

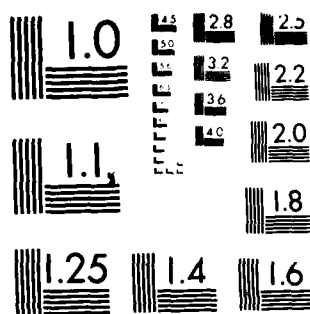
TACTICAL ELECTRONIC WARFARE ENVIRONMENT SIMULATOR DATA
ANALYSIS STUDY REPORT(U) AMHERST SYSTEMS INC BUFFALO NY
E G EBERL JAN 78 N00173-77-C-0105

1/3

F/G 17/4

NL

END
DATE
FILMED
7-84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

AMHERST SYSTEMS INC.

TACTICAL ELECTRONIC WARFARE
ENVIRONMENT SIMULATOR

DATA ANALYSIS STUDY REPORT

JANUARY 1978

Written By:

Dr. Edward G. Eberl
Amherst Systems, Inc.

Written For:

Naval Research Laboratory
455 Overlook Avenue SW
Washington, D. C. 20375

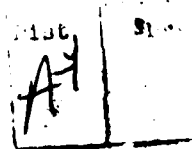
Contract Number N00173-77-C-0105

DTIC
ELECTE
JUN 5 1984
S A

AD-A141 892

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
	INTRODUCTION	1
1	DATA EXTRACTION FILE LISTINGS.	2
2	SCENARIO MAINTENANCE PROGRAM	3
2.1	General Operation.	3
2.2	Operating Instructions	4
2.2.1	Initiating the Program	4
2.2.2	Program Commands	5
2.2.2.1	Code 0 - List Next Event	6
2.2.2.2	Code 1 - Delete Current Event.	7
2.2.2.3	Code 2 - Insert New Event.	7
2.2.2.4	Code 3 - Search For Emitter Reference.	11
2.2.2.5	Code 4 - Search For Platform Reference	12
2.2.2.6	Code 5 - Search For Time Reference	12
2.2.2.7	Code 6 - Rewind the Scenario	13
2.2.2.8	Code 7 - Save Emitter Characteristics.	13
2.2.2.9	Code 8 - Recall Emitter Characteristics.	14
2.2.2.10	Code 9 - Edit The Current Event.	14
2.2.2.11	Code 10 - Abort, Delete All Scratch Files.	16
2.2.2.12	Code 11 - Abort, Delete Output File.	16
2.2.2.13	Code 98 - Exit	16
3	SCENARIO LISTINGS.	19
4	PERIPHERAL INTERCHANGE PROGRAM	20
4.1	File Specifiers.	20
4.2	Directory Listing.	21
4.3	Delete Files	21
4.4	Copy Command	22
4.5	Free Space	23



INTRODUCTION

This study report was produced under the direction of Mr. R. Larry Robinson, Code 5724.06 of the Naval Research Laboratory under Contract Number N00173-77-C-0105.

The report documents the operation of the following programs which will support the effective use of the Tactical Electronic Warfare Environment Simulator.

- a. Data Extraction File Listing Program.
- b. Scenario File Maintenance Program.
- c. Scenario File Listing Program.
- d. Peripheral Interchange Program.

SECTION 1
DATA EXTRACTION FILE LISTINGS

A stand alone program is provided to list the contents of data extraction files either on the operator's console or the line printer. The procedure to initiate the program depends on whether the operating system is up and running or not. If the system is not up, the system must first be bootstrapped. When the processor and terminal are first powered up, a \$ prompt will appear on the screen. After placing the disk in the run position, enter a DK carriage return ([CR]) and the system will bootstrap. When the executive is loaded, it requests the current time and date. After this information has been entered, the operating system is up and running, and the program can be initiated by entering RUN LISDX [CR]. The program will request the name of the file to be listed, a destination code of 1 for the console or 3 for the printer, and the time limits for the list. All scenario events and operator commands with times in the range specified will be listed. The data is listed with appropriate identification fields, so most of the output is self-explanatory.

Some output fields require care in their interpretation. For the emitter events, the scan type is listed as an actual value. Scan sector data is listed in BAMS, and not corrected for the scan multiplier effect. Also, the ~~scan sector~~ ^{RF Channel Pilot Code & Pulse}, jitter, ^{as, chronms, offset} ~~micro-jitter~~, chirp and frequency agility are listed as the actual codes stored in the scenario. Similarly, the scan rate index is output as the number of pulses per increment or number of increments per pulse and is not converted to a scan period. It will be helpful to reference a listing of the input scenario file to find the data in a more convenient format.

When the list is complete, a new destination code is requested, followed by a new set of time limits. This allows the operator to step through selected portions of the ^{same} DX file. If a destination code of 0 is entered, the program will terminate. If the new lower limit is the same as the old upper limit, events at that time are not listed, since this would be redundant.

SECTION 2

SCENARIO MAINTENANCE PROGRAM

The scenario maintenance program is a multipurpose program capable of modifying existing scenario files and creating new scenario files in the format required by the TEWES Realtime Software. For further description of the scenario file format and content, consult the Realtime Software documentation. The program is designed to run under the Digital Equipment Corporation's RSX-11M V03 operating system which resides in the Control Subsystem. It was written using the Fortran-IV and Macro-11 languages supplied with RSX-11M.

2.1 GENERAL OPERATION

The basic function of the scenario maintenance program is to input events from an existing scenario file on disk or from the operator's terminal and output events to a new scenario file on disk. When the program is initiated, the first event from the existing scenario is displayed on the operator's console, and is referred to as the current event. Commands are provided to modify or delete the current event, insert new events, advance the scenario, return to the beginning of the scenario, store and recall emitter characteristics, and terminate the program. As the scenario is advanced, events are input from the existing file and output to the new file sequentially. The existing scenario remains untouched and is always preserved, and a new scenario file is created containing the old scenario events plus any modifications or additions made. During an editing session, events need not be kept in chronological order. When the program is terminated normally, the final version of the scenario is automatically sorted into the proper order. This requires creating a scratch copy of the scenario file. Users should be aware of this disk space usage and not attempt to edit a scenario file unless sufficient disk space is available. The work space needed is approximately twice the number of blocks expected for the final version of the scenario. When editing an old scenario, this means work space equal to twice the original scenario file size is needed. When creating a new scenario, the size required for the final version can be estimated using Table I in Appendix A and knowing the expected numbers of the various events to be placed in the scenario.

Whenever information is required by the program, the operator is prompted with a description of the data to be entered, and the program then waits for the data to be entered on the operator's terminal. When entering alphanumeric data, leading and trailing spaces should be avoided, as they are interpreted as zeroes. The program is also equipped with a backup or cancel feature, which allows for simple correction of data entry errors. When data is requested, answering with a -12 will cancel the command currently being processed. A response of -11 causes the program to back up one question and repeat the previous input request. Note that in the case of commands requiring a single input, both responses have the same effect. At some critical points in the program, these features must be modified due to the nature of the data being input. Such instances are noted throughout the rest of the documentation as they occur. In most cases, depressing the return key without typing any characters is translated to a zero input, except for some special cases, notably when editing a particular event, or entering a time field, when an explicit zero must be entered. Input prompts also contain the applicable units associated with the response. The program includes legal limits for every request, and responses outside these limits produce an error message and the request is repeated. Similarly, if non-numeric data is entered in response to a request for a numeric value, the data is ignored and the request is repeated.

2.2 OPERATING INSTRUCTIONS

2.2.1 Initiating The Program

The procedure for initiating the program includes running the program and entering the required file names. If the RSX-11M operating system is not up and running the system must first be bootstrapped. The first step is to turn on the power for the console terminal and then for the main processor. As the terminal warms up, there will be a \$ prompt on the screen. Switch the disk drive to the run position and enter a DK [CR] on the terminal. The operating system will be loaded and will request the current time of day and the date. After this data has been entered, the scenario creation program is initiated by typing RUN EDSCEN [CR]. If the operating system is already up and running, all that is required is the RUN EDSCEN [CR] command.

When the program is initiated, it requests the name of the existing scenario file to be used for input. If a new file is to be created, a blank followed by a carriage return is input. Next, the name of the output scenario file name is requested. Since this need not be the same as the input file name it is possible to create different versions of a base scenario and store them under different names. If an input file was specified, and the program was unable to open that file, a blank line followed by a carriage return entered as the output file name will terminate the program, allowing the operator to determine why the input file was not opened, and try again. If both input and output files are successfully opened, the program displays the first input event and then waits for one of the fourteen basic commands to be entered. If the program is unable to open the specified output file, an immediate exit is taken to allow the operator to rectify the problem. If the output file is successfully opened and the input file was either not specified or not opened successfully, the program assumes that a new scenario is to be created, and generates an Insert New Event command internally and requests that a scenario event be entered.

2.2.2 Program Commands

The scenario maintenance program has fourteen commands available to the user. Commands are given by entering a numeric code associated with the desired command. Command codes are assigned as follows:

- | | |
|----|---------------------------------|
| 0 | List Next Event |
| 1 | Delete Current Event |
| 2 | Insert a New Event |
| 3 | Search for Emitter Reference |
| 4 | Search for Platform Reference |
| 5 | Search for Time Reference |
| 6 | Rewind the Scenario |
| 7 | Save Emitter Characteristics |
| 8 | Recall Emitter Characteristics |
| 9 | Edit the Current Event |
| 10 | Abort, Delete All Scratch Files |
| 11 | Abort, Delete Output File |
| 98 | Exit |
| 99 | List This Menu |

Note that since command code zero corresponds to listing the next event, it is possible to step through portions of a scenario one event at a time by continually inputting blank lines followed by carriage returns.

A special condition occurs when either the end of the input scenario file is encountered or a new scenario is being created. In both of these situations, it is not valid to attempt to read an event from an input scenario file. In fact the current event is invalid, since it is the flag used to signify the end of file. Therefore, some of the fourteen commands are illegal at this point and attempts to execute these commands will result in error messages. The commands will be ignored whenever they are not legal. When the end of file has been encountered, the operator is notified of the condition and reminded of the abbreviated command list. At this point, the only valid commands are Insert New Event; Rewind the Scenario; Recall Emitter Characteristics; Abort, Delete All Scratch Files; Abort, Delete Output File; and Exit. Individual commands are described in detail below.

2.2.2.1 Code 0 - List Next Event

As noted previously, this is the default command which is executed when either a 0 or no command code is entered before depressing the return key. The current event is output to the new scenario file and the next event is read from the old scenario file. The event just read becomes the new current event and is displayed on the operator's terminal. In general, all data are converted from integer codes back to the format and units used when the values are input. Units used for all output fields are found in Table II. Users should realize that the internal format for data storage is quite different from the formats used for data entry and output. All values are stored in integer format, and in many cases are actually stored as codes representing table values. For this reason, many of the values output will be slightly different from those input due to round off errors and table approximations. Users should bear this in mind when evaluating event listings.

In some cases, data cannot be converted back to the original input format. This occurs, for example, when listing an enter new emitter event in which the use of special RF channels has been specified. In determining the frequency agile code

stored in the scenario, the frequency of the emitter must be known. If special channels are in use, this information is not available when listing the event, and the frequency agile code cannot be converted to a frequency agility deviation in megahertz. Instead, the actual frequency agile code is listed. Similarly, when listing the change emitter parameter event, the pulse width, which is used in converting the chirp rate to a chirp limit is not known. Therefore, the actual chirp rate is listed. In special cases, the values output are identified as to their exact content, so the listings are always clear, but users should take care in reading the identifiers associated with the various fields.

2.2.2.2 Code 1 - Delete Current Event

Command code 1 is used to remove an event from a scenario. It is similar to the List Next Event command, except that the current event is not output before reading the next event from the input scenario. Instead, the current event is overwritten in memory by the next event read in from the old scenario. This then becomes the current event and is displayed on the operator's console. Note that the Delete Current Event command is invalid when inputting a new scenario or after an end of file has been encountered when editing an old scenario.

2.2.2.3 Code 2 - Insert New Event

Command code 2 is used to insert new events into either an old or new scenario file. The current event is first output to the new scenario file. The user is then prompted to enter an event type code based on the following.

- 1 Enter New Platform
- 2 Delete Platform
- 3 Platform Velocity Change
- 4 Platform Reposition
- 5 Enter New Emitter
- 6 Delete Emitter
- 7 Emitter Off
- 8 Emitter On
- 9 Change Emitter Parameter

Depending on the event type selected, the program then requests appropriate data to specify all necessary event fields. Units are listed in Table II in Appendix A.

The Insert command is designed to enter a series of events. After each event has been entered, it is immediately output to the new scenario file. If the event is an Enter New Emitter event, the operator has the option of saving the emitter characteristics in a disk file as discussed below. The operator is then prompted to enter the next event code, and in turn all the data required by the specified event. To signal the end of the new event list, a cancel command code of -12 is entered. The next event is then read in from the old scenario file and becomes the current event, unless this is a new scenario or an endfile has been encountered. In this case, there is no current event and the abbreviated command list is in effect. Note that if an Enter New Event command is specified, and a cancel code is input before any event has been input, it has the same effect as a List Next Event command.

For the Enter New Platform event, the program requests one field at a time including the platform number, position update method, east-west position, north-south position, altitude and heading. If the program update method was specified, the speed and climb rate are requested. Otherwise, all velocity fields are zeroed, and finally the desired time for the event must be input.

For the Delete Platform Event, all that is required are the platform number and the desired time. To specify a Velocity Change event, the platform number, heading, speed and climb rate are each requested in turn, and finally the desired time is input. Similarly, for the Platform Reposition event the operator is prompted to enter in turn the platform number, east-west position, north-south position, altitude, heading and event time.

The Enter New Emitter event requires the largest number of responses from the operator and is the most complex event. First the emitter and platform numbers are specified. The next field to be entered is a scan type code, which specifies the type of antenna scan to be used, where 1 signifies a standard scan type, 2 signifies a conical scan type and 3 signifies an omnidirectional antenna. If an omnidirectional antenna is specified, beamwidth information is not requested.

For either standard or conical emitters, the user is prompted to enter the antenna beamwidth, and a code of 0 or 1 to specify either a unidirectional or bidirectional scan. Since the scan type and beamwidth are combined to determine the actual code placed in the scenario, both fields must be reentered if the -11 backup feature is used to back up once the beamwidth has been entered. To complete the scan data entry, the lower scan sector boundary and the scan sector width are next requested. Then the radiated power, pulse width and pulse jitter values are requested. A priority flag, with 0 signifying low priority and 1 signifying high priority is requested followed by the asynchronous offset and the number of pulse repetition intervals (PRI's). PRI's are then entered one at a time until the specified number have been input. Utilizing the backup feature while entering PRI's causes the program to move back to the point where the number of PRI's is entered and each PRI must be reentered.

The next field requested is the emitter frequency. Although this field is not stored for special emitters using dedicated special channels, it must always be entered because the frequency agile code and chirp code conversions require a knowledge of the frequency band in use. Next the frequency agility deviation is requested, followed by a code to specify if any special RF channels are to be used, where a code of 1 signifies a special channel is to be used and 0 specifies no special channel is desired. If no special channel is requested, the next value input is the chirp limit.

If a special channel was specified, the RF selection code and the switch position are now requested. The RF selection code is determined from Table III in Appendix A. Codes 1 through 10 are included for reference in determining selection codes when special time-shared channels are used in combination with standard VCO sources. To specify channel 1 time-shared combined with a standard VCO source, add 16 to the VCO number corresponding to the frequency band desired. To specify special channel 2 time-shared combined, add 32 to the VCO number corresponding to the desired band. For both channels 1 and 2 combined, add 48 to the base code number. Note that the special channel dedicated mode is allowed only for emitters having a single PRI. Attempts to specify any other configuration will be rejected, and input data will be requested again. The switch position is entered, and the RF distribution switch position is determined from the VCO number associated with the frequency entered above.

When using timeshared channels combined with standard VCO sources the base VCO number should correspond to the frequency previously entered. Similarly, when using special channels alone, the frequency entered should correspond to the frequency associated with the special source selected by the channel and switch position specified.

If a scan sector width of zero has been specified, the enter new emitter event is now complete. When the scan sector width is not zero, the scan period must now be entered. The scan return blank flag, with 1 enabling blanking and 0 disabling blanking is input to complete the event. Finally, the event time is requested, the event is output and the next event requested.

The Emitter On, Emitter Off, and Delete Emitter events are straight forward, each requiring only the emitter number and event time to be input.

Data entry for the Change Emitter event is identical to the Enter New Emitter event, except that the emitter number is entered instead of the data is required. After the emitter number is entered, the scan period is requested which determines which scan parameters are entered. The scan return blank flag specifies that scan return blanking is enabled or disabled. The scan period, radiated power parameters, frequency related parameters, micro jitter, and priority are entered. The event time, priority, and PRF are entered.

When scan parameters are entered, the scan period is requested. If the scan type code, beamwidth, scan rate, and scan sector width are all zero, the event time is requested. If the scan sector width is not zero, the average PRI and the scan rate are requested. The scan rate is not a part of this event, but is a parameter that is used to calculate data from the scan period. For a single emitter, the scan rate is the actual PRI. For multiple PRI emitters, the scan rate is the average of the actual PRI's in use. The scan return blank flag, 1 enables blanking, 0 disables blanking.

For radiated power updates, three fields must be input. The first is the radiated power, entered just as for an Enter New Emitter event. The second is a threshold level below which the digital subsystem will not attempt to output a pulse. This field is not present for the Enter New Emitter event since values are provided by a set of default levels. The Change Emitter Parameter event allows overrides of these default levels. Event time now completes the event.

Group three includes all frequency related data. Frequency, frequency agility deviation and the special channel flag are input as with the Enter New Emitter event. If special sources are to be used, the RF selection code and switch select code are input. If no special sources are specified then the pulse width and chirp limit are input. Although pulse width is not stored as part of this event, it must be known in order to convert the chirp limit into a chirp rate. The event time completes this event.

Pulse width and asynchronous offset are specified by group code four. Both these values are entered just as for the Enter New Emitter event, along with the desired event time. Group five includes pulse jitter, the emitter priority flag, the number of PRI's and the individual PRI's. When this event is utilized, each of the PRI's must be entered. It cannot be used to change only one PRI out of a group without reentering the rest of the PRI's in the group. Similar to the Enter New Emitter Event, if the backup feature is used while entering a series of PRI's, the entire string must be reentered. The event time completes this event.

2.2.2.4 Code 3 - Search For Emitter Reference

Command code three is used to search the input scenario file for a reference to a specific emitter. The current event is first output to the new scenario file, and the emitter number to be searched for is entered. If a cancel command is given in response to the emitter number request, the next event is read in from the old scenario file and displayed as the new current event. This has the same effect as the List Next Event command. When a valid emitter number has been entered, successive events are read in from the old scenario file, and the event type is determined. If it is not an emitter type event, or does not reference the specified emitter, the event is output to the new scenario and the next event is tried. When a match is found, the event becomes the current event and is

displayed to the operator. A new command can now be entered. If the end of the old scenario is reached and no match is found: a warning message is output, the command is aborted, and the abbreviated command list is in effect. Note that since the editing program does not store information relating emitters to their associated platforms, a Search for an Emitter will not detect any events which reference the platform to which it is linked. Also the Search for Emitter Reference is invalid after an end of file or when inputting a new scenario.

2.2.2.5 Code 4 - Search For Platform Reference

Command code four is similar to command code three, except that a specified platform is searched for. Events are sequentially read from the old scenario and output to the new scenario until an event referencing the specified platform is encountered. If the command is cancelled, it has the same effect as a List Next Event command. If the end of the old scenario file is reached and no match is found, a warning message is output, the command is aborted, and the abbreviated command list is in effect. The Search for Platform Reference is invalid after an end of file or when inputting a new scenario.

2.2.2.6 Code 5 - Search For Time Reference

Command code five is used to advance the scenario to a specific time. The current event is first written to the new scenario file. The desired time is requested, and events are sequentially read from the old scenario and output to the new scenario until a time greater than or equal to the desired time is encountered. The associated event is displayed to the operator and becomes the current event. Note that at this point, the scenario is not necessarily in chronological order, so the search for a given time does not imply that all events in the remainder of the old scenario have a time greater than the current time, nor that all events already read from the old scenario have a lesser time. If the end of the old scenario is encountered before the desired time is found, a warning is issued, the command is aborted, and the abbreviated command list is in effect. The search for time reference is invalid after an end of file has been encountered or when inputting a new scenario.

2.2.2.7 Code 6 - Rewind The Scenario

Command code six is used to rewind the scenario file to the beginning. All of the previous commands have the effect of moving forward through the scenario file. There is no way to step backwards a given amount through the scenario. This command moves the current event all the way back to the beginning of the scenario. Both the current input and output files are closed. The file just written now becomes the input scenario file, and a new version number of the same file name is created as the output file. The first event from the input scenario file is read and displayed, becoming the current event just as when the program was initiated. The program will now accept any of the fourteen available command codes. Before processing any further commands, however, the program checks to see if the file just read is the original file, or an intermediate copy. If it is an intermediate copy, it is deleted after it is closed. This minimizes the amount of disk space used, and is an important feature when a scenario is rewound several times during the course of one editing session. The intermediate files have no use, since they have been superseded by later versions. The original scenario is always preserved, however, allowing the operator at least one backup copy of the file.

2.2.2.8 Code 7 - Save Emitter Characteristics

Command code seven allows the operator to store all of the data needed to specify an Enter New Emitter event in a permanent file on disk. All of the parameters contained in the current enter new emitter event are saved, and the current event is not affected. The command is invalid unless the current event is an Enter New Emitter event. The operator is prompted to enter the file name under which the emitter characteristics are to be stored. If a file with the same name already exists, the operator has the option of either superseding the old file, or aborting the command. If the old file is superseded, the data previously stored under the given file name are lost, being replaced by the new data. There is no fixed limit on the number of sets of emitter characteristics which can be stored simultaneously. Each set is stored in a separate file and referenced by a distinct file name. The only limit on how many can be stored is the amount of disk space available, with each set of emitter characteristics requiring two blocks of disk space. Note that this command cannot be used after an end of file has been encountered unless the operator is inserting new events.

2.2.2.9 Code 8 - Recall Emitter Characteristics

Command code eight is used to retrieve data stored on disk by the Store Emitter Characteristics command. Its operation is similar to the Enter New Event command, except that most of the necessary data can be obtained from a permanent disk file. The current event is first output to the new scenario file. The operator is then prompted to enter the name of the file containing the emitter characteristics to be recalled. If the command is now cancelled, it will have the same effect as a List Next Event command. When a valid file name has been entered an attempt is made to open the specified file and read the necessary data. If the file is not found or the data cannot be read, a warning is issued and the operator is prompted to reenter a file name, or cancel the command using the -12 cancel code. When emitter data have been successfully read from the permanent storage, the operator is then prompted to enter the platform number and emitter number to be associated with the new emitter, and the time desired for the event. The command is now complete unless an end of file has been encountered, or the operator is creating a new scenario. In these cases, the new event is immediately output to the new scenario file and the abbreviated command list is in effect. When the data is recalled from the disk, it is read non-destructively. The same file can be recalled as many times as needed, as long as different emitter numbers are used each time. Even after the program is terminated, the data is retained on disk in permanent storage, so it can be recalled later into a different scenario.

2.2.2.10 Code 9 - Edit The Current Event

Using command code nine, it is possible to modify single fields associated with the current event. It operates like a combination of the List Event command and Enter New Event command, except that all operations are performed on the data pertaining to the current event. Each field associated with the event is displayed on the operator's console. After a given field is listed, the operator is prompted to reenter the data. If a numeric value is entered, the new value replaces the value just listed and the event is updated. If a blank line followed by a carriage return is input, the field is left unchanged. In some cases, two fields are interrelated. The chirp code stored in the scenario, is a combination of the chirp limit and the pulse width. Changes in the pulse width require

modifications to the chirp code stored in the scenario to maintain the same chirp limit. The program automatically adjusts all related fields when any given field is updated, and the user need not be concerned with the internal interactions of the various fields.

Some special conditions occur when editing the Enter new Emitter and Change Emitter Parameter events. Because the frequency, frequency agility, chirp limit and special channel fields are so closely interrelated, all of the frequency parameters must be entered together. If the frequency request is answered with a blank line followed by a carriage return, no other frequency data is requested. If the frequency is reentered then all other frequency data must be reentered, whether it is to be changed or not. To change a value to zero, an explicit zero must be entered. Whenever data must be entered while editing an event, the program ignores blank lines followed by carriage returns, and reissues the input request until a valid response is given. When the number of PRI's is changed, the entire string of PRI's must be reentered. It is not possible to decrease the number of PRI's for example, and simply delete the last PRI's in the list. A new PRI value must be entered for each PRI specified. If the number of PRI's is left unchanged, the operator has the option of modifying any or all of the individual PRI's independently. If the scan sector width or the PRI of an emitter is changed drastically, the scan rate period may become invalid. Table II shows that the legal limits for the scan period are calculated dynamically, based upon the sector width and the average PRI. If these values are changed it may be necessary to enter a new scan period based on the new limits. In such cases, the program does not list the old scan period since it is invalid. Instead, a prompt is issued requesting the scan period, and a value between the new limits must be input. As a special case, the -12 cancel command feature is suspended when editing an Enter New Emitter event. This is done to prevent the user from changing some fields and then aborting the command before all related fields are updated. Specifically, this protects against the case where the scan sector width and PRI data are modified, possibly invalidating the current scan period. If the command could be aborted in midstream, these types of interactions would go undetected.

After the editing is complete, the event is displayed again using all of the latest information. This enables the operator to review the changes made, and reedit the event if the results are still not satisfactory.

2.2.2.11 Code 10 - Abort, Delete All Scratch Files

Command code ten is an abort command, and should be used with caution. It is designed mainly for situations where the operator wishes to exit the program when no modifications have been made. When the command is issued, all scratch files created during the current editing session are removed from the disk. The original scenario is preserved in its state before the editing session was started unless a new scenario was being input. In this case, nothing will be left on disk. All changes made or events input are lost. After all scratch files have been deleted, the program is terminated.

2.2.2.12 Code 11 - Abort, Delete Output File

Command code eleven is also an abort command, similar to command code ten. It is designed to provide an orderly exit procedure when the program is unable to output successfully to the new scenario file. The most common cause of this is a lack of sufficient disk space. If care is taken in estimating disk space requirements as described above, this condition can generally be avoided. When the disk space is exhausted, an appropriate warning message is output, instructing the operator that the disk cannot be accessed and the program should be terminated. In this case, code eleven should always be utilized. Unlike command code ten, the file currently being read is preserved even if it is an intermediate scratch file. Only the current output file is deleted, which is necessary since it is known to be corrupted. By preserving the scenario currently being read, most of the changes and additions made can be saved. All that is lost are the changes made since the last time the file was rewound using command code six. Caution is advised, however, because the file left on the disk is a scratch file and the events may or may not be in chronological order. As soon as possible, sufficient disk space should be made available and the scratch scenario should be edited and the program exited normally to allow the scenario to be sorted and verified as discussed below. This command is designed for emergency use only.

2.2.2.13 Code 98 - Exit

Command code ninety-eight is the normal command used to terminate the program. This command should always be used in preference to the Abort Command codes ten and eleven. When this command is issued, the latest version of the scenario file

is sorted and checked for errors. All scratch and intermediate files are deleted, so that all that remain are the original version and final version of the scenario file.

As previously noted, it is not necessary to keep scenario events in any specific order when creating and editing scenarios. Events can be inserted at any point in the file, regardless of the time specified for the new event. This is made possible by the sort routine which is automatically executed whenever the normal program exit is taken. The entire scenario is read and sorted into the proper chronological order. In addition, all events with equal time fields are sorted in order of increasing event number, so that at a given time all Enter New Platform events will be processed first, then the rest of the platform type events, then the Enter New Emitter events, and finally the rest of the emitter events. When two or more events of the same type are scheduled to occur at the same time, they will be processed in order of increasing platform or emitter number. The sort process requires a considerable amount of scratch space. If sufficient disk space is not available, or the sorted file cannot be properly written for any reason, a warning is issued and the program terminated. If this happens, the operator should make more space on the disk available, or rectify any other disk problem and edit the scenario as soon as possible to create a sorted version. When the sort failure warning message is issued, the final copy of the scenario may be out of order and not in an acceptable state for the real-time software. The amount of time for the sort can be minimized by keeping the scenario as well ordered as possible.

When the scenario has been successfully sorted a verification routine is executed. The scenario is read and a table is created showing all platforms which are in use, all active emitters, and linking all emitters to the appropriate platforms. As events are processed they are first checked against the current table for validity. Enter New Platform events are checked for platform availability, and all other platform events are tested to be sure that the platform is already there. Emitter events are processed in a similar way, except that a check is made to verify that there is enough room for all specified PRI's in a chain when a multiple PRI emitter is being entered. If the base emitter, the first emitter in the chain, is unavailable, an emitter already in system warning is issued. If the base is available, but there is not sufficient room for the entire chain, a

warning is issued that there are too many PRI's. All emitter events other than the Enter New Emitter event are checked to see that they reference only base emitters in multiple PRI chains. When a base emitter is turned on or off, for example, all of the emitters making up the multiple PRI chain are automatically included. References to emitters in the middle of a PRI chain are not allowed. Any events which are found to be in error are listed on the line printer with an appropriate warning message. As an added feature, an error limit can be specified. When the error limit is reached, the verification process is terminated. If no error limit is entered, a default value of 32,767, the integer limit of the machine, is used. In any case, the final error count is reported when the program exits. Note that no changes are made to the scenario during the verification process. Any events listed as being in error still remain in the scenario file. It is up to the operator to determine the action necessary to rectify the problem. No attempts should be made to run a realtime simulation using any scenario which has produced verification errors, as the simulation may produce unexpected results.

SECTION 3

SCENARIO LISTINGS

An independant program is provided to list scenario files. It is initiated by typing RUN LISCEN on the operator's console. The program then requests the name of the scenario file to be listed. A code is then requested to specify whether the list should be directed to the console, 1, or to the printer, 3. If a zero is entered, the program will exit. Time limits are then requested to specify the portion of the scenario to be listed. All events with times greater than or equal to the start time and less than or equal to the stop time will be listed. The listing format is identical to that of the event lists provided by the editing program, except that the data is presented more compactly. Units used are shown in Table II in Appendix A.

SECTION 4

PERIPHERAL INTERCHANGE PROGRAM

The Peripheral Interchange Program (PIP) is a utility program supplied by Digital Equipment Corporation as part of the RSX-11M operating system. It is designed to copy files from one device to another, delete unnecessary files, and list the names of existing files on a device. Some of the more pertinent commands will be discussed briefly here. For further details, reference the Utilities Procedures Manual, supplied as part of the RSX-11M system documentation. The program is initiated simply by typing PIP on the operator's console, and responds with a prompt, PIP>, whenever it is ready to accept a command. To terminate the program, strike a Z while the control key is depressed.

4.1 FILE SPECIFIERS

A complete file specifier includes a device name, user code, file name and type, and version number as shown below.

DEV:[UIC]NAME.TYP;N

The device specification includes a two letter device code and a unit number. For this system, the only valid device is DK, and unit numbers may include 0, 1, or 2. Note that a specification of DK0: is not required, as this is the default device. The colon is required whenever a device and unit number are specified. The UIC is an optional user identification code of the form [g,o] where g and o represent octal values in the range 1 to 377. The brackets are required only if a UIC is specified. The UIC is an identifier by which the Executive keeps track of file ownership; as the system is configured, the operator's console is always logged into account [11,1]. For convenience, all task files have been stored in account [11,2], and installed into the system. Since they are installed, the UIC need not be specified to execute the tasks. The terminal UIC of [11,1] is used as the default UIC whenever a UIC is not specified. As long as the default UIC is always used, the operator need not concern himself with account codes. The file name and extension, NAME.TYP specify the file name to be used. There is no default for the file name, and the extension defaults to a null type. The file name consists of from one to nine characters including the entire alphabet and

the numerals 0 to 9. The file extension consists of from one to three characters from the same character set. The extension is generally used to log all files of a given type with a common code. For example all scenario files may be named XXXXXXXXXX.SCN by convention, where X represents any legal character. The version number is represented by ;N. Each time a scenario is edited, a new file having the same name and a higher version number is created. When a file is specified, the default is to use the latest version number, unless an explicit version number is requested. Generally, only one version of each file is preserved, so no version number specifications need be used. The ; is required only when a version number is explicitly specified.

4.2 DIRECTORY LISTING

A directory listing prints the file names stored on a particular device and other associated data. The listing can be directed either to the operator's console or the line printer. It is issued as follows.

```
PIP> /LI
PIP> LP:=/LI
PIP> DK1:/LI
```

The first command will produce a listing on the console, while the second produces a permanent record on the line printer. The default device, DK0: will be used. The third form will produce a listing of all the files stored on device DK1: on the console. Directory listings are useful in determining what file names are already in use. Other information displayed includes file lengths, dates of file creation, and total disk space used.

4.3 DELETE FILES

PIP has the ability to delete files from permanent storage. When a file is deleted, the file name is removed from the directory and the space previously allocated to it is now marked as unused. In this instance, there is no default for the version number. It must be explicitly stated as part of the command.

```
PIP> DK1:NAME.TYP;3/DE
```

The /DE is the delete command. A wildcard feature included allows a set of files to be deleted with a single command.

```
PIP> DK1:NAME.TYP;*/DE
```

The above command will delete all versions of files named NAME.TYP on DK1: a subset of the delete command is the purge command. The purge command selectively deletes all files with a given name except the latest version.

```
PIP> DK1:NAME.TYP/PU
```

This command deletes all but the newest version of the file NAME.TYP on DK1:. A wild card feature allows all files in an account to be purged simultaneously.

```
PIP> DK1:*/PU
```

This command deletes all but the newest versions of all files stored on DK1:. Version numbers must never be specified in conjunction with the purge command. It is advisable to periodically purge all files on the disk.

4.4 COPY COMMAND

PIP can also be used to copy files from one device to another, or make duplicate copies on the same device. Whenever a scenario file is copied, it is preferable to force the new copy to occupy contiguous file space. Contiguous files are stored in consecutive blocks on the disk, which allows faster access by the real time software. The copy command is issued as follows.

```
PIP> DK1:NEW.SCN/CO=OLD.SCN  
PIP> NEW.SCN/CO=NEW.SCN  
PIP> DK1:=OLD.SCN
```

The first example copies OLD.SCN from the default device to DK1: under the name NEW.SCN, and the new file will be contiguous. The second command will simply create a new version of file NEW.SCN, with the latest version occupying contiguous disk space. The old file should be removed using the purge command. The

final command will copy OLD.SCN onto DK1:, using the same name as the default. The new file will not be contiguous unless the file OLD.SCN was originally contiguous. If sufficient contiguous space is not available, PIP will respond with an allocation failure message. The command may be retried without the /CO specification. If this also results in an allocation failure, the copy command cannot be processed until some space is created by deleting some files as described above.

4.5 FREE SPACE

PIP can also be used to determine how much free space is available on a device. This is useful when editing scenario files in determining if enough free space is available. The command format follows.

```
PIP> /FR  
PIP> DK1:/FR
```

In the first instance, PIP will list the amount of free space on DK0:, the default device. The second command will list the amount of free space on DK1:.

APPENDIX A

Reference Tables

TABLE I
SCENARIO EVENT DATA

EVENT	TYPE	WORDS/EVENT	EVENTS/BLOCK
1	Enter Platform	13	19
2	Delete Platform	3	85
3	Velocity Change	7	36
4	Platform Reposition	10	25
5	Enter Emitter	14*	18
6	Delete Emitter	3	85
7	Emitter Off	3	85
8	Emitter On	3	85
9	Change Emitter	5**	51

* Exact Length Depends on the Number of PRI's. Length Shown is Based on a Single PRI, and Each Additional PRI Requires one extra word.

** Length Varies According to Group Changed. Length of 5 is average.

TABLE II

UNITS

PLATFORM PARAMETERS	UNITS	LEGAL RANGE
East-West Position	Kilometers	-1000. - +1000.
North-South Position	Kilometers	-1000. - +1000.
Altitude	Kilometers	0. - +33.
Speed	Kilometers/Hour	0. - +5000.
Climb Rate	Meters/Second	-180. - +135.
EMITTER PARAMETERS	UNITS	LEGAL RANGE
Beam width	Degrees	0. - +180.
Lower Sector Boundary	Degrees	-180. ¹⁸⁰ - ¹⁸⁰ +180.
Sector Width	Degrees	0. - +360.
Radiated Power	DBM	0. - +255.
Threshold	DBM	0. - +255
Pulse Width	Microseconds	0. - +102.3
Pulse Jitter	Microseconds	0. - +4095.
Asynchronous Offset	Nanaseconds	0. - +800
Number of PRI's	-	+1 - +240
Pulse Repetition Interval	Microseconds	² +1 - +4095
Frequency	Megahertz	+500. - +18,000.
Frequency Agility Deviation	Megahertz	0. - +2000.*
Chirp Limit	Megahertz	*
Scan Period	Seconds	*
RF Selection Code	-	+10 - +57
Switch Select Code	-	0 - +124

* Legal Ranges for these parameters vary in response to other related parameters. Limits are computed as needed as based on the other related fields.

TABLE III
RF SELECTION CODES

CODE	DESCRIPTION
1	.5 -1. GHZ
2	1. -2. GHZ
3	2. -4. GHZ
4	4. -6. GHZ
5	6. -8. GHZ
6	8. -10. GHZ
7	10. -12. GHZ
8	12. -14. GHZ
9	14. -16. GHZ
10	16. -18. GHZ
11	Special Channel 1 Dedicated
12	Special Channel 2 Dedicated
13	Special Channel 1 Time Shared
14	Special Channel 2 Time Shared
15	Special Channel 1 and Special Channel 2

