOTES: NONE

EXAMPLES:

```
TYPE
DOT HEADER RECORD
SEGMENT NUMBER.............................   200
CROP YEAR..................................    78
DATE OF LABELING.......................... 80151
ANALYST NAME..............................PETER LIESENFELT
ANALYST IDENTIFIER........................  8978
ACQUISITION DATES USED     DATE 1......... 78234
                           DATE 2.........     0
                           DATE 3.........     0
                           DATE 4.........     0
                           DATE 5.........     0
                           DATE 6.........     0
LABELING CONVENTION.......................     8
DATE AIRCRAFT IMAGERY ACQUIRED............ 78275
DATE INVENTORY MADE OR RECIEVED AT JSC.... 78340
DATA INPUT TYPE........................... A
EXPERIMENT NUMBER.........................     1
USER COMMENTS ;
 THIS IS A SAMPLE TERMINAL SESSION ON THE DOT EDITOR
DATA ENTRY CLERK NAME.....................SECRETARY
GRID NUMBER 1.............................     1
GRID NUMBER 2.............................     1

(HIT THE RETURN KEY FOR THE SPECIAL CROP CODES)

SPECIAL CROP CODE S0 IS...................RHUBARB
SPECIAL CROP CODE S1 IS...................PEAS
SPECIAL CROP CODE S2 IS...................CABBAGES
SPECIAL CROP CODE S3 IS...................KUMQUATS
SPECIAL CROP CODE S4 IS...................PINEAPPLES
SPECIAL CROP CODE S5 IS...................HEMP
SPECIAL CROP CODE S6 IS...................POPPIES
SPECIAL CROP CODE S7 IS...................CUCUMBERS
SPECIAL CROP CODE S8 IS...................RADISHES
SPECIAL CROP CODE S9 IS...................RUBBER TREES
```

CHANGE COMMAND

DESCRIPTION: TO CHANGE A VALUE OF A FIELD IN THE HEADER TO A NEW VALUE.

FORMAT: (CHANGE) FIELD (VALUE)

NOTES: THE FIELDS AVAILABLE ARE AS FOLLOWS :

HEADER FIELD	ABBREVIATION
SEGMENT NUMBER	SEGNUM
CROP YEAR	CRCPYR
DATE OF LABELING	LABDATE
ANALYST NAME	NAME
ANALYST IDENTIFIER	ID
ACQUISITION USED IN LABELING	ACQ (I1)
LABELING CONVENTION	LC
DATE AIRCRAFT IMAGERY ACQUIRED	ACDATE
DATE INVENTORY MADE OR RECEIVED AT JSC	JSCDATE
SPECIAL CROP CODES	SC (I1)
EXPERIMENT NUMBER	EXPNUM
USER COMMENTS	CCMMENT
DATA INPUT TYPE	DITYPE
DATA ENTRY CLERK NAME	DENAME
GRID NUMBER 1	GRID1
GRID NUMBER 2	GRID2

EXAMPLES:

SEGNUM 2345 .....CHANGES SEGNUM TO 2345

C LABDATE 4 21 80 .....CHANGES LABDATE TO 4 21 80

INPUT COMMAND

DESCRIPTION: INPUT ENTERS THE USER INTO AN INPUT MODE FOR INDIVIDUALLY LABELING DOTS. IN ROW MODE, DOT LABELS ARE ENTERED ROW BY ROW, AND IN COLUMN MODE THEY ARE ENTERED BY COLUMNS. EACH IS PROMPTED WITH THE ROW (COLUMN) BEING ENTERED AND THE COLUMN (ROW) AT WHICH INPUT IS BEGINNING. IF FOR SOME REASON AN INVALID LABEL OCCURS, THEN THE REST OF THE LINE IS IGNORED, AND INPUT CONTINUES FROM WHERE THE ERROR WAS DETECTED.

FORMAT: INPUT (XXX (,YYY))

NOTES: THE NUMBERS XXX AND YYY ,IF BOTH PRESENT, SPECIFY WHICH DOT TO START LABELING. IF JUST XXX IS PRESENT, THE STARTING ROW (COLUMN IF THE PROCESSING MODE IS SET TO 'COL' (SEE THE DOCUMENTATION FOR 'MODE' OR 'LABEL' FOR FURTHER DETAILS ON THE PROCESSING MODE)) IS SET TO XXX. THE FORMAT FOR THE LABELS IN INPUT MODE IS SPECIFIED IN THE DOCUMENTATION FOR 'LABEL'. THE LABELS ARE SEPARATED BY COMMAS. IF TWO COMMAS FOLLOW EACH OTHER, THEN THE INTERVENING DOT IS SKIPPED, AND THE LABEL IS LEFT TO WHAT IT WAS BEFORE. IF AN ERROR OCCURS, THE REST OF THE LABELS ON THE LINE ARE IGNORED, AND INPUT IS RESUMED AT THE DOT WHERE THE ERROR OCCURRED. UPON REACHING THE LAST DOT IN THE GRID SECTION OF THE FILE, THE INPUT MODE IS AUTOMATICALLY EXITED, OTHERWISE TO EXIT INPUT MODE, IT IS MERELY NECESSARY TO ENTER THE NULL LINE (HIT RETURN AFTER THE PROMPT.)

IF ONE OF THE TWO GRID VARIABLES IS SET, THEN INPUT AUTOMATICALLY INCREMENTS THE POINTER IN A PRESET GRID PATTERN. IF THE GRID TYPE IS 1, INPUTTING IS DONE ONLY FOR EVEN NUMBERED ROWS AND COLUMNS. IF THE TYPE IS 2, THEN IT IS PERFORMED FOR ODD ROWS AND COLUMNS. IF THE TYPE IS 3, THEN LABELS CAN ONLY BE ENTERED FOR EVEN ROWS AND ODD COLUMNS, AND FOR TYPE 4 LABELS ARE INPUT FOR ODD ROWS AND EVEN COLUMNS. IF THE POINTER IS NOT SET TO THE PROPER EVEN/ODD COMBINATION WHEN THE INPUT MODE IS ENTERED, THE USER IS PROMPTED TO INPUT THE CORRECT ROW OR COLUMN. THIS IS LEFT AS A USER INPUT RATHER THAN HAVING THE POINTER SET AUTOMATICALLY TO THE NEAREST CORRECT ROW/COLUMN IN ORDER TO GIVE GREATER FLEXIBILITY IN CONTROLLING INPUT. (CAUTION: THIS ROW/COLUMN CHECKING ONLY OCCURS IN INPUT MODE. IN THE EDIT MODE, NO CHECKING TAKES PLACE. IF BOTH GRID VARIABLES ARE SET TO VALUES, THEN THE CHECKING IS TURNED OFF.)

EXAMPLES:

```
>INPUT 7,4
ENTERING ROW:  7
STARTING AT COLUMN  4
>S U /T ,C X ,W /C H
ENTERING ROW:  7
STARTING AT COLUMN  7
>X U T ,Y Z INVALID,U T /C
             ¬¬
INVALID LABEL NAME
ENTERING ROW:  7
STARTING AT COLUMN  8
>T ,,,U ,
ENTERING ROW:  7
STARTING AT COLUMN 12
>
LABEL:
>MODE COL
>PTR
ROW:  7
COL: 12
```

```
>INPUT  2
ENTERING COL:  2
STARTING AT ROW 7
>U C /X
ENTERING COL:  2
STARTING AT ROW 8
>
LABEL:
>
```

TYPE COMMAND

DESCRIPTION: THIS COMMAND DISPLAYS SEVERAL ROWS (OR COLUMNS) OF LABELLED DOTS.

FORMAT: TYPE (NNN)

NOTES: DEPENDING ON WHETHER THE MODE IS SET TO 'COLUMN' MODE OR 'ROW' MODE, TYPE WILL DISPLAY ON THE TERMINAL NNN ROWS OR COLUMNS RESPECTIVELY OF LABELS. IF NNN IS NOT SPECIFIED, IT IS DEFAULTED TO 1. THE ROWS (OR COLUMNS) TYPE OUT, START FROM THE CURRENT ROW (OR COLUMN) AS SPECIFIED BY THE INTERNALLY MAINTAINED POINTER. (FOR FURTHER EXPLANATION OF THE POINTER SEE THE DOCUMENTATION FOR 'LABEL'.) THE POINTER IS NOT AFFECTED BY THE TYPE COMMAND. (I.E. AFTER EXECUTING 'TYPE 3') THE POINTER WILL STILL HAVE THE SAME VALUE IT HAD BEFORE EXECUTING THE COMMAND.

EXAMPLES:

```
>TYPE
ROW  1
1          2          3          4          5
S U X      C H /U     SB/BF      C H /U     ... ETC. ...
>MODE COL
>T 1
COLUMN 1
1          2          3          4          5
S U X      I SB/X     U W /C      /SBBF     ... ETC. ...
```

FOR FURTHER EXAMPLES OF HOW THE OUTPUT FOR TYPE LOOKS, SEE THE SAMPLE TERMINAL SESSION.

PTR COMMAND

DESCRIPTION: IF THE TWO ARGUMENTS ARE PRESENT, THE POINTER IS SET TO THE VALUE XXX,YYY. OTHERWISE, THE CURRENT VALUE OF THE POINTER IS DISPLAYED ON THE TERMINAL.

FORMAT: PTR (XXX) (,YYY)

NOTES: THE INTERNALLY MAINTAINED POINTER IS DESCRIBED IN THE DOCUMENTATION FOR THE 'LABEL' COMMAND. THE KEYWORD 'PTR' DOES NOT HAVE TO BE SPECIFIED. IF THE EDITOR SEES A NUMBER AS THE FIRST ITEM ON A LINE, IT AUTOMATICALLY SUPPLIES THE KEYWORD: 'PTR'. IF THE FILE EDITING MODE IS ROW (AS SET BY DEFAULT), THEN ROW IS SET TO XXX, AND COLUMN IS SET TO YYY. (NOTE BOTH XXX AND YYY NEED NOT BE PRESENT). IF, HOWEVER, THE MODE IS COLUMN MODE, THEN THE COLUMN IS SET TO XXX, AND THE ROW TO YYY.

EXAMPLES:

```
>PTR
ROW:  6
COL:  7
>PTR 3,4
>PTR
ROW:  3
COL:  4
>
```

SET COMMAND

DESCRIPTION: THIS COMMAND SETS AN AREA TO A GIVEN VALUE.

FORMAT:

```
             (              (       (UP    ) ) )
             (              (       (DOWN  ) ) )
             (              (       (LEFT  ) ) )
             (              (ALL    (RIGHT ) ) )
             (              (       (UL    ) ) )
             (              (       (UR    ) ) )
             (              (       (DL    ) ) )
             (              (       (DR    ) ) )
SET          (   (XX , YY)                     ) =VALUE
             (              (       (UP    ) ) )
             (              (COL    (DOWN  ) ) )
             (              (       (*     ) ) )
             (                                 )
             (              (       (LEFT  ) ) )
             (              (ROW    (RIGHT ) ) )
             (              (       (*     ) ) )
```

NOTES: IF NO ARGUMENTS ARE GIVEN EXCEPT THE VALUE, THEN JUST THE DOT POINTED TO BY THE CURRENT POINTER IS SET TO THE VALUE. SPECIFYING XX,YY WILL MOVE THE POINTER TO THAT PARTICULAR DOT. THE OPTION 'ALL' MEANS THAT THE ENTIRE FILE IS TO BE SET. THE OPTIONS SPLIT THE FILE INTO SEVERAL REGIONS, DEPENDENT ON THE VALUE OF THE POINTER. UP, DOWN, LEFT, RIGHT SPECIFY WHICH HALF PLANE IS TO BE SET. ALL THESE SPECIFICATIONS ARE RELATIVE TO WHERE THE POINTER IS POINTING. UL, UR, DL, AND DR ARE ABBREVIATIONS FOR UP-LEFT, UP-RIGHT, DOWN-LEFT, AND DOWN-RIGHT. THEY ARE USED TO SPECIFY WHICH QUADRANT IN THE PLANE IS AFFECTED. IF NO OPTION IS SPECIFIED, THEN THE ENTIRE PLANE IS AFFECTED. COL AFFECTS THE ENTIRE COLUMN THE POINTER IS AT, BY DEFAULT (*). THE OPTIONS UP AND DOWN CAN BE SPECIFIED TO INDICATE WHICH HALF OF THE COLUMN IS TO BE AFFECTED. SIMILARLY, ROW AFFECTS THE CURRENT ROW TO WHICH THE POINTER IS POINTING. LEFT AND RIGHT CAN BE USED TO SPECIFY WHICH HALF ROW SHOULD BE AFFECTED. THE FORMAT FOR 'VALUE' IS SPECIFIED IN THE DOCUMENTATION FOR 'LABEL'. IF AN ERROR OCCURS, TWO ARROWS POINT TO THE AREA APPROXIMATELY WHERE THE ERROR OCCURRED. WHEN SETTING LABELS WITH THE SET COMMAND, ANY LABELS PREVIOUSLY ASSOCIATED WITH THE DOTS IN THE REGION AFFECTED ARE OVERWRITTEN. IT IS POSSIBLE TO SET A REGION TO THE NULL LABEL. (BY PUTTING AN EQUALS SIGN WITH NO LABELS AFTER IT, THE REGION IS ERASED.)

THE ANALYST WILL WANT TO USE THE SET COMMAND WHEN LARGE REGIONS OF THE GRID HAVE THE SAME LABEL. IF FOR EXAMPLE ALL OF THE GRID TO THE RIGHT OF COLUMN 3 IS WHEAT, WITH THE EXCEPTION OF A ROAD RUNNING DOWN ROW 6, IT IS MUCH QUICKER TO TYPE 'SET 4,1 ALL RIGHT =WH' THAN TO ENTER INPUT MODE AND INPUT THE WHEAT ONE AT A TIME. ONCE A LARGE AREA IS SET, THE 'ADD', 'REM' AND 'CHANGE' COMMANDS CAN BE USED TO ALTER THE LABELS ON THE FEW DOTS WHICH MIGHT BE DIFFERENT.

EXAMPLES:

```
>SET  4,3 ROW LEFT =X U /H
     THIS SETS ROW 4 FROM COLUMN THREE
     TO THE PARTICULAR  LABEL SPECIFIED  AFTER
     THE EQUALS SIGN.
>SET  COL  =X
     THIS SETS THE CURRENT COLUMN TO THE LABEL
     VALUE =X
>SET  ALL  =H SB/BF
     THIS SETS EVERYTHING TO THE  GIVEN VALUE
```

```
>SET ROW 5,4 =HRBF/SU
           ¬¬
SYNTAX ERROR
    THIS DEMONSTRATES HOW AN ERROR IS HANDLED. IN THIS CASE THE
    PROBLEM IS THAT THE ROW,COLUMN COORDINATES SHOULD BE
    SPECIFIED BEFORE THE 'ROW' PARAMETER.
>SET ROW =
    THIS COMMAND SETS THE ENTIRE ROW POINTED TO BY THE CURRENT
    SETTING OF THE INTERNALLY MAINTAINED POINTER TO THE NULL
    LABEL. THAT IS, IT HAS THE SAME EFFECT AS REMOVING EACH LABEL
    IN THE ROW
>
```

CHANGE COMMAND

DESCRIPTION: THIS COMMAND IS USED TO CHANGE ONE LABEL TO ANOTHER.

FORMAT:

```
               (              (      (UP    ) ) )
               (              (      (DOWN  ) ) )
               (              (      (LEFT  ) ) )
               (              (ALL   (RIGHT ) ) )
               (              (      (UL    ) ) )
               (              (      (UR    ) ) )
               (              (      (DL    ) ) )
               (              (      (DR    ) ) )
     CHANGE    (    (XX , YY)                   ) =V1 =V2
               (              (      (UP    ) ) )
               (              (COL   (DOWN  ) ) )
               (              (      (*     ) ) )
               (                            ) ) )
               (              (      (LEFT  ) ) )
               (              (ROW   (RIGHT ) ) )
               (              (      (*     ) ) )
```

NOTES:

THE FORMAT OF VALUE 1 (V1) AND VALUE 2 (V2) ARE GIVEN IN THE DOCUMENTATION FOR 'LABEL'. FOR EACH LABEL IN VALUE 1 THERE MUST BE A CORRESPONDING LABEL IN VALUE 2, TO WHICH THE LABEL IN VALUE ONE IS CHANGED (IF IT IS FOUND.) FOR THIS REASON IT IS ILLEGAL TO TRY TO USE 'CHANGE' TO CHANGE A VALUE TO THE NULL VALUE.

ALL THE ARGUMENTS TO CHANGE ARE OPTIONAL EXCEPT FOR V1 AND V2. IF XX AND YY ARE SPECIFIED, THEY SET THE POINTER TO THAT VALUE. OTHERWISE THE CURRENT VALUE OF THE INTERNALLY MAINTAINED POINTER IS USED. (FOR FURTHER INFORMATION ON THE INTERNALLY MAINTAINED POINTER, SEE THE DOCUMENTATION FOR 'LABEL'.) IF NO OTHER ARGUMENTS ARE SPECIFIED, THE CHANGE COMMAND HAS EFFECT FOR ONLY THE DOT SPECIFIED BY THE POINTER. THE 'ALL' PARAMETER CHANGES THE VALUES OVER THE ENTIRE GRID, UNLESS IT IS MODIFIED BY A FURTHER PARAMETER. THE 'UP', 'DOWN', 'LEFT', AND 'RIGHT' OPTIONS SPECIFY HALF PLANES IN THE GRID, USING THE VALUE OF POINTER AS THE REFERENCE. (NOTE: THE ROW AND/OR COLUMN CONTAINING THE DOT POINTED TO BY THE INTERNALLY MAINTAINED POINTER IS ALSO INCLUDED IN THE AREA AFFECTED.) 'UL', 'UR', 'DL', AND 'DR' ARE ABBREVIATIONS FOR UP-LEFT, UP-RIGHT, DOWN-LEFT, AND DOWN-RIGHT, RESPECTIVELY. THESE OPTIONS SPECIFY QUADRANTS OF THE GRID TO BE AFFECTED BY THE COMMAND. AS WITH THE OTHER OPTIONS THEY ARE REFERENCED BY THE POINTER, AND INCLUDE THE ROW AND COLUMN IN WHICH THE DOT POINTED TO RESIDES. THE 'COL' OPTION SPECIFIES THAT THE COMMAND IS TO HAVE EFFECT ONLY IN THE COL THAT POINTER IS SET TO. BY DEFAULT ('*') THE ENTIRE COLUMN IS CHANGED. THE OPTIONS 'UP' AND 'DOWN' MODIFY THIS TO BE ONLY THE COLUMN ABOVE THE POINTER AND BELOW IT, RESPECTIVELY. THE 'ROW' OPTION HAS A SIMILAR EFFECT, ALLOWING CHANGE TO BE MADE ONLY IN THE ROW POINTED TO. THE OPTIONS 'LEFT' AND 'RIGHT' FURTHER SPECIFY WHICH HALF OF THE ROW IS TO BE CHANGED.

ONE DEFINITE USE FOR THE CHANGE COMMAND IS IN CORRECTING TYPING ERRORS THE ANALYST MADE DURING INPUT MODE. IF A WRONG LABEL VALUE IS ENTERED, THE CHANGE COMMAND CAN BE USED TO CORRECT THE ERROR. ANOTHER POSSIBLE USE OF THE CHANGE COMMAND IS THE FOLLOWING. SUPPOSE THE USER DEFINED HIS SPECIAL CROP CODE S3 TO BE CABBAGES, BECAUSE CABBAGES IS NOT A STANDARD PREDEFINED CROP CODE IN THE CROP CODE FILE. A FEW WEEKS LATER IT IS DECIDED THAT SO MANY ANALYSTS ARE LABELING THE CROP CABBAGE THAT IT IS WORTHWHILE ADDING A NEW PREDEFINED CROP CODE. THE ANALYST CAN THEN USE THE CHANGE COMMAND TO CHANGE ALL OCCURENCES OF S3 TO THE NEW CABBAGE CODE.

EXAMPLES:

```
>CHANGE ALL =X H    =I U
    ALL OF THE DOTS WITH A LABEL OF X WILL
    NOW HAVE A LABEL I IN ITS PLACE, AND
    ALL THOSE WITH H WILL HAVE U INSTEAD.
>CHANGE ALL LEFT =X H I  =I SB
INCOMPATIBLE VALUES
    THIS ERROR OCCURRED BECAUSE THERE WAS NOT A CORRESPONDING
    LABEL IN THE SECOND VALUE FOR THE LABEL 'I' IN THE FIRST
    VALUE.
```

ADD COMMAND

DESCRIPTION: THIS COMMAND ADDS AN ADDITIONAL SET OF LABELS TO THOSE DOTS IN THE REGION SPECIFIED, WITHOUT DESTROYING THE LABELS ALREADY THERE.

FORMAT:

```
          (              (      (UP    ) ) )
          (              (      (DOWN  ) ) )
          (              (      (LEFT  ) ) )
          (              (ALL   (RIGHT ) ) )
          (              (      (UL    ) ) )
          (              (      (UR    ) ) )
          (              (      (DL    ) ) )
          (              (      (DR    ) ) )
ADD       (   (XX , YY)                    ) =VALUE
          (              (      (UP    ) ) )
          (              (COL   (DOWN  ) ) )
          (              (      (*     ) ) )
          (
          (              (      (LEFT  ) ) )
          (              (ROW   (RIGHT ) ) )
          (              (      (*     ) ) )
```

NOTES: THIS COMMAND AUGMENTS THE LABELS ALREADY ASSOCIATED WITH THE SPECIFIED DOTS. IT DOES NOT OVERWRITE ANY LABELS ALREADY THERE, AND IF THE LABELS AS SPECIFIED BY =VALUE ARE MORE ABUNDANT THAN THE AVAILABLE SPACE FOR A GIVEN DOT, THEN ONLY THE FIRST LABELS ARE TAKEN AND ADDED TO THE DOT UNTIL IT IS FILLED. THE FORMAT OF =VALUE IS SPECIFIED IN DETAIL IN THE DOCUMENTATION FOR 'LABEL'. THE REGION TO WHICH THE LABELS ARE ADDED ARE SPECIFIED AS FOLLOWS. IF XX,YY IS PRESENT, THE INTERNALLY MAINTAINED POINTER IS SET TO THE VALUES XX, YY, OTHERWISE THE CURRENT VALUE OF THE POINTER IS USED. WITH NO OTHER OPTIONS, THE LABELS ARE ADDED TO JUST THE DOT AT THE LOCATION SPECIFIED BY THE POINTER. 'ALL' ADDS THE LABELS TO THE ENTIRE GRID, WHILE 'UP', 'DOWN', 'LEFT', AND 'RIGHT' SPECIFY HALF PLANES (RELATIVE TO THE POINTER SETTING) IN WHICH THE DOTS ARE TO HAVE LABELS ADDED. 'UL', 'UR', 'DL', AND 'DR' ARE ABBREVIATIONS FOR UP-LEFT, UP-RIGHT, DOWN-LEFT, AND DOWN-RIGHT RESPECTIVELY, AND THEY ARE USED TO SPECIFY QUADRANTS IN THE GRID. THE 'COL' OPTION ALLOWS LABELS TO BE ADDED ONLY TO THE DOTS IN THE COLUMN SPECIFIED TO BY POINTER, WHILE 'ROW' DOES THE SAME FOR THE ROW SPECIFIED BY THE POINTER. BY DEFAULT THE ENTIRE ROW OR COLUMN IS AFFECTED. THE DIRECTIONAL OPTIONS --- UP, DOWN, ETC. --- SPECIFY WHICH HALF OF THE ROW (OR COLUMN) ARE TO BE AFFECTED BY THE COMMAND. IT IS POSSIBLE AND ENTIRELY PERMITTED TO ADD A DUPLICATE LABEL TO A GIVEN DOT. THE ANALYST MIGHT USE THIS COMMAND IN THE CASE THAT HE HAS FORGOTTEN TO PUT A LABEL ON A DOT WHILE HE WAS IN INPUT MODE. ADD CAN BE USED TO CORRECT THE PROBLEM, BY ADDING THE MISSING LABEL TO THE DOT WITHOUT HAVING TO GO INTO INPUT MODE.

EXAMPLES:

```
>ADD ALL UL =X H /C
     THIS WOULD ADD THE GIVEN DOT LABELS TO
     EVERYTHING IN THE UPPER-LEFT CORNER OF
     THE PLANE. THAT IS, ALL THE ROWS THAT
     ARE GREATER THAN THE CURRENT ONE, AND
     ALL THE COLUMNS WHICH ARE GREATER THAN
     THE CURRENT COLUMN.
>ADD 9,7 =SB
     THIS WOULD ADD THE LABEL TO THE DOT
     IN ROW NINE, COLUMN SEVEN.
```

REM COMMAND

DESCRIPTION: THIS COMMAND REMOVES THE SPECIFIED LABEL(S) FROM ALL THE DOTS IN THE GIVEN AREA.

FORMAT:

```
                (              (       (UP    ) ) )
                (              (       (DOWN  ) ) )
                (              (       (LEFT  ) ) )
                (              (ALL    (RIGHT ) ) )
                (              (       (UL    ) ) )
                (              (       (UR    ) ) )
                (              (       (DL    ) ) )
                (              (       (DR    ) ) )
REM             (   (XX , YY)                     )  =VALUE
                (              (       (UP    ) ) )
                (              (COL    (DOWN  ) ) )
                (              (       (*     ) ) )
                (                                 )
                (              (       (LEFT  ) ) )
                (              (ROW    (RIGHT ) ) )
                (              (       (*     ) ) )
```

NOTES: FOR THE FORMAT OF THE VALUE PARAMETER, SEE THE DOCUMENTATION FOR 'LABEL'. IF NONE OF THE OPTIONS ARE PRESENT, THE SPECIFIED LABELS ARE REMOVED ONLY FROM THE DOT AT THE LOCATION SPECIFIED BY THE CURRENT VALUE OF POINTER. THE OPTION XX,YY CHANGES THE VALUE OF THE POINTER TO HAVE ROW XX AND COLUMN YY. USING THE OPTION ALL, THE ENTIRE GRID IS REMOVED OF THE LABELS SPECIFIED BY =VALUE. IF FURTHER OPTIONS APPEAR, THE AREA AFFECTED IS MODIFIED AS FOLLOWS. THE OPTIONS 'UP', 'DOWN', 'LEFT', AND 'RIGHT' CHANGE THE AREA TO BE THE HALF PLANE IN THE GRID IN THE SPECIFIED DIRECTION FROM THE CURRENT POINTER SETTING. FOR EXAMPLE, IF THE POINTER IS CURRENTLY AT 4,3, THEN 'REM ALL RIGHT =H U ' WILL REMOVE THE LABELS 'H ' AND 'U ' FROM THE HALF PLANE TO THE RIGHT OF COLUMN 3. (COLUMN 3 IS ALSO INCLUDED IN THOSE COLUMNS AFFECTED.) 'UL', 'UR', 'DL', AND 'DR' ARE ABBREVIATIONS FOR UP-LEFT, UP-RIGHT, DOWN-LEFT AND DOWN-RIGHT, AND THESE OPTIONS SPECIFY QUADRANTS IN THE GRID WHICH WILL BE AFFECTED BY THE REMOVE COMMAND. THE TWO OTHER OPTIONS 'COL' AND 'ROW' CAUSE THE AREA OF EFFECT TO BE LIMITED TO THE CURRENT COLUMN OR ROW (RESPECTIVELY) AS POINTED TO BY THE INTERNALLY MAINTAINED POINTER. BY DEFAULT ('*') THE ENTIRE ROW OR COLUMN IS AFFECTED, UNLESS A FURTHER OPTION APPEARS, WHICH SPECIFIES WHICH HALF OF THE ROW OR COLUMN IS TO BE AFFECTED.

A POSSIBLE USE FOR THE REMOVE COMMAND IS TO DELETE MISTAKEN LABELS ADDED TO DOTS. ANOTHER USED IS FOR CHANGING INACCURATE DATA. FOR EXAMPLE, A FARMER DECIDES TO PLOW OVER AN OLD ABANDONED ROAD. THE DOTS WHICH CONTAIN THE 'RD' LABEL WILL NOW HAVE TO HAVE IT REMOVED AS THE FARMER HAS PLANTED CABBAGES THERE, TOO.

EXAMPLES:

```
>REM    ROW  RIGHT  =X H
        THIS WOULD REMOVE THE LABELS X AND H
        FROM ALL THE DOTS IN THE CURRENT ROW THAT
        ARE TO THE RIGHT OF THE CURRENT COLUMN.
```

HOME COMMAND

DESCRIPTION: THIS COMMAND SETS THE POINTER TO THE HOME DOT.

FORMAT: HOME

NOTES: THE DOT EDITOR MAINTAINS A POINTER WHICH KEEPS TRACK OF WHICH ROW AND WHICH COLUMN THE ANALYST IS CURRENTLY PROCESSING. SOME COMMANDS ALLOW THE POINTER TO BE CHANGED, OTHERS LEAVE IT UNCHANGED. IF, AFTER THE POINTER HAS BEEN CHANGED, IT IS NECESSARY TO MOVE IT TO ITS STARTING POSITION (THE HOME DOT), THIS COMMAND DOES IT. THE HOME DOT IS IN THE UPPER LEFT HAND CORNER OF THE GRID, WITH COORDINATES: ROW 1, COL 1.

EXAMPLES:

```
>PTR
ROW: 6
COL: 9
>HOME
>PTR
ROW: 1
COL: 1
>
```

UP COMMAND

DESCRIPTION: THIS COMMAND MOVES THE POINTER UP NNN LINES.

FORMAT: UP (NNN)

NOTES: IF NNN IS NOT SPECIFIED, IT IS DEFAULTED TO 1. THE 'UP' COMMAND ALWAYS AFFECTS THE ROW HALF OF THE INTERNALLY MAINTAINED POINTER (SEE THE DOCUMENTATION FOR 'LABEL' FOR FURTHER DETAILS ABOUT THE POINTER). THIS COMMAND IS USED TO POSITION THE POINTER AFTER IT HAS BEEN MOVED BY OTHER COMMANDS. IN SOME CASES IT MIGHT BE EASIER TO SPECIFY MOVING UP ONE ROW, THAN TO SUPPLY BOTH COLUMN AND ROW COORDINATES.

EXAMPLES:

```
>PTR
ROW:  3
COL:  1
>UP  2
>PTR
ROW:  1
COL:  1
>
```

DOWN COMMAND

DESCRIPTION: THIS COMMAND MOVES THE POINTER DOWN BY NNN LINES.

FORMAT: DOWN (NNN)

NOTES: IF NNN IS NOT SPECIFIED, IT IS DEFAULTED TO 1. THE 'DOWN' COMMAND ALWAYS AFFECTS THE ROW HALF OF THE INTERNALLY MAINTAINED POINTER (SEE THE DOCUMENTATION FOR 'LABEL' FOR FURTHER DETAILS ABOUT THE POINTER). THIS COMMAND IS USED FOR POSITIONING THE POINTER AFTER IT HAS BEEN MOVED BY OTHER COMMANDS. IT MIGHT BE MORE CONVENIENT IN SOME CASES TO SPECIFY MOVING DOWN ONE ROW, THAN TO SUPPLY BOTH ROW AND COLUMN COORDINATES.

EXAMPLES:

```
>PTR
ROW:  3
COL:  5
>DOWN
>PTR
ROW:  4
COL:  5
>D 6
>PTR
ROW: 10
COL:  5
>
```

LEFT COMMAND

DESCRIPTION: THIS COMMAND MOVES THE POINTER NNN COLUMNS TO THE LEFT.

FORMAT: LEFT (NNN)

NOTES: IF NNN IS NOT SPECIFIED, IT IS DEFAULTED TO 1. THE 'LEFT' COMMAND ALWAYS AFFECTS THE COLUMN HALF OF THE INTERNALLY MAINTAINED POINTER (SEE THE DOCUMENTATION FOR 'LABEL' FOR MORE INFORMATION ABOUT THE POINTER). THIS COMMAND IS USED IN POSITIONING THE POINTER AFTER IT HAS BEEN MOVED BY OTHER COMMANDS. IN SOME CASES IT MIGHT BE MORE DESIRABLE TO SPECIFY MOVING NNN ROWS, THAN TO SUPPLY SPECIFIC COLUMN AND ROW COORDINATES.

EXAMPLES:

```
>PTR
ROW:  4
COL:  4
>LEFT 2
>PTR
ROW:  4
COL:  2
>
```

RIGHT COMMAND

DESCRIPTION: THIS COMMAND MOVES THE POINTER NNN COLUMNS TO THE RIGHT.

FORMAT: RIGHT (NNN)

NOTES: IF NNN IS NOT SPECIFIED, IT IS DEFAULTED TO 1. THE 'RIGHT' COMMAND ALWAYS AFFECTS THE COLUMN HALF OF THE INTERNALLY MAINTAINED POINTER (SEE THE DOCUMENTATION FOR 'LABEL' FOR FURTHER DETAILS ABOUT THE POINTER). THIS COMMAND IS USED FOR POSITIONING THE POINTER AFTER IT HAS BEEN MOVED BY OTHER COMMANDS. IT MIGHT BE MORE CONVENIENT IN SOME INSTANCES TO MOVE ONE OR SEVERAL ROWS TO THE RIGHT, THAN TO SPECIFY BOTH ROW AND COLUMN COORDINATES.

EXAMPLES:

```
>PTR
ROW: 11
COL:  9
>RIGHT 2
>PTR
ROW: 11
COL: 11
>R 3
>PTR
ROW: 11
COL: 14
>
```

TOP COMMAND

DESCRIPTION: THIS COMMAND MOVES THE POINTER TO ROW ONE.

FORMAT: TOP

NOTES: THIS COMMAND ALWAYS AFFECTS THE ROW HALF OF THE INTERNALLY MAINTAINED POINTER. (SEE THE DOCUMENTATION FOR 'LABEL' FOR FURTHER INFORMATION ON THE INTERNALLY MAINTAINED POINTER.) IT CAN BE ISSUED AT ANY TIME DURING LABEL EDITING, EXCEPT DURING INPUT MODE. IT MIGHT BE USED TO POSITION THE POINTER BEFORE ISSUING THE TYPE COMMAND OR ANY OTHER COMMAND WHICH USES THE POINTER.

EXAMPLES:

```
>PTR
ROW:  5
COL:  3
>TO
>PTR
ROW:  1
COL:  3
>
```

BOTTOM COMMAND

DESCRIPTION: THIS COMMAND MOVES THE POINTER TO THE LAST ROW IN THE FILE.

FORMAT: BOTTOM

NOTES: THIS COMMAND ALWAYS AFFECTS THE ROW HALF OF THE INTERNALLY MAINTAINED POINTER. (SEE THE DOCUMENTATION FOR 'LABEL' FOR FURTHER INFORMATION ON THE INTERNALLY MAINTAINED POINTER.) IT CAN BE ISSUED AT ANY TIME DURING LABEL EDITING, EXCEPT DURING INPUT MODE. IF THE FILE IS A BIG DOT FILE, THEN 'BOTTOM' SETS THE LAST ROW TO ROW 11. FOR A SMALL DOT FILE, THE LAST ROW IS SET TO ROW 22.

EXAMPLES:

```
>PTR
ROW: 13
COL:  4
>BOT
>PTR
ROW: 22
COL:  4
>
```

FIRST COMMAND

DESCRIPTION: THIS COMMAND MOVES THE POINTER TO THE FIRST COLUMN IN THE ROW.

FORMAT: FIRST

NOTES: THIS COMMAND ALWAYS AFFECTS THE COLUMN HALF OF THE INTERNALLY MAINTAINED POINTER. (SEE THE DOCUMENTATION FOR 'LABEL' FOR FURTHER INFORMATION ON THE INTERNALLY MAINTAINED POINTER.) IT CAN BE ISSUED AT ANY TIME DURING LABEL EDITING, EXCEPT DURING INPUT MODE.

EXAMPLES:

```
>PTR
ROW:  3
COL: 18
>FIRST
>PTR
ROW:  3
COL:  1
>
```

LAST COMMAND

DESCRIPTION: THIS COMMAND MOVES THE POINTER TO THE LAST COLUMN IN THE CURRENT ROW OF THE FILE.

FORMAT: LAST

NOTES: THIS COMMAND ALWAYS AFFECTS THE COLUMN HALF OF THE INTERNALLY MAINTAINED POINTER. (SEE THE DOCUMENTATION FOR 'LABEL' FOR FURTHER INFORMATION ON THE INTERNALLY MAINTAINED POINTER.) IT CAN BE ISSUED AT ANY TIME DURING LABEL EDITING, EXCEPT DURING INPUT MODE.

EXAMPLES:

```
>PTR
ROW:  3
COL:  5
>LAST
>PTR
ROW:  3
COL: 19
>
```

MODE COMMAND

DESCRIPTION: THIS COMMAND IS USED TO DISPLAY OR CHANGE THE CURRENT OPERATING MODE. UPON ENTERING THE EDITOR, ALL PROCESSING DEFAULTS TO OPERATIONS ON ROWS. IF FOR SOME REASON IT IS MORE DESIRABLE TO PROCESS THE DOTS BY COLUMNS, THIS COMMAND ALLOWS THE CHANGE.

FORMAT:

```
MODE      (ROW)
          (COL)
```

NOTES: IF NO OPTION IS SPECIFIED, THE CURRENT MODE SETTING IS DISPLAYED. IF THE OPTION IS ALREADY SET TO THE ONE REQUESTED, THERE IS NO CHANGE, OTHERWISE THE MODE IS CHANGED IMMEDIATELY.

EXAMPLES:

```
>MODE
ROW MODE
>MODE      COL
>MODE
COLUMN MODE
>
```

INPUT COMMAND

DESCRIPTION: THIS COMMAND ALLOWS THE USER TO ENTER A CONTINUOUS STREAM OF COORDINATES AND LABELS.

FORMAT: INPUT

NOTES: WHILE IN INPUT MODE, IT IS NOT PERMITTED TO ENTER ANY OTHER COMMAND. THE FORMAT FOR INPUT MODE CONSISTS OF TWO COORDINATES SEPARATED BY A COMMA, AND AN EQUALS SIGN FOLLOWED IMMEDIATELY BY THE LIST OF LABELS. ONLY ONE SUCH PAIRING IS ALLOWED PER LINE. IF THE COORDINATES DEFINE A POINT NOT ALREADY IN THE FILE, IT IS CREATED; OTHERWISE, THE POINTER IS MOVED TO THE EXISTING POINT. INPUTTING ONTO A POINT WITH LABELS ALREADY THERE, OVERWRITES THOSE LABELS WITH THE NEW ONES. TO EXIT FROM INPUT MODE, TYPE A NULL LINE. (HIT THE RETURN WITH NOTHING AFTER THE PROMPT) THE ONLY WAY TO CREATE OFFGRID POINTS IS IN INPUT MODE

EXAMPLES:

```
>INPUT
OFFGRID INPUT MODE
ENTER THE COORDINATES (X,Y) AND THEN THE VALUE
(HIT RETURN TO EXIT)
>5,6 =H I X     ,    6,8 =H I X
                ¬
SYNTAX ERROR

      THE ERROR HERE IS THAT ONLY ONE PAIR OF COORDINATES AND
      SET OF LABELS ARE ALLOWED ON A SINGLE LINE.

ENTER THE COORDINATES (X,Y) AND THEN THE VALUE
(HIT RETURN TO EXIT)
>5,6 =H I X
>6,8 =U FXWA
>90 , 121 =U X I

      THE THREE VALUES WERE INPUT SUCCESSFULLY

>
OFFGRID:
>TYPE *
COORDINATES      LABELS
   90,121        U X I /
  220, 89        C SY/RD
  255,244        G I /
      ALL THE OFFGRID POINTS ARE TYPED FROM WHERE THE POINTER
      IS SET. IT REMAINS SET TO THE LAST OFFGRID POINT INPUT
      WHILE IN INPUT MODE.
>TOP
      THIS MOVES US TO THE FIRST OFFGRID POINT IN THE FILE.
      IT IS NOT NECESSARILY THE POINT 1,1 UNLESS IT HAD BEEN
      ENTERED.
>TYPE *
COORDINATES      LABELS
    5,  6        H I X /
    6,  8        U FXWA/
   13, 42        /H T C H
   64, 64        X RI/U C  S3S3S5
   90,121        U X I /
  220, 89        C SY/RD
  255,244        G I /
>
OO 1 FORMAT:
```

TYPE COMMAND

DESCRIPTION: TYPE IS USED TO DISPLAY THE COORDINATES OF THE OFFGRID POINTS AND THEIR LABELS.

FORMAT:

```
(XXX,YYY)    TYPE       (  *   )
                        ( NNN  )
                        (  1   )
                           -
```

NOTES: IF THE OPTIONAL TWO COORDINATES ARE SPECIFIED, THE POINTER IS MOVE TO THAT POINT, AND TYPING STARTS THERE. IF A POINT WITH THOSE COORDINATES DOES NOT EXIST, IT IS CONSIDERED AN ERROR.
WHEN THE NUMBER NNN IS SPECIFIED, TYPE DISPLAYS THAT MANY POINTS, STARTING FROM WHERE THE POINTER IS CURRENTLY SET.
TYPE DEFAULTS TO DISPLAYING ONE OFFGRID POINT.
TO GET A LISTING OF ALL THE POINTS FROM THE CURRENT LINE TO THE EN OF THE FILE, 'TYPE *' IS USED. AFTER DISPLAYING THE POINTS, THE CURRENT LINE IS LEFT UNCHANGED. THE INTERNALLY MAINTAINED POINTER STILL POINTS TO THE SAME OFFGRID POINT.

EXAMPLES:

```
>13,42 TYPE 2
COORDINATES     LABELS
    13, 42      /H T C H
    64, 64      X RI/U C  S3S3S5
        THIS TYPED OUT TWO LINES STARTING AT THE POINT 13,42
>1,1 TYPE *
COORDINATES     LABELS
     1,  1      SW/
     2,  4      G /
     6,  5      W I T C H /
     6,  6      W I T C H /I T
     8,  7      T SYU C /U I FXWA
    13, 42      /H T C H
    64, 64      X RI/U C  S3S3S5
   220, 89      C SY/RD
   255,244      G I /
   255,255      G U X I A G /
        THIS TYPED OUT ALL OF THE POINTS STARTING FROM THE POINT 1,1
        THE FIRST POINT MAY NOT NECESSARILY BE 1,1.
>255,244 TYPE *
COORDINATES     LABELS
   255,244      G I /
   255,255      X I X R I X /
OO 1 FORMAT:
```

(XXX,YYY) COMMAND

DESCRIPTION: THIS COMMAND SETS THE POINTER TO THE OFFGRID POINT WITH THE SPECIF COORDINATES.

FORMAT: XXX , YYY

NOTES: IF THERE IS NO OFFGRID POINT WITH THESE COORDINATES, THEN IT IS CONSIDERED TO BE AN ERROR. IT IS NOT NECESSARY FOR A COMMAND TO FOLLOW THE COORDINATE SPECIFICATIONS, ALTHOUGH A SELECTED FEW MAY. IN THE EVENT THAT NO COMMAND FOLLOWS THE COORDINATES, THE OFFGRID POINT WITH THE COORDINATES IS DISPLAYED ON THE TERMINAL

EXAMPLES:

```
>5,13
COORDINATES   LABELS
    5, 13        H U / T C H
```

HERE THE COORDINATES ARE SET TO 5,13. SINCE NO COMMAND FOLLOWS, THE POINT IS DISPLAYED.

```
>3,5 SET =IT
```

THE COORDINATES ARE SET TO 3,5 AND THEN THE COMMAND 'SET' IS EXECUTED. SET PRODUCES NO OUTPUT

```
>
```

SET CCMMAND

DESCRIPTION: THIS COMMAND SETS THE LABELS FOR AN OFFGRID POINT TO THE GIVEN LIST.

FORMAT: (XXX,YYY) SET =LABEL-LIST

NOTES: THE OPTIONALLY SPECIFIED COORDINATES MOVE THE INTERNALLY MAINTAINED POINTER TO THE OFFGRID POINT WITH THOSE COORDINATES. IF THE POINT DOES NOT EXIST, IT IS CONSIDERED AN ERROR. THE LABEL LIST OF THE POINT IS SET TO THE LABEL LIST SPECIFIED ON THE COMMAND LINE. IF THERE ARE ALREADY ANY LABELS IN THE OFFGRID POINT'S LABEL LIST, THEY ARE OVERWRITTEN. ALSO, IF THE DOT FILE BEING EDITED IS A SMALL DOT FILE AND BIG DOTS ARE SPECIFIED IN THE LABEL-LIST OF THE COMMAND, THEY ARE IGNORED WHEN BEING ATTACHED TO THE OFFGRID POINT.

EXAMPLES:

>5,8 SET =WARDSY

THIS COMMAND MOVED THE POINTER TO THE OFFGRID POINT 5,8. AND THEN SET THE LABELS TO 'WA', 'RD' AND 'SY'. IF THERE WERE ANY LABELS ON 5,8 THEY ARE NOW SET TO THE LABELS IN THE SPECFIED LIST.

CHANGE COMMAND

DESCRIPTION: THE CHANGE COMMAND CHANGES THE LABELS ASSOCIATED WITH AN OFFGRID POINT TO A DIFFERENT SET OF LABELS. FOR EACH LABEL THAT THE USER WISHES TO CHANGE, THERE SHOULD BE A CORRESPONDING LABEL IN THE SECOND ARGUMENT.

FORMAT: (XXX,YYY) CHANGE =LABEL-LIST1 =LAPEL-LIST2

NOTES: THE LABEL LISTS SHOULD BE OF THE CORRECT TYPE FOR THE FILE. THAT IS, IF THE FILE IS A BIG DOT FILE, THE LABELS SHOULD BE BIG DOT LABELS, IF SMALL DOT, THE LABELS SHOULD BE SMALL DOT. IF BIG DOT LABELS ARE SPECIFIED AND THE FILE IS A SMALL DOT FILE, THE BIG DOT LABELS WILL BE IGNORED, AND THE CHANGE PERFORMED ON THE SMALL DOT LABELS ONLY. THE CHANGE COMMAND AFFECTS ONLY THE OFFGRID POINT AT THE CURRENT LOCATION IN THE FILE. AS WITH MOST OF THE OFFGRID COMMANDS, THE COORDINATES XXX,YYY MAY BE SPECIFIED. IF THEY APPEAR, THE POINTER WILL MOVE TO THAT OFFGRID POINT BEFORE CHANGING THE LABELS. IT IS AN ERROR IF THE POINT AT XXX,YYY DOES NOT EXIST.

EXAMPLES:

```
>5,17 CHANGE =SBBF    =F I
```

PRECEDING THE COMMAND, THE TWO NUMBERS SET THE POINTER TO THE OFFGRID POINT WITH THOSE COORDINATES. THEN THE LABELS 'SB' AND 'BF' ARE CHANGED TO 'F' AND 'I' RESPECTIVELY.

```
>CHANGE =F I X      =U WA
INCOMPATIBLE LABEL PAIRINGS
```

THIS COMMAND CAUSED THE ERROR BECAUSE THERE WAS NO CORRESPONDING VALUE FOR THE LABEL 'X ' IN THE SECOND LIST. IT IS NOT PERMITTED TO CHANGE A LABEL TO THE NULL LABEL.

```
>
```

ADD COMMAND

DESCRIPTION: THIS COMMAND ADDS LABELS TO AN OFFGRID POINT.

FORMAT: (XXX,YYY) ADD =LABEL-LIST

NOTES: ONLY AS MANY LABELS ARE ADDED AS WOULD FILL THE POINT. IF, FOR EXAMPLE, A POINT ALREADY HAS THREE LABELS AND THE USER IS TRYING TO ADD SEVEN MORE, THEN ONLY THE FIRST FIVE LABELS IN THE 'LABEL-LIST' ARE ADDED TO THE POINT. NO ERROR OR WARNING MESSAGE IS PRINTED.

AS WITH SOME OTHER OFFGRID COMMANDS, 'ADD' MAY BE PRECEDED BY TWO COORDINATES TO MOVE THE POINTER TO A DESIRED OFFGRID POINT. ONLY ONE POINT IS AFFECTED BY THE 'ADD' COMMAND. ALSO, IT IS POSSIBLE FOR A LABEL TO APPEAR MORE THAN ONCE ON AN OFFGRID POINT.

EXAMPLES:

```
>TYPE

COORDINATES    LABELS

    5, 13        I SB/X F

>ADD =X WAU /C SYS3

>TYPE

COORDINATES    LABELS

    5, 13        I SBX WAU /X F C

>
```

REM COMMAND

DESCRIPTION: THIS COMMAND IS USED TO REMOVE LABELS FROM A POINT.

FORMAT: (XXX,YYY) REM =LABEL-LIST

NOTES: THE OPTIONAL COORDINATES APPEARING BEFORE THE COMMAND SPECIFY FROM WHICH OFFGRID POINT THE LABELS ARE TO BE REMOVED. OTHERWISE, THE LABELS ARE REMOVED FROM THE OFFGRID POINT POINTED TO BY THE CURRENT SETTING OF THE INTERNALLY MAINTAINED POINTER. IF THE POINT WITH THOSE COORDINATES DOES NOT EXIST, IT IS CONSIDERED AN ERROR. 'REM' AFFECTS ONLY ONE POINT AT A TIME. ALL THE LABELS THAT ARE BOTH IN THE SPECIFIED LABEL-LIST AND IN THE POINT'S OWN LABEL-LIST ARE REMOVED. IF THE FILE IS SMALL DOT AND BIG DOT LABELS ARE SPECIFIED, THEY ARE IGNORED.

EXAMPLES:

```
>TYPE

COORDINATES   LABELS

    5, 13       I SBX WAU /X F C

>REM =X I /X

>TYPE

COORDINATES   LABELS

    5, 13       SBWAU /F C

>
```

TOP COMMAND

DESCRIPTION: THIS COMMAND POSITIONS THE POINTER TO THE FIRST OFFGRID POINT IN THE LIST.

FORMAT: TOP

NOTES: AFTER THE 'TOP' COMMAND HAS BEEN EXECUTED, THE OFFGRID POINT NOW POINTED TO IS DISPLAYED ON THE TERMINAL.

EXAMPLES:

```
>TOP

COORDINATES   LABELS

   1,  1        H I /T H

>
```

BOTTOM COMMAND

DESCRIPTION: THIS COMMAND POSITIONS THE POINTER TO THE LAST OFFGRID POINT IN THE LIST.

FORMAT: BOTTOM

NOTES: AFTER MOVING THE POINTER, THE BOTTOM OFFGRID POINT IS DISPLAYED ON THE TERMINAL.

EXAMPLES:

```
>BOTTOM

COORDINATES   LABELS

  215,215      H U T C H

>
```

UP COMMAND

DESCRIPTION: THIS COMMAND MOVES THE POINTER UP A NUMBER OF LINES.

FORMAT: UP (NNN)

NOTES: IF NNN IS NOT SPECIFIED, IT DEFAULTS TO 1. THE POINTER IS MOVED UP NNN LINES. IF THE POINTER IS MOVED BEYOND THE BEGINNING OF THE FILE, IT IS AUTOMATICALLY RESET TO THE FIRST ITEM IN THE OFFGRID FILE. AFTER MOVING UP, THE LINE TO WHICH THE POINTER IS SET, IS AUTOMATICALLY PRINTED OUT.

EXAMPLES:

```
>TYPE *

COORDINATES   LABELS

  244,244      G I R /A F F

>UP 3

COORDINATES   LABELS

     1,  1     H U T

>TYPE *

COORDINATES   LABELS

     1,  1     H U T

    40, 40     W A T /C H

   244,244     G I R /A F F

>
```

DOWN COMMAND

DESCRIPTION: THIS COMMAND MOVES THE POINTER DOWN A NUMBER OF POINTS IN THE USER'S OFFGRID LIST.

FORMAT: DOWN (NNN)

NOTES: IF NNN IS NOT SPECIFIED, IT DEFAULTS TO ONE. IF THE END OF THE FILE IS REACHED IN MOVING DOWN NNN LINES, THE POINTER IS SET TO THE LAST OFFGRID POINT IN THE FILE. AFTER MOVING THE POINTER, THE NEW LINE IS AUTOMATICALLY PRINTED OUT.

EXAMPLES:

```
>TYPE *

COORDINATES   LABELS

    1,  1       H U T

   40, 40       W A T /C H

  244,244       G I R /A F F

>DOWN 789

COORDINATES   LABELS

  244,244       G I R /A F F

>
```

DEL COMMAND

DESCRIPTION: THIS COMMAND IS USED TO DELETE OFFGRID POINTS FROM THE FILE.

FORMAT: (XXX,YYY) DEL (NNN)

NOTES: WHEN THE NUMBER NN IS SPECIFIED, THEN THAT MANY OFFGRID POINTS WILL BE DELETED STARTING AT THE CURRENT DOT. THE DEFAULT IS ONE, IF NO NUMBER IS SPECIFIED. AFTER POINTS HAVE BEEN DELETED, THE INTERNAL POINTER IS SET TO THE FIRST OFFGRID POINT AFTER THE LAST DELETED POINT. IF THE LAST OFFGRID POINT HAS BEEN DELETED THEN THE POINTER IS SET TO ZERO, A SPECIAL VALUE WHICH INDICATES THAT THERE ARE CURRENTLY NO OFFGRID POINTS IN THE FILE. THE COORDINATES XXX AND YYY SPECIFY AT WHICH OFFGRID POINT THE DELETING IS TO BEGIN. IT IS CONSIDERED AN ERROR IF THE POINT XXX, YYY DOES NOT EXIST.

EXAMPLES:

```
>TYPE *
COORDINATES   LABELS
   5, 13      SBWAU /F C
  11, 24      F X I /C
  11,103      S3/S7I WA
 124,101      SBWAU
 125,101      G I /R A F F
 210,210      W A /T C H
> 11,24 DEL 3
     THIS COMMAND MOVES THE POINTER TO 11,24 AND THEN
     DELETES THREE OFFGRID POINTS.
>TOP
>TYPE *
COORDINATES   LABELS
   5, 13      SBWAU /F C
 125,101      G I /R A F F
 210,210      W A /T C H
>
```

SAMPLE TERMINAL SESSION

IN THE FOLLOWING SAMPLE TERMINAL SESSION, THERE ARE SEVERAL IMPORTANT THINGS TO NOTE. FIRST OF ALL, THE USER SHOULD GET THE JSCDISK 19A. THIS DISK CONTAINS ALL OF THE ROUTINES IN WHICH THE ANALYST IS INTERESTED.

THE NEXT THING TO NOTE IS THE USE OF THE 'DOTINIT' PROGRAM, WHICH IS USED TO INITIALIZE A DOT FILE. IN THE SESSION, A NEW FILE IS CREATED. HOWEVER, THE PROGRAM ALSO ALLOWS A USER TO QUERY THE DATA BASE TO FIND OUT WHAT FILES ARE THERE, AND THEN GET A COPY OF ANY ONE OF THOSE FILES.

THE NEXT STEP IS TO 'DOTEDIT' THE FILE AND MAKE ANY CHANGES NECESSARY. IN THE SAMPLE SESSION, A DOT FILE FROM TIM WHITE IS INPUT AS AN EXAMPLE. FIRST THE HEADER INFORMATION IS ENTERED, THEN THE LABELS FOR ALL THE DOTS IN THE GRID ARE INPUT.

FINALLY THE USER RUNS FORMS TO GENERATE A HARD COPY OUTPUT OF THE DOT FILE. AFTER THE SAMPLE SESSION IS A COPY OF THE OUTPUT PRODUCED BY FORMS. THEN THE USER SUBMITS THE FILE TO BE INSERTED INTO THE DATABASE. ONLY AFTER HE HAS MADE SURE THAT IT IS CORRECT AND THAT IT MEET HIS OWN SPECIFICATIONS. ONCE THE FILE IS IN THE DATABASE, IT IS DIFFICULT TO DELETE IT, SO THAT THE ANALYST SHOULD ONLY SUBMIT THE FILES AFTER HE IS SURE THAT THEY ARE CORRECT.

```
.GETDISK JSCDISK 19A
JSCDISK  19A HAS BEEN ATTACHED AS 19A.
DMSACC723I F (19A) R/O
19A HAS BEEN ACCESSED AS F  DISK.
R;

.DOTINIT
INITALIZATION OF USER'S DISK WITH A NEW DOT FILE OR A DOT FILE IN THE RT&E
DATA BASE.  INPUT THE FILENAME, FILETYPE, (FILEMODE) THAT THE DOT FILE
IS TO BE CALLED.  THE FILEMODE IS DEFALTED TO 'A'.  IF FILEMODE IS DIFFERENT
THAN 'A', INPUT CORRECT FILEMODE.
.DEMO BIGDOTS
DMSLIO740I EXECUTION BEGINS...
 IF THE DOT FILE IS IN RT&E DATA BASE TYPE 'OLD'.
 IF THE DOT FILE IS NOT IN RT&E DATA BASE TYPE 'NEW'.
 TYPE 'STOP' TO STOP DOTINIT.
.NEW
IF THE FILE BEING CREATED IS A SMALL DOT FILE TYPE '1'
IF THE FILE BEING CREATED IS A BIG DOT FILE TYPE '2'
.2
 THE FILE IS INITALIZED.
R;


.DOTEDIT
 EDITING OF DOT FILES.
 INPUT THE FILENAME AND FILETYPE OF THE DOT FILE
 (FILEMODE IS DEFAULTED TO A, IF YOU NEED A DIFFERENT FILEMODE  INPUT IT)
 SEPARATE EACH BY A SPACE.
.DEMO BIGDOTS
DMSLIO740I EXECUTION BEGINS...
EDIT DOT HEADER
INPUT A COMMAND
.HELP

EDIT FIELDS AND THEIR ABBREVATIONS.
SEGMENT NUMBER...........................SEGNUM
CROP YEAR................................CROPYR
DATE OF LABELING.........................LABDATE
ANALYST NAME.............................NAME
ANALYST IDENTIFIER.......................ID
ACQUISITION USED IN LABELING.............ACQ (I1)
LABELING CONVENTION......................LC
DATE AIRCRAFT IMAGERY ACQUIRED...........ACDATE
DATE INVENTORY MADE OR RECEIVED AT JSC...JSCDATE
SPECIAL CROP CODES.......................SC (I1)
EXPERIMENT NUMBER........................EXPNUM
USER COMMENTS............................COMMENT
DATA INPUT TYPE..........................DITYPE
DATA ENTRY CLERK NAME....................DENAME
GRID NUMBER 1............................GRID1
GRID NUMBER 2............................GRID2

(HIT RETURN KEY FOR THE LISTING OF VALID COMMANDS)
.
COMMANDS FOR EDITING DOT HEADER
SAVE                      SAVES THE FILE AND STOPS EDITING
QUIT                      STOPS EDITING, FILE NOT SAVED
LABEL                     PUTS YOU INTO EDITING THE LABELS
OFFGRD                    PUTS YOU INTO EDITING OFFGRID
HELP                      LISTS EDIT FIELDS AND COMMANDS
INPUT (FIELD)             ALLOWS YOU TO ENTER THE DATA STARTING AT THE FIELD
TYPE                      TYPE OUT THE ENTIRE HEADER
CODES                     LISTS AVAILABLE CROP CODES
(CHANGE) FIELD (VALUE) CHANGE THE VALUE OF THE FIELD

INPUT A COMMAND
.INPUT
INPUT PROMPTED DATA
ENTER SEGMENT NUMBER (0 - 9999)
```

```
.200
ENTER CROP YEAR (75 - 99) OR (1975 - 1999)
.78
ENTER DATE OF LABELING (MM DD YY)
(ONLY A SPACE CAN APPEAR BETWEEN EACH NUMBER)
.5  30 80
ENTER ANALYST NAME (A24)
.JOHN P. DOLAN
ENTER ANALYST IDENTIFIER (I4)
.190
ENTER ACQUISITION DATE 1 (YYDDD)
.78234
ENTER ACQUISITION DATE 2 (YYDDD)
.
ENTER LABELING CONVENTION (I4)
.
EDIT DOT HEADER
INPUT A COMMAND
.TYPE
DOT HEADER RECORD
SEGMENT NUMBER..........................    200
CROP YEAR...............................     78
DATE OF LABELING........................  80151
ANALYST NAME............................JOHN P. DOLAN
ANALYST IDENTIFIER......................    190
ACQUISITION DATES USED     DATE 1.........  78234
                           DATE 2.........      0
                           DATE 3.........      0
                           DATE 4.........      0
                           DATE 5.........      0
                           DATE 6.........      0
LABELING CONVENTION.....................      0
DATE AIRCRAFT IMAGERY ACQUIRED..........      0
DATE INVENTORY MADE OR RECEIVED AT JSC...     0
DATA INPUT TYPE.........................
EXPERIMENT NUMBER.......................      0
USER COMMENTS :

DATA ENTRY CLERK NAME...................
GRID NUMBER 1...........................      0
GRID NUMBER 2...........................      0

(HIT THE RETURN KEY FOR THE SPECIAL CROP CODES)

.INPUT ACDATE
INPUT PROMPTED DATA
ENTER DATE AIRCRAFT IMAGERY ACQUIRED (YYDDD)
.78275
ENTER DATE INVENTORY MADE OR RECEIVED AT JSC (YYDDD)
.78340
ENTER SPECIAL CROP CODE SO (A16)
.
ENTER EXPERIMENT NUMBER (I4)
.001
ENTER USER COMMENTS (UP TO 72 CHARACTORS)
.THIS IS A SAMPLE TERMINAL SESSION ON THE DOT EDITOR
EDIT DOT HEADER
INPUT A COMMAND
.TYPE
DOT HEADER RECORD
SEGMENT NUMBER..........................    200
CROP YEAR...............................     78
DATE OF LABELING........................  80151
ANALYST NAME............................JOHN P. DOLAN
ANALYST IDENTIFIER......................    190
ACQUISITION DATES USED     DATE 1.........  78234
                           DATE 2.........      0
                           DATE 3.........      0
                           DATE 4.........      0
                           DATE 5.........      0
```

```
                                DATE 6.........      0
LABELING CONVENTION.....................             0
DATE AIRCRAFT IMAGERY ACQUIRED...........        78275
DATE INVENTORY MADE OR RECEIVED AT JSC...        78340
DATA INPUT TYPE.........................
EXPERIMENT NUMBER.......................             1
USER COMMENTS :
 THIS IS A SAMPLE TERMINAL SESSION ON THE DOT EDITOR
DATA ENTRY CLERK NAME...................
GRID NUMBER 1...........................             0
GRID NUMBER 2...........................             0

(HIT THE RETURN KEY FOR THE SPECIAL CROP CODES)

.LABEL

EDITING LABEL FIELDS
.HELP
LIST OF LABEL EDITING COMMAND SYNTAX AND ABBREVIATIONS

AD(D) AREA    =VALUE        ADDS THE LABELS TO THE REGION
B(OTTOM)                    SET PTR TO ROW 22 (11 FOR BIG DOT)
C(HANGE) AREA =V1 =V2       CHANGES VALUE 1 TO VALUE 2 IN REGION
CO(DES)                     LIST VALID CROP CODES AND ABBREV.
D(OWN)        - N -         MOVE POINTER DOWN N ROWS
F(IRST)                     SET POINTER TO COLUMN 1
HEADER                      TRANSFERS TO THE HEADER EDITOR
HE(LP)                      TYPES THIS MESSAGE
H(OME)                      MOVES THE POINTER TO 1,1
I(NPUT)       -X (,Y)-      INPUT LABELS STARTING AT X,Y
LA(ST)                      MOVE POINTER TO COLUMN 19
L(EFT)        - N -         MOVES POINTER LEFT N COLUMNS
M(ODE)        - ROW -       SETS MODE TO ROW PROCESSING
              - COL -       SETS MODE TO COLUMN PROCESSING
              -     -       DISPLAYS CURRENT MODE SETTING
OFFGRID                     TRANSFERS TO OFFGRID POINTS EDITOR
P(TR)                       DISPLAYS CURRENT POINTER SETTING
QUIT                        HALTS EXECUTION IMMEDIATELY
RE(M) AREA    =VALUE        REMOVES THE LABEL FROM THE REGION
R(IGHT)       - N -         MOVES POINTER RIGHT N COLUMNS
SAVE                        SAVES FILE AND EXITS
SE(T) AREA    =VALUE        SETS A REGION TO 'VALUE'
TO(P)                       SETS THE POINTER TO ROW 1
T(YPE)        - N -         DISPLAYS N LINES (COLUMNS) OF LABELS
U(P)          - N -         MOVES POINTER UP N ROWS
X(,Y)                       SETS POINTER TO DOT AT X,Y
.INPUT
ENTERING ROW:  1
STARTING AT COLUMN  1
.ICRDC /P .C /X U .SYU .SY/RDC .SY/C .H /C .SY/RD .SYRD.P P /A SY
ENTERING ROW:  1
STARTING AT COLUMN 10
.C .SY/X .SYC .SY.IC/H SY.C /C .C .P C /SY.T /SY.T .
ENTERING ROW:  2
STARTING AT COLUMN  1
.C /SY.C /SY.C .C .C /SY.SY.C /RD.C .C .H /RDC P .C /H .SY.C /OF.C X
ENTERING ROW:  2
STARTING AT COLUMN 15
.C /SY.C RDX U .SY.SYC .C RDSY.
ENTERING ROW:  3
STARTING AT COLUMN  1
.C /RDSY RDSY.SYX .C .C /C .SY/C .C /C .SY/RDP .SY/RD H C
ENTERING ROW:  3
STARTING AT COLUMN 10
.C .
ENTERING ROW:  3
STARTING AT COLUMN 11
.C SY. SY/C .C .C .C /RD
ENTERING ROW:  3
STARTING AT COLUMN 16
```

```
.C .C /X .IC/RDH .SY/DSRDC .
ENTERING ROW:  4
STARTING AT COLUMN  1
.C .SYRD/H .H DS.SYDS.C RD/H SY.DSSY.SY.C RD.SYC .C /SY.C IC/RD.C SY.
ENTERING ROW:  4
STARTING AT COLUMN 13
.C SY/RD.SY/C .H /C .C SY.P X /C .C SY.U RD/C .
ENTERING ROW:  5
STARTING AT COLUMN  1
.C /P .C /SY.C .SYC .SY.C .SY.C .C RD/SY.C .C X .SY.C RD.C X /RDP .C .
ENTERING ROW:  5
STARTING AT COLUMN 16
.C P /IC.C /X .C /SYT .SY.
ENTERING ROW:  6
STARTING AT COLUMN  1
.C .C X .C RDX .SY.C /SY.X IC/RDC .C /RDIC.C SY.C /RDSY.P RD/C .
ENTERING ROW:  6
STARTING AT COLUMN 11
.SY.U U /SY.C U .U /C .C /P .U C /SY.C SY.C .C SY
ENTERING ROW:  7
STARTING AT COLUMN  1
.SY/C .C .C /SY.C .C .C SY.C /SY.C H .SY/RD.C RDSY.C /SY.C RD/SY.
ENTERING ROW:  7
STARTING AT COLUMN 13
.SY.SY.DSC .SY/RD.SY/RDC .C /SY.C /X .
ENTERING ROW:  8
STARTING AT COLUMN  1
.C .C .C /RDSY.SY/RDC .C /RD.C X .C .C /RDX .C .C .C /RD.P C .
ENTERING ROW:  8
STARTING AT COLUMN 13
.C /DS.C .C .C .SYC .C .X C /SY
ENTERING ROW:  9
STARTING AT COLUMN  1
.C /RD.C /SY.C .SY/C .SY/P .SY.C IC.C .C /X .SY.C .C /SY.SY/RDC .
ENTERING ROW:  9
STARTING AT COLUMN 14
.W IC/SY.C RD.SYC .SYC .C P .C .
ENTERING ROW: 10
STARTING AT COLUMN  1
.SY.SY.C .C RDH .C .C .C RD/SY.C .SY/RDC .C .C SYRD.C /SYX .C .
ENTERING ROW: 10
STARTING AT COLUMN 14
.C SY.P /C T H .C SYIC.C RD/IC.C /RDSY.C /P .SY/C .SYRDC .C .C SY.
ENTERING ROW: 11
STARTING AT COLUMN  5
.P /RDDS.SY.C C /H .C SY.SY/C .SYC .SYRDC .C /SY.SY/RDC .C RD.C /H .T /SY
ENTERING ROW: 11
STARTING AT COLUMN 17
.T .T RDP /C .
ENTERING ROW: 11
STARTING AT COLUMN 19
.C /P X
END OF FILE REACHED
LABEL:
.TYPE 3

ROW 11:
1        2          3    4        5            6    7          8        9
SY/C     SYRDC /    C / C SY/    P /RDDS      SY/ C C /H      C SY/    SY/C
10       11         12       13            14        15        16       17
SYC /    SYRDC /    C /SY    SY/RDC        C RD/     C /H      T /SY    T /
18             19
T RDP /C       C /P X
.HOME
.PTR
ROW:  1
COL:  1
.TYPE 3

ROW  1:
```

```
1              2          3          4            5          6         7          8
ICRDC /P       C /X U     SYU /      SY/RDC       SY/C       H /C      SY/RD      SYRD/
9              10  11         12          13  14            15         16  17
P P /A SY      C / SY/X       SYC /       SY/ IC/H SY       C /C       C / P C /SY
18         19
T /SY      T /

ROW  2:
1        2         3   4   5          6    7           8   9   10                11
C /SY    C /SY     C / C / C /SY      SY/ C /RD        C / C / H /RDC P          C /H
12  13         14       15          16              17   18         19
SY/ C /OF      C X /    C /SY       C RDX U /       SY/ SYC /       C RDSY/

ROW  3:
1           2           3          4   5          6          7          8           9
C /RDSY     C RDSY/     SYX /      C / C /C       SY/C       C /C       SY/RDP      SY/RD
10  11         12          13  14  15         16  17         18           19
C / C SY/      SY/C        C / C / C /RD      C / C /X       IC/RDH       SY/OSRDC
.SAVE
R;

.FORMS
SPECIFY FILENAME FILETYPE AND FILEMODE PLEASE
R;

.FORMS DEMO BIGDOTS A
DMSLIO740I EXECUTION BEGINS...


FORMS:
   0      EXIT FROM PROGRAM
   1      ONLY PIXELS WITH ONE LABEL FOR: 209 SMALL (P1) DOTS
   2      ONLY PIXELS WITH ONE LABEL FOR: BIGDOTS
   3      ALL LABELS FOR: 209 SMALL (P1) DOTS
   4      ALL LABELS FOR: BIGDOTS

ENTER YOUR CHOICE (I1)
.4
ALL LABELS FOR BIG DOTS PRINTED


FORMS:
   0      EXIT FROM PROGRAM
   1      ONLY PIXELS WITH ONE LABEL FOR: 209 SMALL (P1) DOTS
   2      ONLY PIXELS WITH ONE LABEL FOR: BIGDOTS
   3      ALL LABELS FOR: 209 SMALL (P1) DOTS
   4      ALL LABELS FOR: BIGDOTS

ENTER YOUR CHOICE (I1)
.3
ALL LABELS FOR SMALL DOTS PRINTED


FORMS:
   0      EXIT FROM PROGRAM
   1      ONLY PIXELS WITH ONE LABEL FOR: 209 SMALL (P1) DOTS
   2      ONLY PIXELS WITH ONE LABEL FOR: BIGDOTS
   3      ALL LABELS FOR: 209 SMALL (P1) DOTS
   4      ALL LABELS FOR: BIGDOTS

ENTER YOUR CHOICE (I1)
.2
BIG DOTS PRINTED


FORMS:
   0      EXIT FROM PROGRAM
   1      ONLY PIXELS WITH ONE LABEL FOR: 209 SMALL (P1) DOTS
   2      ONLY PIXELS WITH ONE LABEL FOR: BIGDOTS
   3      ALL LABELS FOR: 209 SMALL (P1) DOTS
```

```
   4      ALL LABELS FOR: BIGDOTS

ENTER YOUR CHOICE (I1)
.1
SMALL DOTS PRINTED

FORMS:
   0      EXIT FROM PROGRAM
   1      ONLY PIXELS WITH ONE LABEL FOR: 209 SMALL (P1) DOTS
   2      ONLY PIXELS WITH ONE LABEL FOR: BIGDOTS
   3      ALL LABELS FOR: 209 SMALL (P1) DOTS
   4      ALL LABELS FOR: BIGDOTS

ENTER YOUR CHOICE (I1)
.0
R;
```

SUBMIT
ROUTINE WHICH SENDS DOT FILES
TO BE ADDED TO THE DOT DATA BASE

DESCRIPTION

SUBMIT SENDS DOT FILES TO A UTILITY ACCOUNT WHICH IS USED FOR MAINTENANCE OF THE DATA BASE. THERE THE DOT FILES ARE ADDED TO THE DATA BASE UNDER STRICT SCRUTINY. THE FILE IS CHECKED TO MAKE SURE IT MEETS CERTAIN STANDARDS BEFORE IT IS ADDED TO THE DATA BASE. (FOR EXAMPLE, THERE SHOULD BE SOME DATA POINTS IN EITHER THE GRID OR THE OFFGRID. IF BOTH SECTIONS ARE EMPTY, THE FILE WILL NOT BE ADDED TO THE DATA BASE.)

DOT GROUND INVENTORY DATA BASE SYSTEM

SOFTWARE FOR QUERYING THE DATA BASE.

THE OUTPUT FOR THE APOVE INQUIRY APPEARS AS :

QUERY OF RT&E DATA BASE FOR DOT FILES

SEGMENT NUMBER 6900

FILE SEQUENCE NUMBER	CROP YEAR	LABEL TYPE	LATITUDE	LONGITUDE	EXPERIMENT NUMBER	ANALYST IDENTIFIER	DATE OF LABELING	DATE ENTERED DATA BASE
1	77	1	36 43 N	95 32 E	432	56	80145	80151
1	80	2	36 43 N	95 32 E	2	4	80146	80151
2	80	2	36 43 N	95 32 E	23	14	80146	80151
1	84	2	36 43 N	95 32 E	4	9	80123	80151

DOTQUERY

DESCRIPTION :

DOTQUERY IS USED TO QUERY THE RT&E DATA BASE FOR INFORMATION ON DOT FILES. THE USER CAN QUERY ONE OR ALL SEGMENT NUMBERS, ONE OR ALL CROP YEARS. THE USER HAS THE CHOICE OF HAVING THE OUTPUT PRINTED OR RETURNED TO THE TERMINAL. THE PROGRAM IS INITIATED BY TYPING 'DOTQUERY'.

EXAMPLE OF OPERATION :

```
SOURCE    RESPONSE
--------  ----------

USER :    DOTQUERY

PROGRAM:  AN INQUIRY INTO THE DOT LABEL FILES HAS BEEN INITIATED.
          IF YOU WANT THE OUTPUT ON YOUR TERMINAL ENTER 'TERMINAL'.
          IF YOU WANT THE OUTPUT TO BE PRINTED ENTER 'PRINTER'.
          ENTER 'STOP' TO HALT EXECUTION.

USER:     TERMINAL

PROGRAM:  INPUT THE SEGMENT NUMBER OF THE DOT FILES TO BE QUERIED.
          (INPUT A '0' FOR A QUERY OF ALL SEGMENT NUMBERS)
          (INPUT A '0' FOR A QUERY OF ALL CROP YEARS)

USER:     77
```

THE PROGRAM WILL DISPLAY THE INFORMATION FOR THE DOT LABEL FILES SATISFYING THE INQUIRY, OR A MESSAGE GIVING THE ERRORS INCURRED.

TO QUERY ALL CROP YEARS FOR SEGMENT NUMBER 6900 THE FOLLOWING EXAMPLE IS GIVEN :

```
USER:     DOTQUERY

PROGRAM:  (ASKS FOR TERMINAL, PRINTER, OR STOP)

USER:     PRINTER

PROGRAM:  INPUT THE SEGMENT NUMBER OF THE DOT FILES TO BE QUERIED.
          (INPUT A '0' FOR A QUERY OF ALL SEGMENT NUMBERS)

USER:     6900

PROGRAM:  INPUT THE CROP YEAR TO BE QUERIED.
          (INPUT A '0' FOR A QUERY OF ALL CROP YEARS)

USER:     0
```

THE OUTPUT OF THE QUERY IS PRINTED ON THE USERS PRINTER. IF ANY ERROR MESSAGES ARE GENERATED, THEY ARE LISTED ON THE TERMINAL.

FORMS
PROGRAM TO PRINT DOT FILES

DESCRIPTION: FORMS PRINTS OUT DOT FILES IN ONE OF FOUR FORMATS AS SPECIFIED BY THE USER. DEPENDING ON WHETHER THE FILE IS BIG DOT OR SMALL DOT, FOUR DIFFERENT FORMS ARE AVAILABLE. THE USER IS PROMPTED WITH HIS CHOICE OF THESE FORMS OR EXITING. IF THE USER SPECIFIES A FORM, IT IS PRINTED OUT, AND THEN THE USER IS PROMPTED AGAIN WITH THE FOUR FORMS AND THE OPTION OF EXITING. THIS CONTINUES UNTIL THE ANALYST EXERCISES HIS OPTION TO EXIT. FOLLOWING THE SAMPLE SESSION (WHICH ILLUSTRATES THE USE OF FORMS) IS A SAMPLE OF THE OUTPUT FORMS PRODUCES.

DOT GROUND INVENTORY DATA BASE SYSTEM

DOCUMENTATION OF SUBROUTINES FOR DELIVERY OF CCT DATA
TO EXECUTING FORTRAN PROGRAMS.

CALL DOTFO(SEGNUM, CROPYR, FILENO, NDOTS, INFO, UNIT, MODE, ERROR)

DOTFO SEARCHES THE RT&E DATA BASE FOR SELECTED INFORMATION CONCERNING THE DOT FILES SPECIFIED. AN ARRAY IS RETURNED WITH THIS INFO.

VARIABLE	USAGE
SEGNUM	I*4 SEGMENT NUMBER TO SEARCH FOR
CROPYR	I*4 CROP YEAR OF THE SEGMENT NUMBER REQUESTED. ONLY THE LAST TWO DIGITS OF THE YEAR ARE EXPECTED. IF ALL CROP YEARS ARE DESIRED, SET THIS VARIABLE TO '0'.
FILENO	I*4 ARGUMENT TO INDICATE WHICH DOT FILE FROM THE SEGMENT AND CROP YEAR SPECIFIED TO RETURN INFO OF. IF A '0' IS PASSED, INFO FROM ALL FILES FROM SEGNUM AND CROPYR ARE RETURNED. NOTE, SINCE IT WILL BE POSSIBLE TO HAVE MORE THAN ONE FILE FOR A GIVEN SEGNUM AND CROPYR, FILE NUMBERS ARE NECCESSARY FOR IDENTIFICATION.
NDOTS	I*4 VARIABLE WHICH RETURNS THE NUMBER OF VALID FILES FOUND.
INFO	I*4 ARRAY OF SIZE (13,64) WHICH WILL COLLECT THE FOLLOWING: INFO(1,*) - FILE SEQUENCE NUMBER INFO(2,*) - CROP YEAR INFO(3,*) - LABEL TYPE (1=SMALL 2=BIG) INFO(4,*) - DEGREES LATITUDE INFO(5,*) - MINUTES LATITUDE INFO(6,*) - 'N' OR 'S' INFO(7,*) - DEGREES LONGITUDE INFO(8,*) - MINUTES LONGITUDE INFO(9,*) - 'E' OR 'W' INFO(10,*) - EXPERIMENT NUMBER INFO(11,*) - ANALYST IDENTIFICATION INFO(12,*) - DATE OF LABELING INFO(13,*) - DATE ENTERED DATA BASE
UNIT	I*4 FORTRAN I/O UNIT NUMBER TO BE USED BY DOTFO
MODE	I*2 MODE THAT DATA BASE DISK WAS ACCESSED AS. FOR EXAMPLE, IF THE JSCDISK 19A WAS ACCESSED AS A C-DISK, THE MODE WOULD BE 'C1'.
ERROR	I*4 ERROR RETURN CODE. CALL RTEERR FOR ERROR MESSAGE.

```
CALL GETDOT(SEGNUM, CROPYR, FILENO, HEADER, GRID, OFFGRD, UNIT,
            MODE, ERROR, FORMAT)
```

THIS SUBROUTINE WILL SEARCH THE SEGMENT CATALOG FOR THE SPECIFIED DOT LABEL FILE AND FILL ARRAYS WITH LABEL INFORMATION.

VARIABLE	USAGE
SEGNUM	I*4 SEGMENT NUMBER TO SEARCH FOR
CROPYR	I*4 CROP YEAR OF DESIRED SEGMENT
FILENO	I*4 FILE SEQUENCE NUMBER
HEADER	L*1 ARRAY OF SIZE 400 TO HOLD HEADER INFO. SEE FILE DOTLABEL INFO FOR FORMAT.
GRID	ARRAY OF SIZE (8,19,22) WHICH STORES LABEL INDEXES. IN BIG DOT FILES, SMALL DOTS ARE STORED IN ODD NUMBERED ROWS AND BIG DOTS ARE STORED IN EVEN NUMBERED ROWS.
OFFGRD	ARRAY OF SIZE (20,200) WHICH STORES OFF-GRID LABELS. FOR BIG DOT FILES THE FORMAT IS: OFFGRD(1,*) - X COORDINATE OFFGRD(2,*) - Y COORDINATE OFFGRD(3-4,*) - UNUSED OFFGRD(5-12,*) - SMALL DOT LABELS OFFGRD(13-20,*) - BIG DOT LABELS FOR SMALL DOT FILES, STORAGE LOCATIONS 13-20 WILL BE ZERO FILLED.
UNIT	I*4 FORTRAN I/O UNIT NUMBER FOR GETDOT
MODE	I*2 MODE THAT DATA BASE DISK WAS ACCESSED AS
ERROR	I*4 ERROR RETURN CODE. CALL RTEERR FOR ERROR MESSAGE
FORMAT	I*4 PARAMETER INDICATING ARRAY TYPES (1=LOGICAL *1, 2=INTEGER *2, 4=INTEGER *4). THE ARRAYS GRID AND OFFGRD CAN BE ANY OF THESE TYPES AND GETDOT WILL STORE ONE LABEL INDEX PER STORAGE LOCATION.

NOTE: ANALYST DEFINED SPECIAL CROP CODES WILL HAVE POINTERS OF 246 - 255. NULL CROP CODES WILL HAVE A POINTER OF 0.

```
CALL DTLOAD(FNAME, FTYPE, FMODE, HEADER, GRID, OFFGRD, UNIT,
            ERROR, FORMAT)
```

THIS SUBROUTINE WILL GET THE DOT FILE UNDER THE FILENAME, FILETYPE, AND FILEMODE GIVEN AND FILL THE ARRAYS WITH THE DOT INFORMATION.

VARIABLE	USAGE
FNAME	R*8 FILENAME OF THE DOT FILEA
FTYPE	R*8 FILETYPE OF THE DOT FILE.
FMODE	I*2 FILEMODE OF THE DOT FILE.
HEADER	L*1 ARRAY OF SIZE 400 TO HOLD HEADER INFO. SEE FILE DOTLABEL INFO FOR FORMAT.
GRID	ARRAY OF SIZE (8,19,22) WHICH STORES LABEL INDEXES. IN BIG DOT FILES, SMALL DOTS ARE STORED IN ODD NUMBERED ROWS AND BIG DOTS ARE STORED IN EVEN NUMBERED ROWS.
OFFGRD	ARRAY OF SIZE (20,200) WHICH STORES OFF-GRID LABELS. FOR BIG DOT FILES THE FORMAT IS: OFFGRD(1,*) - X COORDINATE OFFGRD(2,*) - Y COORDINATE OFFGRD(3-4,*) - UNUSED OFFGRD(5-12,*) - SMALL DOT LABELS OFFGRD(13-20,*) - BIG DOT LABELS FOR SMALL DOT FILES, STORAGE LOCATIONS 13-20 WILL BE ZERO FILLED.
UNIT	I*4 FORTRAN I/O UNIT NUMBER FOR DTLOAD
ERROR	I*4 ERROR RETURN CODE. CALL RTEERR FOR ERROR MESSAGE
FORMAT	I*4 PARAMETER INDICATING ARRAY TYPES (1=LOGICAL *1, 2=INTEGER *2, 4=INTEGER *4). THE ARRAYS GRID AND OFFGRD CAN BE ANY OF THESE TYPES AND DTLOAD WILL STORE ONE LABEL INDEX PER STORAGE LOCATION.

NOTE: THIS SUBROUTINE WILL FAIL TO RUN IF THE FILENAME, FILETYPE, AND FILEMODE ARE NOT OF A VALID DOT FILE.

```
CALL DTUNLD(FNAME, FTYPE, FMODE, HEADER, GRID, OFFGRD, UNIT,
            ERROR, FORMAT)

THIS SUBROUTINE WILL PUT THE DOT FILE UNDER THE FILENAME, FILETYPE, AND FILEMODE
GIVEN FROM THE ARRAYS WITH THE DOT INFORMATION.

VARIABLE  USAGE

FNAME     R*8 FILENAME OF THE DOT FILEA

FTYPE     R*8 FILETYPE OF THE DOT FILE.

FMODE     I*2 FILEMODE OF THE DOT FILE.

HEADER    L*1 ARRAY OF SIZE 400 TO HOLD HEADER INFO.  SEE FILE
          DOTLABEL INFO FOR FORMAT.

GRID      ARRAY OF SIZE (8,19,22) WHICH STORES LABEL INDEXES.  IN BIG
          DOT FILES, SMALL DOTS ARE STORED IN ODD NUMBERED ROWS AND
          BIG DOTS ARE STORED IN EVEN NUMBERED ROWS.

OFFGRD    ARRAY OF SIZE (20,200) WHICH STORES OFF-GRID LABELS.  FOR BIG
          DOT FILES THE FORMAT IS:
               OFFGRD(1,*)       - X COORDINATE
               OFFGRD(2,*)       - Y COORDINATE
               OFFGRD(3-4,*)     - UNUSED
               OFFGRD(5-12,*)    - SMALL DOT LABELS
               OFFGRD(13-20,*)   - BIG DOT LABELS

          FOR SMALL DOT FILES, STORAGE LOCATIONS 13-20 WILL BE
          ZERO FILLED.

UNIT      I*4 FORTRAN I/O UNIT NUMBER FOR DTUNLD

ERROR     I*4 ERROR RETURN CODE.  CALL RTEERR FOR ERROR MESSAGE

FORMAT    I*4 PARAMETER INDICATING ARRAY TYPES (1=LOGICAL *1,
          2=INTEGER *2, 4=INTEGER *4).  THE ARRAYS GRID AND OFFGRD CAN
          BE ANY OF THESE TYPES AND DTLOAD WILL STORE ONE LABEL INDEX
          PER STORAGE LOCATION.

NOTE: THE DATA HEADER AND OFFGRD MUST BE IN THE LOW ORDER BYTE OF THE FILED.
```

CALL CRPNAM(CCODE, CNAME, LIST, CCOUNT, UNIT, MODE)

CRPNAM WILL READ IN THE CROP NAMES AND CODES LIST STORED ON THE DATA BASE DISK. THE LABEL INDEXES PASSED IN GETDOT POINT TO THESE ENTRIES.

VARIABLE	USAGE
CCODE	I*2 ARRAY OF SIZE 245 TO STORE ALL TWO LETTER CROP CODES
CNAME	2R*8 ARRAY OF SIZE 245 TO STORE ALL 16 CHARACTER CROP NAMES
LIST	LOGICAL*1 THE CROP NAMES LIST TO USE (1, 2 ECT).
CCOUNT	I*4 NUMBER OF CROP NAMES IN USE
UNIT	I*4 FORTRAN I/O UNIT NUMBER TO BE USED BY CRPNAM
MODE	I*2 MODE THAT DATA BASE DISK WAS ACCESSED AS

NOTE: ANALYST DEFINED SPECIAL CROP CODES WILL HAVE POINTERS OF 246 - 255. NULL CROP CODES WILL HAVE A POINTER OF 0.

Appendix II

Purdue/LARS CMS Short Course Schedule
December, 1979

			Level
MONDAY December 10	8:30	Introduction to VM370	(Introductory)
	9:00	HANDS-ON	
	10:00	CMS I	(Introductory)
	11:00	HANDS-ON	
	1:30	Virtual Machine Concepts	(Intermediate)
	2:30	CMS II	(Intermediate)
	3:30	HANDS-ON	
TUESDAY December 11	8:30	EDIT I	(Intermediate
	9:30	EXEC I	(Intermediate)
	10:30	HANDS-ON	
	1:30	Programming I	(Intermediate)
	2:30	Batch I	(Intermediate)
	3:30	HANDS-ON	
WEDNESDAY December 12	8:30	Programming II	(Experienced)
	9:30	Stat Packages	(Intermediate)
	10:30	RT&E Data Bases	(Intermediate)
	10:30	HANDS-ON	
	1:30	CMS III	(Experienced)
	2:30	CP Commands	(Experienced)
	3:30	Other RT&E Software (IMSL, CSMP, LARSYSP1,...)	(Intermediate)
	3:30	HANDS-ON	
THURSDAY December 13	8:30	EXEC II	(Experienced)
	9:30	Batch II	(Experienced)
	10:30	EDIT	
	10:30	HANDS-ON	
	1:30	Graphics Programming	(Intermediate)
	2:30	LARSPEC	(Intermediate)
	4:00	HANDS-ON	
FRIDAY December 14	9:00	Script	(Intermediate)
	9:00 and 11:00	LARSPEC Demonstration	
	9:00 - 2:00 pm	General Consulting in the Remote Terminal Area	(Everyone)

Appendix III

THE FORMAT OF THE MAIL COMMAND IS:

```
+-----------+-----------------------------------+
            -                          -
              -       -       -    -
    MAIL       USERID          FILEID
               LISTNAME        -    -
               SYNONYM
               *
              -       -
            -                          -
+-----------+-----------------------------------+
    LARSYS, LARSYSDV, AND LARSPEC USERS SHOULD
    TYPE "EXEC MAIL ..." TO USE THE MAIL
    SYSTEM.
+-----------------------------------------------+
```

USERID IS THE VIRTUAL MACHINE ID THAT YOU WANT TO SEND MAIL TO.

LISTNAME IS THE NAME OF A MAILING LIST CONTAINING ONE OR MORE VIRTUAL MACHINE ID'S THAT ARE TO RECEIVE THE SAME PIECE OF MAIL.

SYNONYM IS A SYNONYM FOR A VIRTUAL MACHINE ID THAT YOU WANT TO SEND MAIL TO.

* INDICATES THAT YOU WISH TO SEND MAIL TO YOURSELF.

FILEID IS THE FILE NAME, FILE MODE, AND (OPTIONALLY) FILE TYPE OF A FILE CONTAINING THE MAIL YOU WISH TO SEND. IT MUST BE A FIXED FORMAT, 80-BYTE RECORD FILE. IF THIS PARAMETER IS OMITTED, THE CMS EDITOR IS CALLED TO ENABLE YOU TO CREATE A MAIL FILE.

MAIL EXEC IS INTENDED TO FACILITATE SENDING SHORT, MESSAGE-TYPE FILES TO OTHER SYSTEM USERS, PROVIDE AUTOMATIC NOTIFICATION THAT MAIL IS BEING HELD FOR A USER, AND PROVIDE QUICK, EASY DELIVERY OF THESE FILES. MAIL WILL BE AVAILABLE FOR PICKUP WITHIN 3 MINUTES AFTER IT IS SENT.

MAIL MAY BE CHECKED OR RECEIVED BY TYPING "MAIL". TO RECEIVE, YOUR "A" DISK MUST BE ACCESSED R/W. YOU HAVE THE OPTION OF SAVING EACH INDIVIDUAL PIECE OF MAIL YOU RECEIVE. IF ANY MAIL IS SAVED, MAIL WILL TRANSFER YOUR MAIL TO YOU AND TELL YOU THE FILE NAME. MAIL NOT CLAIMED AFTER ONE MONTH WILL BE DESTROYED. YOU MAY CHECK YOUR MAIL AUTOMATICALLY BY INSERTING "EXEC MAIL" IN YOUR PROFILE EXEC.

MAIL ALSO PERMITS SYNONYMS FOR USERID'S, AND DISTRIBUTION LISTS THAT ALLOW SENDING MAIL TO MORE THAN ONE OTHER USER AT A TIME.

```
+--------+------------------------------------+

 (1)           SMITH JSC9999
 (2)     LIST ROUTE1
 (3)           JOHN CONVERT9
 (4)           REFORM99
 (5)     ENDLIST
 (6)           AL LCI9999
 (7)     LIST ROUTE2
 (8)           MIKE FM9999
 (9)           GEORGE TESTID9
 (10)    LIST ROUTE3
 (11)          JSC999A
 (12)          JSC999B
 (13)          PURDUE ROSEBOWL

+--------+------------------------------------+
```

A SAMPLE DISTRIBUTION LIST IS SHOWN ABOVE, IN THE RIGHT SIDE OF THE BOX. LINE (1) IS AN EXAMPLE OF A SYNONYM: "MAIL SMITH" WOULD ALLOW YOU TO EDIT A NEW FILE, WHICH WOULD BE SENT TO JSC9999. SYNONYM ENTRIES MAY BE PLACED AT ANY POINT IN THE FILE. LINES (2), (7), AND (10) SHOW DISTRIBUTION LISTS. EACH LIST IS TERMINATED BY EITHER "ENDLIST", "LIST", OR END OF FILE. LISTS MAY INCLUDE SYNONYM ENTRIES; OR, IF YOU HAVE NO SYNONYM FOR AN ID, JUST THE ID NAME (11). TO SEND MAIL TO A LIST, USE THE LIST NAME, I.E., "MAIL ROUTE1" (FOR LIST IN LINE (2)). IF THE ID YOU WISH TO SEND MAIL TO IS NOT INCLUDED IN YOUR DISTRIBUTION LIST, YOU MAY PUT THE ID NAME IN THE COMMAND LINE, JUST AS IF YOU HAD NO DISTRIBUTION LIST FILE (I.E., "MAIL FM0000").

DISTRIBUTION LISTS ARE ENTIRELY FREE FORMAT. IF YOU CREATE ONE, MAIL WILL EXPECT IT TO BE CALLED "MAILING LIST A1".

DISTRIBUTION LIST: VOLUME IV

NASA Johnson Space Center

SA:	W. Rice
SK:	R. Erb
SG:	R. MacDonald
SH:	J. Erickson
SG2:	D. Hay
	R. Musgrove
	W. Weimer
	J. Gilbert
SG3:	F. Hall
	C. Hallum
	R. Heydorn
	D. Pitts
	M. Steib
SH2:	M. Trichel
SH3:	J. Dragg
	R. Bizzell
	R. Eason
	K. Baker
JM6:	Technical Library (4)
AP:	Public Affairs Office

NASA Headquarters

ER-2: P. Thome

NASA Scientific and Technical Information Facility

R. Horvath:	ERIM
W. Malila:	ERIM
R. Cicone:	ERIM
D. Rice:	ERIM
V. Salomonson:	NASA-GSFC
W. Alford:	NASA-GSFC
T. Minter:	LEMSCO
B. Carroll:	LEMSCO

Additional volumes available upon request from Purdue/LARS.

Vol. I. Field Research on the Spectral Properties of Crops and Soils
Vol. II. Research in the Application of Spectral Data to Crop Identification and Assessment
Vol. III. Data Processing Research and Techniques Development
Vol. IV. Computer Processing Support