Federal Communications Commission (FCC) Transponder Loading Data Conversion Software
User's Guide and Software Maintenance Manual
Version 1.2

Paul G. Mallasch
Analex Corporation
Brook Park, Ohio

January 1993

NASA
National Aeronautics and
Space Administration

Federal Communications Commission (FCC) Transponder

Loading Data Conversion Software

User's Guide and Software Maintenance Manual Version 1.2

Paul G. Mallasch
Analex Corporation
3001 Aerospace Parkway
Brook Park, Ohio 44142

Abstract

This volume contains the complete software system documentation for the Federal Communications Commission (FCC) Transponder Loading Data Conversion Software (FIX-FCC). This software was written to facilitate the formatting and conversion of FCC Transponder Occupancy (Loading) Data before it is loaded into the NASA Geosynchronous Satellite Orbital Statistics Database System (GSOSTATS). The information that FCC supplies NASA is in report form and must be converted into a form readable by the database management software used in the GSOSTATS application.

Both the User's Guide and Software Maintenance Manual are contained in this document.

This volume of documentation passed an independent quality assurance review and certification by the Product Assurance and Security Office of the Planning Research Corporation (PRC). The manuals were reviewed for format, content, and readability. The Software Management and Assurance Program (SMAP) life cycle and documentation standards were used in the development of this document. Accordingly, these standards were used in the review. Refer to the *System/Software Test/Product Assurance Report for the Geosynchronous Satellite Orbital Statistics Database System (GSOSTATS)* for additional information.

GSOSTATS Database

FCC Transponder Loading Data Conversion Software

Version 1.2, June 1992

Table of Contents

List of Figures

GSOSTATS Database

FCC Transponder Loading Data Conversion Software

User's Guide and Software Maintenance Manual

Version 1.2, June 1992

Paul G. Mallasch
Analex Corporation
3001 Aerospace Parkway
Brook Park, Ohio 44142

ABSTRACT

The Federal Communications Commission (FCC) supplies NASA Lewis
Research Center with quarterly reports on domestic communication
satellite usage. The FCC Automated Transponder Loading Report
provides data collected by an automated monitoring system
reflecting spectrum occupancy, a measure of spectrum utilization.
These reports are provided to Lewis as text files on floppy
diskette. Software was developed to facilitate automated
formatting, normalization, and error correction of the
transponder loading report file before it is loaded into the NASA
Geosynchronous Satellite Orbital Statistics Database System
(GSOSTATS). This document contains the User's Guide and Software
Maintenance Manual with information necessary for installation,
initialization, start-up, operation, and termination of the
software system. It also contains implementation details,
modification aids, and software code adaptations for use in
future revisions.

1.0 INTRODUCTION

1.1 Identification of Document

This is the User's Guide and Software Maintenance Manual for the
NASA Geosynchronous Satellite Orbital Statistics Database System
(GSOSTATS) - Federal Communications Commission (FCC) Transponder
Loading Data Conversion Software (FIX-FCC). It is only one
component of the complete GSOSTATS database system document set.

1.2 Scope of Document

This User's Guide and Software Maintenance Manual contains all
the information necessary for installation, initialization,
start-up, operation, and termination of the FIX-FCC software

system. It also contains all the implementation details, modification aids, and code adaptations of the software.

This document adheres to the NASA Software Management and Assurance Program (SMAP) documentation standards (Release 4.3) for a User's Guide and Software Maintenance Manual and is only applicable to the FIX-FCC software system. No information is provided on the actual GSOSTATS database application. A working knowledge of the basic features of the MS-DOS operating system is assumed, and specific knowledge of the BASIC language is necessary if the user wishes to modify the FIX-FCC source code.

1.3 Purpose and Objectives of Document

The purpose of the document is to provide a well organized, easily used guide for the user of the FIX-FCC software system. It is intended to guide the user through the steps necessary for installation, start-up, initialization, operation, and termination of the FIX-FCC program. Modifications to the main satellite data file and recovery from user errors are discussed. This document also presents the role that the FIX-FCC program performs in the overall GSOSTATS application. As a software maintenance manual, this document is intended to guide the developer through the details of the implementation and modification or code adaptation of the software source code.

1.4 Volume Status and Schedule

Release 1.2 is the third complete release for the FCC Transponder Loading Data Conversion Software Documentation. Modifications to Release 1.1 were required to include additional sections clarifying the information presented in this document along with modifications to the source code. The FIX-FCC software system supports GSOSTATS, so this document, along with the software, is a small and separate subset of the overall GSOSTATS application. Collecting all pertinent FIX-FCC software system documentation into one volume allows for easy reference.

No further updates are planned at this time, but should the FCC alter the format or content of the FCC Transponder Usage Data received by the NASA Lewis Research Center (LeRC), modifications to the FIX-FCC program may have to be made.

1.5 Volume Organization and Roll-Out

Sections 1 and 2 of this document identify it, describe its purpose, and cite other related documents. Section 3 provides an overview of the purpose and functions of the FIX-FCC software. Section 4 documents the installation procedures and

initialization process of the software system for the new user. Section 5 presents the software startup and termination procedures while Section 6 describes each function with its corresponding operation. Sections 7 and 8 contain the error and warning messages along with possible recovery steps the user may employ should an error occur. Section 9 contains a list of abbreviations and acronyms, and Section 10 a glossary. Section 11 is available for notes. Section 12 contains appendices showing the form of a representative FCC Transponder Usage Data sample, the master satellite data file, an overview of how FCC automated transponder occupancy reports are generated, samples of the report and file generated by the software, a software change request form, a data change request form, an error reporting form for undiscovered errors, and the software system source code.

2.0 RELATED DOCUMENTS

2.1 Parent Documents

None.

2.2 Applicable Documents

1. NASA Geosynchronous Satellite Orbital Statistics Database Maintenance Manual (Release 3.0), NASA Lewis Research Center, January 1992.

2.3 Information Documents

The following documents, although not directly applicable, amplify or clarify the information presented in this volume, and are not binding:

1. *Microsoft MS-DOS Version 5.0 User's Guide and Reference*, Microsoft Corporation, 1991.

2. *Microsoft QuickBASIC, Learning to use Microsoft QuickBASIC (Version 4.5)*, Microsoft Corporation, 1988.

3. *Microsoft QuickBASIC, Programming in BASIC (Version 4.5)*, Microsoft Corporation, 1988.

4. *NASA Geosynchronous Satellite Orbital Statistics Database User's Manual (Release 3.0)*, NASA Lewis Research Center, June 1991.

5. *NASA Geosynchronous Satellite Orbital Statistics Database Update System User's Manual (Release 1.0)*, NASA Lewis Research Center, June 1991.

6. *Geosynchronous Satellite Orbital Statistics Database Update System/Software Test/Produce Assurance Report*, Planning Research Corporation, Inc., November 1990.

3.0 OVERVIEW OF PURPOSE AND FUNCTIONS

3.1 GSOSTATS Database Overview

GSOSTATS is a computer-based information management system which maintains and allows easy access to information pertaining to various characteristics of in-orbit and planned geosynchronous communications satellites. GSOSTATS was developed using the INGRES Database Management System (DBMS) and is maintained on the NASA Headquarters VAX-Cluster. GSOSTATS is designed to serve a wide range of requirements while addressing the fundamental problem of accommodating several comprehensive and authoritative data sources which are sometimes in conflict.

3.2 FIX-FCC Software Overview

3.2.1 FIX-FCC Software Functions

Data for the GSOSTATS database is gathered from other established systems designed to collect and archive satellite information. GSOSTATS data updates are performed by a series of magnetic tape and floppy diskette file transfers from these other systems. The FCC is just one organization that NASA receives data from on a regular basis.

Every quarter, FCC provides NASA with an Automated Transponder Occupancy Report of Domestic Communications Satellites (DOMSATS). This report is basically a record of the percentage of time that each transponder is being used on each of the domestic satellites. Each data item provides spectrum occupancy data on a single satellite, hour by hour, transponder by transponder, for a total observation period approaching 24 hours. Refer to *Appendix D* for a description of how FCC calculates this information.

The main function of the FIX-FCC software system is to organize the FCC's data in such a way so as to allow the GSOSTATS data tables to be loaded with this new information. Satellite transponder data provided by the FCC is in report form contained on a floppy diskette. This report must be reorganized (or normalized) in such a way as to facilitate the data table loading and retrieval within the GSOSTATS Database. Once a new data file has been produced, it can be transferred to the NASA Headquarters VAX-Cluster in a variety of ways, including electronic file transfer. This new data file can then be loaded into the appropriate database tables. Please refer to the *NASA GSOSTATS Database Update System User's Guide* for additional information on loading the database tables.

Throughout the conversion process, the software will print a report outlining the satellites being processed, as well as information on errors encountered and how they were resolved. Refer to *Appendix C* for a sample output report. If the FIX-FCC software confronts a problem that it cannot fix, it will terminate execution with a detailed message. Refer to *Section 7.0* of this document for a complete guide to the error and warning messages.

3.2.2 FIX-FCC Software Options

FIX-FCC is primarily a data conversion utility with no substantial options. However, the user is able to physically modify the master satellite data file used by the FIX-FCC software. This data file contains a list of valid satellite family names, numbers, and other pertinent information. It is used by the software to verify the FCC data as it is being converted. The user may modify the data file to reflect additional satellites being sampled by the FCC or other items important to the operation of the FIX-FCC software. Refer to *Section 4.3* of this document for a description of how to modify the master satellite data file.

3.2.3 FIX-FCC Software Restrictions and Limitations

The FIX-FCC software system must read and format an ASCII text file so it relies on the present structure of the FCC data received on floppy diskette. Keep in mind that if the FCC changes the format of the original satellite transponder occupancy report, the software **may also** have to be modified to reflect those changes. Please refer to *Appendix A* for a description of the items used by the FIX-FCC software to organize the data conversion.

While a variety of errors occurring in the original FCC data have been taken into consideration by the developers, it is still possible to encounter an error that has not been considered. The FIX-FCC software is able to correct several types of errors during the conversion process, but should an unexpected error occur, the user should send the original FCC data file, the unfinished file containing the converted data, and a completed *Abnormal Errors Encountered Report Form* to the GSOSTATS database manager. Refer to *Appendix H* for instructions and a copy of the *Abnormal Errors Encountered Report Form*.

6

3.2.3.1 Non-Standard Satellite Numbers

Satellite numbers which are not in a standard format (e.g. a 0 through 9) must be handled as a special case within the FIX-FCC program source code. Roman numerals I through VI are converted automatically, however.

Satellite numbers which are not in a standard format (e.g. 0 through 9) are handled as special cases within the software. The SBS 1 and SBS 2 satellites are currently the only example of this special case. While the SBS 1 and SBS 2 satellites are two separate objects, they are co-located and treated as one entity know as SBS 1/2. Since the SBS 1/2 satellites are sampled as one item, they must be treated as one satellite, but still remain logically separate within the GSOSTATS database for user queries.

The FIX-FCC software system automatically filters out any extraneous characters (i.e., the slash character (/) would be removed) so SBS 1/2 would be converted to SBS 12 if not for the special conditional case. Any additions to the satellite sampling which contain a slash or any other extra characters must be treated as special cases and extracted before the normal satellite number filtering takes place. Refer to *Appendix I, SUB GetSatelliteInfo* for the actual pre-filtering source code.

3.2.3.2 Unordered Transponder Loading Data

The FCC Transponder Loading Report file provided on diskette may have to be edited for the software to work correctly. The ordering of the satellites sampled by the FCC and listed in the file must conform to the following standard:

1. All C-band, 36 MHz transponder loading data **must** be listed first, followed by;

2. All C-band, 72 MHz transponder loading data **must** be listed second, and;

3. All Ku-band transponder loading **must** be listed last.

This editing is required if the loading report file provided by FCC is not in the order listed above, or if the satellite sampling is split among two or more loading report files. The user can easily edit loading report files with the MS-DOS 5.0 editor (EDIT).

3.3 Implementation Details

3.3.1 Specific Data Representations

There are several specific data representations which must be adhered to during any modification to the FIX-FCC software system. Refer to *Appendices A, B, and E* for additional information on each of the data representations required for a working software system.

3.3.2 Operating System Interfaces and Dependencies

The FIX-FCC software system operates under Microsoft's MS-DOS (or fully compatible) operating system, version 3.1 or later.

3.3.3 Support Software and Libraries

The FIX-FCC software system requires only the **SATNAMES.DAT** Master Satellite Data File for system execution. This ASCII text file contains valid satellite family names, numbers, and other pertinent information to the successful conversion of the FCC data. Refer to *Appendix E* for more information on this data file.

> **Note:** The **SATNAMES.DAT** file **must** be present on the same disk drive and in the same directory as the **FIX-FCC.EXE** file in order for the program to function properly.

3.3.4 Hardware Dependencies

The FIX-FCC software system operates on an IBM (or fully compatible) personal computer.

A printer must be connected to the **LPT1:** parallel printer port of the microcomputer as a report will be printed automatically during the file conversion process.

This manual assumes that the microcomputer is equipped with a hard disk system that specifically minimizes the frequency of floppy diskette swapping.

3.3.5 Other Interfaces

None.

4.0 INSTALLATION AND INITIALIZATION

4.1 Equipment Requirements and Set-up

The FIX-FCC software system operates on an IBM (or fully
compatible) personal computer under Microsoft's MS-DOS operating
system, version 3.1 or later.

The software and related data files are distributed on one 5¼
inch floppy diskette labeled **FIX-FCC Software System**. A user may
also request a copy of the files on a 3½ inch floppy diskette.

A printer must be connected to the **LPT1:** parallel port of the
microcomputer as a report will be printed automatically during
the file conversion process.

This manual assumes that the microcomputer is equipped with a
hard disk system that specifically minimizes the frequency of
floppy diskette swapping.

4.2 Bootstrap and Loading of Software Files

The FIX-FCC software system consists of two separate, but related
files; **FIX-FCC.EXE** and **SATNAMES.DAT**. **FIX-FCC.EXE** is the actual
executable software while **SATNAMES.DAT** is an ASCII text file
containing valid satellite family names, numbers, and other
pertinent information.

> **Note:** The **SATNAMES.DAT** file **must** be present on the same
> disk drive and in the same directory as the **FIX-FCC.EXE** file
> in order for the software to function properly.

The following steps help you create a subdirectory within the
hard disk root directory from MS-DOS to copy the software files
into (user entered commands are in bold):

1. Turn on the computer

    *If you are prompted for the date and time, type the
    date, press Enter, type the time, and press Enter
    again.*

2. Create a directory named GSOSTATS.

    C:\> **MKDIR C:\GSOSTATS**

    *This allows the user to concentrate both the software
    and data files in one location on disk.*

3. Insert the **FIX-FCC Software System** software and data diskette into drive A.

4. Copy the software and data file from floppy diskette:

   C:\> COPY A:\FIX-FCC.EXE C:\GSOSTATS*.*

   C:\> COPY A:\SATNAMES.DAT C:\GSOSTATS*.*

   The user also has the option to locate the software and data files in another directory or leave them on the floppy diskette.

If any error messages are encountered during the installation procedure, check the command for spelling and/or syntax errors, then retype the command that failed. If the files cannot be successfully copied, please complete and return the *Abnormal Errors Encountered Report Form* contained in *Appendix F* of this document.

When the files have been successfully copied onto the hard disk, store the original floppy diskettes in a safe place for backup purposes. The software is now ready for use.

4.3 Modifying the SATNAMES.DAT Satellite Data File

The FIX-FCC software system accesses an ASCII data file named **SATNAMES.DAT** during the conversion process. This user accessible text file contains the valid satellite family names, satellite numbers, the type of frequency band (C-band, Ku-band, or a combination of both), the number of times the satellite is being sampled, the number of C-band/36 MHz transponders, the number of C-band/72 MHz transponders, and the number of Ku-band transponders. Refer to *Appendix E* for a complete list of the current satellites sampled by the FCC.

> **Note:** Should the FCC discontinue sampling one or more of the satellites listed in *Appendix E*, or if additional satellites begin to be sampled by the FCC, the **SATNAMES.DAT** file **must** be modified to reflect these changes. Other changes in the original FCC Transponder Loading Data may or may not require a modification of this data file.

Any standard text editor or word processor with the ability to read and write ASCII text files may be used to alter this data file. The user must, however, follow the current file format as outlined in *Appendix E* or software errors and erroneous results could result.

10

Note: The **SATNAMES.DAT** file **must** be present on the same disk drive and in the same directory as the **FIX-FCC.EXE** file in order for the software to function properly.

The user may easily modify the **SATNAMES.DAT** data file as many times as necessary. Possible FCC transponder loading data modifications include, but are not limited to the following:

1. An existing satellite or satellites are no longer being sampled by the FCC.

2. The FCC begins sampling one or more new satellites.

3. An existing satellite has one or more transponders removed from the FCC sampling.

The user must exercise caution when adding or modifying the records in the **SATNAMES.DAT** data file as software errors and erroneous results could result from improperly placed record items. Also note that the **SATNAMES.DAT** data file must be in ASCII format in order for the file to be read and utilized by the FIX-FCC software. Please refer to *Appendix E* for information on the correct record entry and position for each item listed in the data file.

4.4 Obtaining a Copy of the Software

The original and backup copies of the FIX-FCC software system reside at LeRC and are controlled by the GSOSTATS database manager. Additional copies of the FIX-FCC source software, **SATNAMES.DAT** data file, and related documentation may be obtained by contacting:

> James E. Hollansworth
> Mail Stop 54-2
> NASA Lewis Research Center
> 21000 Brookpark Road
> Cleveland, Ohio 44135
> Telephone: (216) 433-3458
> FAX: (216) 433-8705

5.0 STARTUP AND TERMINATION

5.1 Startup Procedures

To start the FIX-FCC software system, perform the following steps from within MS-DOS (user entered commands are in bold):

1. Turn on the computer and the printer and start MS-DOS.

   Note: The user may wish to put a write protect label on the FCC Transponder Usage Data input file prior to executing the FIX-FCC software. This will prevent the original data file from being inadvertently corrupted or destroyed.

2. Change to the directory where the data file **FIX-FCC.EXE** is located.

   C:\> **CHDIR C:\GSOSTATS**

3. Enter the name of the executable software file.

   C:\> **FIX-FCC**

You cannot start the FIX-FCC software from a directory other than where **FIX-FCC.EXE** is located because the data file **SATNAMES.DAT** must be accessed from the same location.

The FIX-FCC software system immediately accesses the **SATNAMES.DAT** satellite data file and the main menu screen in *Figure 5-1* is displayed on your terminal.

Figure 5-1 Main Menu Display Screen (Cursor is not visible.)

```
┌─────────────────────────────────────────────────────────────────────┐
│          * * *  N A S A  G S O S T A T S  D A T A B A S E  * * *       │
│                                                                       │
│          FCC Transponder Usage Data File Conversion Program           │
│                                                                       │
│                              WARNING                                  │
│  This program must format an ASCII text file so it relies on the present │
│  structure of the FCC data on the floppy diskette.  If the FCC changes │
│  the format of the original ASCII text file, this program MAY ALSO have │
│  to be modified to reflect those changes in order to work properly.   │
│                                                                       │
│                                                                       │
│             Input File Name:                                          │
│                                                                       │
│             Output File Name:                                         │
│                                                                       │
│  Please enter the input and output file names and make sure your printer │
│  is on-line as a report will be printed as the conversion takes place. │
│                                                                       │
│                                                                       │
│                                                                       │
│  Convert File    Input File Name    Output File Name    Help    Quit Program │
└─────────────────────────────────────────────────────────────────────┘
```

5.2 Normal Termination Procedures

To exit the FIX-FCC software system and return to MS-DOS, simply choose the *Quit Program* main menu option by typing a "Q". (Pressing **Enter** is not required.)

> **Note:** This method of exiting the software is only available when the software is **not** engaged in converting a data file.

If the software has terminated normally, the screen will clear and the user will be returned to MS-DOS without error messages.

5.3 Abnormal Termination Procedures

To exit the FIX-FCC software system during the data conversion processing or after the occurrence of some unknown software error, hold down the **Control** key and press the **Break** key (**Ctrl-Break**). This sequence of keystrokes will abort the current process regardless of what it is doing and return the user to MS-DOS.

> **Note:** This method of aborting the software will **not** change the original satellite data input file being converted, but the newly converted (and possibly incomplete) data output file will be unusable in the GSOSTATS database.

Aborting the software with **Ctrl-Break** in no way harms the original satellite data file or the **SATNAMES.DAT** data file. Any converted satellite data written to the user-defined output file before the abort will be available for the user's review.

5.4 Abnormal Restart Procedures

If the user aborts the conversion of satellite data or encounters some unknown software error that halts the program, the FIX-FCC software may be restarted by retyping the required commands found in *Section 5.1 Startup Procedures* of this document.

If the software cannot be successfully restarted, please complete and return the *Abnormal Errors Encountered Report Form* contained in *Appendix H* of this document.

5.5 Loading the Data Into the GSOSTATS Database

Once the FCC Transponder Usage Data has been successfully converted and a new data file has been produced, the normalized data can be transferred to the NASA Headquarters VAX-Cluster. This transfer can be accomplished in a variety of ways, but is not described in this document. The user may employ whatever method best satisfies his needs.

Refer to the *NASA Geosynchronous Satellite Orbital Statistics Database Update System User's Manual* for more information on converting, modifying, and loading the normalized FCC Transponder Usage Data into GSOSTATS.

6.0 FUNCTIONS AND THEIR OPERATION

6.1 Input File Name Function

6.1.1 Input File Name Function Purpose

The purpose of the *Input File Name* function is to allow the user to name the FCC Transponder Loading Data file to be converted. The original input file is completely separate from the output file and is left unchanged for archive purposes. File conversion cannot begin without first naming a valid input data file.

6.1.2 Input File Name Function Execution and Results

To enter a file name, choose the *Input File Name* menu option by typing an "**I**". (Pressing **Enter** is not required.) The cursor will immediately appear on the line reading **Input File Name:** where you should then enter the complete path and file name of the file you wish to convert. After entering the file name and striking the **Enter** key, the cursor disappears and the user may enter any main menu function, including this one.

If this field does not contain a valid MS-DOS file name and/or extension, an error message will result and the user will be prompted to enter the appropriate name. Refer to *Section 7.0* of this document for a description of the error and warning messages possible during software execution and file conversion.

Refer to *Figure 6-1* for an example of a screen with the input and output file names filled in.

6.2 Output File Name Function

6.2.1 Output File Name Function Purpose

The purpose of the *Output File Name* function is to allow the user to define the name of the FCC Transponder Loading Data File to receive the converted data. This file will either be created if it does not exist, or rewrite an existing file with the same name. The original input file is completely separate from the output file and is left unchanged for archive purposes. File conversion cannot begin without first naming a valid output data file.

6.2.2 Output File Name Function Execution and Results

To enter a file name, choose the *Output File Name* menu option by typing an "O". (Pressing **Enter** is not required.) The cursor will immediately appear on the line reading **Output File Name:** where you should then enter the complete path and file name of the file you wish to receive the converted data. After entering the file name and striking the **Enter** key, the cursor disappears and the user may enter any main menu function, including this one.

If this field does not contain a valid MS-DOS file name and/or extension, an error message will result and the user will be prompted to enter the appropriate name. Refer to *Section 7.0* of this document for a description of the error and warning messages possible during software execution and file conversion.

Please refer to *Figure 6-1* for an example of a screen with the input and output file names filled in.

Figure 6-1   Example File Names Display Screen

```
┌─────────────────────────────────────────────────────────────────┐
│          * * *  N A S A  G S O S T A T S  D A T A B A S E  * * *  │
│                                                                   │
│          FCC Transponder Usage Data File Conversion Program       │
│                                                                   │
│                              WARNING                              │
│  This program must format an ASCII text file so it relies on the present │
│  structure of the FCC data on the floppy diskette.  If the FCC changes │
│  the format of the original ASCII text file, this program MAY ALSO have │
│  to be modified to reflect those changes in order to work properly. │
│                                                                   │
│                                                                   │
│          Input File Name: A:\FCCAUTO.DOC                          │
│                                                                   │
│          Output File Name: C:\WORKAREA\NEWFCC.DAT                 │
│                                                                   │
│  Please enter the input and output file names and make sure your printer │
│  is on-line as a report will be printed as the conversion takes place. │
│                                                                   │
│                                                                   │
│  Convert File    Input File Name    Output File Name    Help    Quit Program │
└─────────────────────────────────────────────────────────────────┘
```

6.3 Convert File Function

6.3.1 Convert File Function Purpose

The purpose of the *Convert File* function is to begin the actual FCC transponder loading data file conversion process. The FIX-FCC software begins reading the records from the input file

16

specified by the user. Converted records are written to the output file specified by the user.

> **Note:** When processing is completed, the converted FCC Transponder Occupancy Report output data file will require several times the amount of disk space needed by the original data file. The user should insure enough storage space on the destination disk to permit a complete, reformatted file to be written.

During the conversion, transponder usage data found in the input file will be verified for valid satellite family names, satellite numbers, frequency band (C-band, Ku-band, or a combination of both), the number of times the satellite is being sampled, the number of C-band/36 MHz transponders, the number of C-band/72 MHz transponders, and the number of Ku-band transponders. Any discrepancies found by the software during the conversion process will be reported to the user. Please refer to *Appendix A* for a description of the required input file format, and *Appendix B* for an outline of the output file format.

Refer to *Section 7.0* of this document for a description of the error and warning messages possible during software execution and file conversion.

6.3.2 Convert File Function Execution and Results

In order to begin converting a file, the **Output File Name** and **Input File Name** fields **must** have been entered by the user. If these fields do not contain valid MS-DOS file names and/or extensions, an error message will result and the user will be prompted to enter the appropriate names.

If both the input and output file names have been filled in and the printer is turned on, begin the conversion process by choosing the *Convert File* option by typing a "**C**". (Pressing **Enter** is not required.) If everything is in order the following message will appear briefly on the display screen above the five main menu selections:

File conversion and report printing in progress - please be patient.

A report title with headers containing the following information will also be reproduced on the user's printer:

- Current time and date for future reference.

- Input file name as entered by the user.

- Output file name as entered by the user.

17

Following the information headers, the satellite family name, satellite number, possible warning messages, and possible correction messages are printed to the report as they are encountered. The information printed on this report allows the user to verify the correctness of the original FCC transponder loading data file before loading it into the GSOSTATS database. The verification of the satellite data is important in maintaining the integrity of the GSOSTATS database and alerting the user to possible errors in the FCC reports.

Please refer to *Appendix C* for an example of the report printed during the conversion process.

Should an error occur during processing that the user is able to correct, an alarm will sound and the user will be prompted for the necessary input. Refer to *Section 7.0* of this document for a thorough description of the error and warning messages possible during software execution and file conversion.

When a successful conversion is completed, the following message will be displayed on the screen and output to the printer:

> **+++ File conversion and report printing completed +++**

This message indicates that the data has been successfully corrected and converted, and is ready for loading into the GSOSTATS database. Should a catastrophic error of some type occur during execution, the subsequent error message would be displayed and printed:

> ***** Processing has been terminated *****

This message warns the user that there is a serious problem with the original FCC transponder loading data that requires attention before the converted data can be loaded into the GSOSTATS database. In addition to the preceding message, the satellite family name and number will be printed with the type of error encountered. Refer to *Section 7.0* of this document for a description of the error and warning messages possible during software execution and file conversion.

6.4   Help Function

6.4.1   Help Function Purpose

The purpose of the *Help* function is to aid the user in the proper execution of the software by displaying temporary help text. This function serves mainly as a quick reminder of the correct syntax for the input and output file names and for the proper operation of the software.

6.4.2 Help Function Execution and Results

To get help, choose the *Help* menu option by typing an **"H"**. (Pressing **Enter** is not required.) The help screen will immediately appear for review. After you are finished reading the help screen, simply press any key to return to the main menu. The user may access the help screen at any time **except** during the actual conversion process.

Please refer to *Figure 6-2* for a representation of the help display screen.

6.5 Quit Program Function

6.5.1 Quit Program Function Purpose

The purpose of the *Quit Program* function is to exit the FIX-FCC software system and return to MS-DOS. This function is the normal way of terminating execution of the software.

6.5.2 Quit Program Function Execution and Results

To exit the FIX-FCC software system and return to MS-DOS, simply choose the *Quit Program* main menu option by typing a **"Q"**. (Pressing **Enter** is not required.)

> **Note:** This method of exiting the software is only available when the program is **not** engaged in converting a data file.

If the software has been terminated normally, the screen will clear and the user will be returned to MS-DOS without any error messages.

Figure 6-2   Help Display Screen

```
╔════════════════════════════════════════════════════════════════════════╗
║          * * *  N A S A   G S O S T A T S   D A T A B A S E  * * *       ║
║                                                                          ║
║        FCC Transponder Usage Data File Conversion Program                ║
║                                                                          ║
║                              HELP                                        ║
║ You must enter the complete path name of the input file you wish to      ║
║ convert.                                                                 ║
║                  Input file Name: A:\FCCAUTO.DOC                         ║
║                                                                          ║
║ You will also have to enter the complete path name of the output file    ║
║ where you wish to have the converted data stored for later use.          ║
║                                                                          ║
║            Output File Name:  C:\WORKAREA\NEWFCC.DAT                     ║
║                                                                          ║
║ Once you have entered both file names you choose the Convert option and  ║
║ processing will begin.  A report will be printed listing the names of    ║
║ the satellites converted and any problems encountered during the         ║
║ process.  You may also change either file name before choosing Convert.  ║
║                                                                          ║
║             Press any key to return to the menu                          ║
╚════════════════════════════════════════════════════════════════════════╝
```

7.0 ERROR AND WARNING MESSAGES

7.1 Error Messages

Error messages are those resulting from improper input, or some user procedure that has been omitted. Errors of the following kind are **non-fatal** and require response from the user in order to correct them.

1. **ERROR - Please select from one of the menu choices below.**

   Cause: This message alerts the user that he has tried to choose a main menu option other than *Convert File, Input File Name, Output File Name, Help,* or *Quit Program.* Any other keys besides **C, I, O, H,** or **Q** (except **Ctrl-Break**) will sound an alarm and display this message.

   Action: Strike a **C, I, O, H,** or **Q.**

2. **ERROR - Please turn printer on-line for report printing.**

   Cause: This message alerts the user of a device fault, which is most likely a printer turned off or a printer turned off-line.

   Action: Turn the printer connected to the microcomputer on and make sure it's on-line.

3. **ERROR - Missing file name - please enter missing file name(s).**

   Cause: Either the input or output file name has not been specified by the user.

   Action: Enter a name for either the missing input or output file names.

4. **ERROR - File name invalid - please retype.**

   Cause: Either the input or output file name is in an illegal form. (e.g. the file name has too many characters.)

   Action: Retype the file name in question.

21

5. **ERROR - File does not exist - please retype.**

   Cause: Either the input or output file cannot be found in the current disk directory.

   Action: Check the disk directory for the file and retype the file name in question.

6. **ERROR - One of the files entered is already open for processing.**

   Cause: Either the input or output file has already been opened for processing and the software is attempting to open it again.

   Action: Terminate the FIX-FCC software and make sure no other processes are running in the background, then restart the software. Refer to *Section 5.3* for information on abnormal termination.

7. **ERROR - You may not have the file names equal to one another.**

   Cause: The input and output file names are referencing the same file.

   Action: Re-specify one or both of the file names. Refer to *Section 6.1* and *Section 6.2* for information on defining file names.

8. **ERROR - Family <Satellite Family Name> not found - Enter the correct name:**

   Cause: The software has encountered a satellite family name that is not in the **SATNAMES.DAT** data file. This usually indicates that the FCC has misspelled the name of a satellite in the original transponder loading data file.

   This message may also indicate the **SATNAMES.DAT** data file requires editing to include an additional satellite now being sampled by the FCC.

   Action: Type the satellite name you feel is correct. In most cases the user will be able to accurately guess the satellite family name. If the user cannot guess the satellite name the software should be aborted using **Ctrl-Break**.

7.2 Catastrophic Error Messages

Catastrophic error messages are those resulting from improper
input, or the omission of some important input data file element.
Errors of the following kind are **fatal** and will terminate normal
execution of the software. These errors also require some
investigation from the user in order to correct them. In most
cases the FCC transponder loading data file is seriously
corrupted and is at fault.

1. **CATASTROPHIC ERROR - The <Satellite Name and Number>
    satellite does not exist.**

    Cause: The software has encountered a satellite number
               that does not appear in the **SATNAMES.DAT** data
               file. The software terminates as a satellite
               number is required for processing. The user is
               unable to know the correct satellite number
               without reviewing the original data file.

    Action: Review the original FCC transponder loading data
               file and correct the satellite number in question
               if possible.

2. **CATASTROPHIC ERROR - The <Satellite Name and Number>
    transponder data is incorrect.**

    Cause: The software has encountered a satellite sample
               those transponder count does not match that given
               in the **SATNAMES.DAT** data file. This indicates
               either the original FCC transponder loading data
               file is in the wrong order, corrupt, or the
               SATNAMES.DAT file is incorrect.

    Action: Compare the original FCC transponder loading data
               file and the contents of the **SATNAMES.DAT** file
               with the known satellite transponder
               specifications. May require editing and
               reordering the original data file as specified in
               Section 3.2.3.2.

3. **CATASTROPHIC ERROR - The <Satellite Name and Number> satellite isn't in SATNAMES.DAT.**

    Cause: The software has encountered a satellite sample provided by the FCC which is not in the **SATNAMES.DAT** file. This indicates that the **SATNAMES.DAT** data file needs to be modified.

    Action: Compare the original FCC transponder loading data file and the contents of the **SATNAMES.DAT** file with the known satellite transponder specifications and modify the **SATNAMES.DAT** file as necessary.

7.3 Warning Messages

Warning messages are those resulting from transponder sampling data that has been omitted. Warnings of the following kind are **non-fatal** and require no response from the user. The user should take note of the missing data, however.

1. **WARNING - This satellite's transponder sample is missing <Number> hours(s).**

    Cause: The software has encountered a transponder sample in the original FCC transponder loading data file that has not been sampled for the full 24 hour period.

    Action: Take note of the missing sample for future reference. The converted data may still be loaded into the GSOSTATS database without consequences.

2. **WARNING - The FCC is missing <Number> satellite sample(s).**

    Cause: This message is in addition to the first warning message and contains the sum of the missing samples. The software has encountered one or more transponder samples in the original FCC transponder loading data file that have not been sampled for a full 24 hour period.

    Action: Take note of the missing sample(s) for future reference. The converted data may still be loaded into the GSOSTATS database without consequences.

7.4 Correction Messages

Correction messages are those resulting from satellite sampling data being altered to conform with GSOSTATS data standards. Corrections of the following kind are **non-fatal** and require no response from the user.

1. **CORRECTION - The <Misspelled Satellite Name> satellite has been changed to <Correct Satellite Name>.**

    Cause: The FCC has misspelled the family name of a satellite in the original Transponder Usage Data File. (The original data is entered by hand.)

    Action: The FIX-FCC software sounds an alarm, displays the misspelled satellite family name, then prompts the user to enter the correct spelling. The name entered by the user is then matched with a name in the **SATNAMES.DAT** data file for validity.

2. **CORRECTION - The satellite number <Satellite Number> has been changed to <Satellite Number>.**

    Cause: The FCC has entered a satellite number as a Roman numeral. The GSOSTATS database requires numbers be in the form: 0, 1, 2,... (The original data is entered by hand.)

    Action: The FIX-FCC software replaces the Roman numeral with the corresponding number. (e.g. IV to 4.)

8.0 RECOVERY STEPS

The user is able to easily recover from almost any of the corrections, warnings, or errors listed in *Section 7.0*. Only the catastrophic errors in *Section 7.2* cause the FIX-FCC software to terminate abnormally. The possible recovery methods available to the user for these types of failures follows.

1. Review the original FCC transponder loading data file and correct the satellite number in question if possible. This method is only possible in cases where the user is confident that the new satellite number is the correct one.

2. Compare the original FCC transponder loading data file and the contents of the **SATNAMES.DAT** file with the known satellite transponder specifications and modify the **SATNAMES.DAT** data file as necessary.

3. Edit the original FCC transponder loading data file to properly order the satellite transponder samples. Refer to *Section 3.2.3.2* for the correct ordering.

The user should seek an independent source to verify the satellite's physical characteristics any time a change is made.

Should the preceding methods fail, or if the user is unable to correct the original FCC transponder loading data file correctly, the user's only recourse is to notify the FCC of the problem with the original data file and request a corrected data file.

9.0 ABBREVIATIONS AND ACRONYMS

All abbreviations are defined when they first appear in the text.
An alphabetized list of the definitions for abbreviations and
acronyms used in this document is defined here.

ABBREVIATION	MEANING
ASCII	American Standard Code for Information Interchange
BASIC	Beginners All-Purpose Symbolic Instruction Code
dB	Decibel
DBMS	Database Management System
DOMSATS	Domestic Satellites
FCC	Federal Communications Commission
FIX-FCC	FCC Transponder Loading Data Conversion Software
GHz	Gigahertz
GSOSTATS	Geosynchronous Satellite Orbital Statistics Database System
IBM	International Business Machines
K	Kelvin (Degrees)
LeRC	Lewis Research Center
MHz	Megahertz
MS-DOS	Microsoft Disk Operating System
NASA	National Aeronautics and Space Administration
RF	Radio Frequency
SMAP	Software Management and Assurance Program

10.0 GLOSSARY

Bandwidth - The range of frequencies occupied by a signal, or passed by a transmission channel. A range of frequencies between upper and lower limits.

C-band - The bandwidth associated loosely with satellite services in the 6/4 GHz bands.

DataBase Management System - A set of procedures and data structures that isolates the applications from the details of the creation, retrieval, storage, modification, security, and physical storage structure of a computerized data base. It presents an application with a view, as required by its processing needs, without consideration for the physical storage or access of the data. The INGRES database management system from Relational Technology Inc. was used to develop GSOSTATS.

Ku-band - The bandwidth used by satellite systems employing frequencies between 10.7 and 18 GHz.

Normalization - A step-by-step reversible process of replacing a given collection of relations by successive collections in which the relations have a progressively simpler and more regular structure. The simplifying process is based on non-statistical criteria. The reversibility guarantees that the original collection of relations can be recovered and therefore no information has been lost.

Geosynchronous Satellite - An artificial satellite, placed in a circular orbit at a distance of 22,300 miles above the earth with a period of precisely one day. Launched toward the east (in the direction of the earth's rotation) the satellite will hover over one point on the earth.

Spectrum Occupancy - An indication or measure of the percentage of utilization or activity of a given transponder on a particular satellite over a specified period of time.

Transponder - The equipment in a communications satellite which receives signals from earth, then amplifies, translates, and retransmits these signals back to earth.

11.0 NOTES

The Transponder Occupancy Report is produced by the FCC for use within that organization and is provided to NASA as a free service. NASA receives a copy of the report on floppy diskette in a format that is basically a duplicate of hard copy or paper output. The FCC must prepare the report specifically for NASA every quarter.

Since the report is compiled manually, various errors may be introduced into the data received by NASA. The user will find that it is sometimes necessary to physically edit the report before the FIX-FCC software is able to successfully convert the data without errors. Any standard text editor or word processor with the ability to read and write ASCII text files may be used to correct this report file. The user must remember to follow the current file format (including the width of the current data file) as outlined in *Appendix A* or software errors and erroneous results could result.

Possible items that the user may have to edit in the Transponder Occupancy Report date file include, but are not limited to the following:

1. A satellite number has been entered incorrectly by the FCC and must be corrected before the FIX-FCC software can be executed. (The correct satellite number for the sample in question should be confirmed by the user.)

2. A satellite sample is not completely listed in the data file. (The sample may be missing one or more of the string constants necessary for the FIX-FCC software to convert the sampling data.)

3. A satellite sample may be listed out of order or more than once, requiring the user to reorder or remove the extra sampling. Only satellites with transponders greater than 36 MHz are sampled and listed twice in the FCC Transponder Occupancy (loading) Report. Refer to *Appendix D* for more information on the FCC sampling constraints and *Section 3.2.3.2* for additional information on satellite transponder ordering.

There may be occasions where a sample is so badly corrupted that the user needs to contact the FCC and request that a new report be produced. This method is often better than trying to correct a badly contaminated Transponder Occupancy Report data file.

12.0 APPENDICES

Appendix A

FCC Transponder Occupancy Report Input Sample

Explanatory Note

The purpose of the FCC Transponder Occupancy Report Input Sample is to document the format of the report received by LeRC on floppy diskette. The occupancy report is an ASCII text file organized in a specific way. The FIX-FCC software system relies on the present format and structure of the sample presented here. Any changes in the format of the original transponder occupancy report **may require** a modification of the FIX-FCC software.

The data sample in *Figure A-1* outlines the constants used as search strings by the FIX-FCC software. These constants and their relative positions in the text are candidates for possible modification should the FCC modify the structure of the transponder occupancy report.

Figure A-1 Representative Input Sample (String constants **bold**)

```
TYPE *.DOC
 Files copied:
DK:S50313.DOC  to TT:
```

```
     SATELLITE NAME: SATCOM 5          ORBITAL ARC POSITION: 143.00
           STARTING DATE:  3/13/89   STARTING TIME: 1000
           ENDING   DATE:  3/14/89   ENDING   TIME: 1000
                                               (EASTERN TIME)
```

OCCUPANCY OVER HOUR PERIOD (PERCENT)

	1	2	3	4	5	6	7	TRANSPONDER 8	9	10	11	···	19	20	21	22	23	24
HOUR																		
1000	100	0	100	0	98	0	100	0	100	0	0	···	0	0	0	0	100	0
1100	100	0	100	0	72	0	100	0	98	0	0	···	0	97	0	0	100	0
1200	100	0	100	0	79	0	100	0	72	0	0	···	0	6	0	0	100	0
.																		
.																		
.																		
700	100	0	100	0	67	0	100	0	100	0	0	···	0	0	0	0	100	0
800	100	0	100	0	62	0	100	0	100	0	0	···	0	42	0	0	100	0
900	100	0	100	0	70	0	100	0	100	0	0	···	0	1	0	0	100	0

OCCUPANCY OVER THE ENTIRE SAMPLING **PERIOD** = 31.0%

The following list outlines the search strings and their use
during the data conversion process (colons listed below are
considered part of the search string.)

Page   This word marks the beginning of a satellite sample and
           is used to search for the top of a transponder
           occupancy report. The satellite name is assumed to be
           on the next line in the text file.

NAME: This word designates the position where the software
           will find the satellite family name and number. The
           FIX-FCC software will extract the first 15 characters
           immediately following the space after the colon. The
           sample's start date and time are assumed to be on the
           next line in the text file.

DATE: This word marks the position where the date of the satellite sample is located. 8 characters representing the sample start date are extracted immediately following the space after the colon. The date is assumed to be DD/MM/YY, where DD represents the day, MM represents the month, and YY are the last two digits of the year.

TIME: This word is used to mark the position of the sampling start time. 5 characters are extracted immediately following the space after the colon. All times are in a 24 hour clock format. The time is assumed to be HH:MM, where HH represents the hour and MM represents the minutes.

HOUR   This word is used to designate the start of the actual transponder occupancy sampling. The hour of the sample and the percentage that the transponder was occupied are in a matrix following this string. The software will continue reading and processing these hours and percentages until the **PERIOD** search string is encountered.

PERIOD   This word marks the end of a satellite transponder occupancy sampling. Another satellite sampling may or may not follow this search string.

The preceding search strings are listed exactly as they appear in the current FCC transponder occupancy report. All characters in the report file are first converted to uppercase characters before any processing is performed.

The FCC manually generates the Transponder Occupancy Report with a computer software and is not planning any changes or modifications in the near future. Since only the satellite family name and number are entered by an operator, the content and format of the report should remain the same, allowing the FIX-FCC software system to successfully convert the data samples.

Appendix B

FCC Transponder Usage Output File Sample

Explanatory Note

The purpose of the FCC Transponder Usage Output File Sample is to document the format of the file produced by the FIX-FCC software system. The output file is produced as an ASCII text file organized in third normal form. This format allows for easy data validation, transfer, and INGRES database table loading. The organization of the data is divided into six fields, each separated by a comma. The following list describes the content of each field:

1. **Satellite Family Name** - The satellite common name.

2. **Satellite Number** - The satellite number.

> **Note:** The combination of satellite family name and number must be unique throughout the sampling report.

3. **Transponder Number** - The number of the transponder that is being sampled.

> **Note:** The characters 01 , 02 , 03 , ..., represent transponders numbers 1, 2, 3, etc. (this includes the trailing blanks preceding each comma). Transponder characters like 13A, 13B, 14A, 14B, ..., represent transponders numbers that are logically treated as two separate transponders, even though they are one physical unit.

> This numbering scheme is used only on satellites with transponders having a bandwidth greater than 36 MHz. Refer to *Appendix D* for an explanation of sampling restrictions for transponders over 36 MHz.

4. **Start Date and Time** - The date and time that the actual satellite sampling began. The date is in DD-MMM-YYYY format while the time is specified as a 24 hour clock in the form HH:MM:0. This organization corresponds with the format used by the INGRES DBMS.

5. **Start Time** - The time that this particular transponder was actually sampled. Only the hour portion of a 24 hour clock is used.

6. **Percent Utilized** - The percentage of time during the sampling period that this particular transponder was actually used. The number 100 represents 100% utilization.

The data in *Figure B-1* lists only a short sample of the actual data file output by the FIX-FCC software during the conversion of the ASC-1 satellite. The sample was taken on July 3, 1989 beginning at 11:00 a.m.. Only the 11:00 a.m. sample of transponders 1 through 12 (C band/36 MHz) and 13 through 18 (C band/72 MHz) are listed.

Please refer to *Section 5.5* for more information on loading the converted output file into the GSOSTATS database.

Figure B-1   Output File Sample

```
ASC,    1 ,01 ,3-Jul-1989 11:0:0,11,100
ASC,    1 ,02 ,3-Jul-1989 11:0:0,11,98
ASC,    1 ,03 ,3-Jul-1989 11:0:0,11,100
ASC,    1 ,04 ,3-Jul-1989 11:0:0,11,100
ASC,    1 ,05 ,3-Jul-1989 11:0:0,11,100
ASC,    1 ,06 ,3-Jul-1989 11:0:0,11,3
ASC,    1 ,07 ,3-Jul-1989 11:0:0,11,100
ASC,    1 ,08 ,3-Jul-1989 11:0:0,11,100
ASC,    1 ,09 ,3-Jul-1989 11:0:0,11,0
ASC,    1 ,10 ,3-Jul-1989 11:0:0,11,0
ASC,    1 ,11 ,3-Jul-1989 11:0:0,11,100
ASC,    1 ,12 ,3-Jul-1989 11:0:0,11,54
ASC,    1 ,13A,3-Jul-1989 11:0:0,11,100
ASC,    1 ,13B,3-Jul-1989 11:0:0,11,100
ASC,    1 ,14A,3-Jul-1989 11:0:0,11,100
ASC,    1 ,14B,3-Jul-1989 11:0:0,11,100
ASC,    1 ,15A,3-Jul-1989 11:0:0,11,40
ASC,    1 ,15B,3-Jul-1989 11:0:0,11,100
ASC,    1 ,16A,3-Jul-1989 11:0:0,11,100
ASC,    1 ,16B,3-Jul-1989 11:0:0,11,100
ASC,    1 ,17A,3-Jul-1989 11:0:0,11,35
ASC,    1 ,17B,3-Jul-1989 11:0:0,11,35
ASC,    1 ,18A,3-Jul-1989 11:0:0,11,34
ASC,    1 ,18B,3-Jul-1989 11:0:0,11,100
                        .
                        .
                        .
```

Appendix C

FIX-FCC Software Report Output Sample

Explanatory Note

The purpose of the FIX-FCC Software Report Output Sample is to document the format of the report that the FIX-FCC software produces during run time. This report will be printed automatically on the user's output device. *Figure C-1* represents a shorter version of what the output report might look like once the FIX-FCC software is finished processing a data file.

Figure C-1 Representative FIX-FCC Output Report

```
            * * *  N A S A   G S O S T A T S   D A T A B A S E  * * *

              FCC Transponder Usage Data File Conversion Program

Data file conversion statistics processed on 09-29-1989 at 12:36:00 AM.

Original Input File Name:  AUTO2.DOC

Converted Output File Name:  FCCLOAD.NO4

               +++ INFORMATION ON INDIVIDUAL SATELLITES FOLLOWS +++

BEGIN CONVERTING - The SATCOM 5 satellite is being operated on.
DONE  CONVERTING - The SATCOM 5 satellite has been successfully converted.

BEGIN CONVERTING - The SATCOM 3 satellite is being operated on.
      WARNING     - This satellite's transponder sample is missing 1 hour(s).
DONE  CONVERTING - The SATCOM 3 satellite has been successfully converted.

BEGIN CONVERTING - The SPACENET III satellite is being operated on.
DONE  CONVERTING - The SPACENET 3 satellite has been successfully converted.

BEGIN CONVERTING - The WESTSAR 4 satellite is being operated on.
      WARNING     - The WESTSAR satellite name has been changed to WESTAR.
DONE  CONVERTING - The WESTAR 4 satellite has been successfully converted.

BEGIN CONVERTING - The SPACENET III satellite is being operated on.
DONE  CONVERTING - The SPACENET 3 satellite has been successfully converted.

              +++ File conversion and report printing completed +++
```

Appendix D

Ascertaining FCC Automated Transponder Loading Reports

Explanatory Note

The FCC Automated Transponder Loading (Occupancy) Report provides data collected by an automated monitoring system reflecting spectrum occupancy, a measure of spectrum utilization as defined below. Each data sheet provides spectrum occupancy data on a single satellite, hour-by-hour, transponder-by-transponder, for a total observation period approaching 24 hours.

The automated monitoring system scans to the center frequency of each satellite transponder, measures RF carrier-to-noise spectral power level in a 36 MHz bandwidth* and records the transponder as occupied if the carrier-to-noise exceeds a .5 dB threshold. This transponder sampling is performed at an average rate of 2400 samples per hour. For the typical C-Band 24 transponder satellite, each transponder is sampled once every 36 seconds. The data provided, occupancy per transponder per hour in percent, is calculated from:

$$\frac{\text{The total active samples} \ast 100}{\text{The total samples}}$$

At the bottom of each data sheet "occupancy over the entire sampling period" is given. This percentage reflects cumulative occupancy for all transponders over the entire sampling period. For this calculation 72 MHz bandwidth transponders are considered to be two 36 MHz transponders.

Users of the data report should consider that some transponders may be utilizing spot or regional beam coverage, and that emissions from these transponders may not be visible to the observing earth station. Additionally, consider that the earth station figure of merit is approximated to be 19 dB/K at C-band and 22 dB/K at Ku-band, and that some active transponder's carrier-to-noise level may fall below the systems detection threshold.

> *Note: Equipment limitations restrict the sampling bandwidth to 36 MHz. Where transponder bandwidths are 72 MHz, sampling is performed by dividing the transponder bandwidth into two 36 MHz samples. Further notations and explanations will appear as necessary.

Appendix E

Master Satellite Data File SATNAMES.DAT

Explanatory Note

The **SATNAMES.DAT** Master Satellite Data File contains (in ASCII text format) the valid satellite family names, satellite numbers, the type of band (C-band, Ku-band, or a combination of both), the number of times the satellite is being sampled, the number of C-band/36 MHz transponders, the number of C-band/72 MHz transponders, and the number of Ku-band transponders.

> **Note:** The **SATNAMES.DAT** file **must** be present on the same disk drive and in the same directory as the **FIX-FCC.EXE** file in order for the software to function properly.

If the user wishes to modify the **SATNAMES.DAT** data file, the *Data Change Request Form* in *Appendix G* must be completed and returned to the GSOSTATS database manager. This allows the database administrator to track and monitor any and all modifications to the software and related data.

The user may employ any standard text editor or word processor with the ability to read and write ASCII text files to alter this data file. The user must, however, enter each of the following data items on a single line of text, separated by commas (one line equals the information for one satellite.)

1. **Satellite Family Name** - The satellite common name.

2. **Satellite Number** - The satellite number.

> **Note:** The combination of satellite family name and number must be unique throughout the sampling report.

3. **Type of Satellite** - The frequency classification of the satellite being sampled. The valid frequency codes are as follows: C = C-Band, K = Ku-Band, and H = Hybrid (Any combination of C-Band and Ku-Band.)

4. **Number of Samples** - The number of times that this satellite will be listed in the report. This number will be either a 1 or a 2 depending on a particular transponder. Only those satellites with transponders having a bandwidth greater than 36 MHz will have two samples listed. Refer to *Appendix D* for more information on the FCC sampling constraints.

5. **Number of C-Band/36 MHz Transponders** - The number of C-band transponders on the satellite whose bandwidth is 36 MHz. This number is in the range of 0 to 24.

6. **Number of C-Band/72 MHz Transponders** - The number of C-band transponders on the satellite whose bandwidth is 72 MHz. This number is in the range of 0 to 12.

7. **Number of Ku-Band Transponders** - The number of Ku-band transponders on the satellite. Bandwidth is ignored in this case. This number is in the range of 0 to 24.

Figure E-1 contains the present contents of the **SATNAMES.DAT** data file for use as a reference.

Figure E-1   Contents of SATNAMES.DAT Data File

```
ASC,1,H,2,12,6,6
GALAXY,1,C,1,24,0,0
GALAXY,2,C,1,24,0,0
GALAXY,3,C,1,24,0,0
GALAXY,6,C,1,24,0,0
GSTAR,1,K,1,0,0,16
GSTAR,2,K,1,0,0,16
GSTAR,4,K,1,0,0,16
SATCOM,1,C,1,24,0,0
SATCOM,2,C,1,24,0,0
SATCOM,3,C,1,24,0,0
SATCOM,4,C,1,24,0,0
SATCOM,5,C,1,24,0,0
SATCOM KU,1,K,1,0,0,16
SATCOM KU,2,K,1,0,0,16
SBS,1,K,1,0,0,10
SBS,2,K,1,0,0,10
SBS,3,K,1,0,0,10
SBS,4,K,1,0,0,10
SBS,5,K,1,0,0,14
SBS,6,K,1,0,0,19
SPACENET,1,H,2,12,6,6
SPACENET,2,H,2,12,6,6
SPACENET,3,H,2,12,6,6
SPACENET,4,H,2,12,6,6
TELSTAR,301,C,1,24,0,0
TELSTAR,302,C,1,24,0,0
TELSTAR,303,C,1,24,0,0
WESTAR,3,C,1,12,0,0
WESTAR,4,C,1,24,0,0
WESTAR,5,C,1,24,0,0
```

Appendix F

Software Change Request Form

Explanatory Note

A Software Change Request Form has been provided in the event that a user wishes to modify the operation or output of the FIX-FCC software system. Once software has been completed and placed under configuration control, it is important to thoroughly document each modification to the system. Each change has the potential of introducing new errors, and requires repeating the entire review, test, acceptance, and configuration control procedure. Software changes represent a major contribution to the software maintenance cost due to the large number of man-hours required to complete the process. It is therefore most important that software changes be well thought out and firmly established before coding begins.

Software changes may be required for a number of reasons. These may include changes in the requirements, input or output formats, as well as errors in the software that were missed in the acceptance testing, and improvements for reliability and quality. Modifications made for any reason must be traceable from proposal, through all reviews and tests, to the final acceptance and placement under configuration control.

All software changes must be requested in writing using a copy of the *Software Change Request Form* in *Figure F-1*. This form should then be submitted to the GSOSTATS database manager, who will be responsible for reviewing and possibly initiating the proposed change.

Figure F-1   Software Change Request Form

FIX-FCC Software System Change Request

Requested By:_____ Date:_____

Address:_____ Telephone:_____

_____

_____

Problem:_____

_____

_____

_____

_____

Description of Change:_____

_____

_____

_____

_____

Reason for Change:

___ Modified Requirement   ___ New Software
___ Programming Error   ___ Obtain Data
___ Reliability   ___ Flexibility
___ Other:_____

Change is: ___ Mandatory   ___ Improvement

Other Systems Affected:_____

Return the completed form   James E. Hollansworth
and any additional sheets to: Mail Stop 54-2
                                 NASA/Lewis Research Center
                                 21000 Brookpark Road
                                 Cleveland, Ohio 44135
                                 FAX: (216) 433-8705

40

Appendix G

Data Change Request Form

Explanatory Note

A Data Change Request Form has been provided in the event that a user wishes to modify the **SATNAMES.DAT** data file of the FIX-FCC software system. Once software has been completed and placed under configuration control, it is important to thoroughly document each modification to the system. Each change has the potential of introducing new errors, and requires repeating the entire review, test, acceptance, and configuration control procedure. Software and data changes represent a major contribution to the software maintenance cost due to the large number of man-hours required to complete the process. It is therefore most important that data modifications be well thought out and firmly established before any changes are made.

Changes to the **SATNAMES.DAT** data file may be required for a number of reasons. These may include changes in the actual FCC sampling, as well as errors in the data file that were missed in the acceptance testing, and improvements for reliability and quality. Modifications made for any reason must be traceable from proposal, through all reviews and tests, to the final acceptance and placement under configuration control.

All changes to the **SATNAMES.DAT** file must be requested in writing using a copy of the *Data Change Request Form* in *Figure G-1*. This form should then be submitted to the GSOSTATS database manager who will be responsible for reviewing and possibly initiating the proposed change.

Please refer to *Appendix E* for information on the correct record entry and position for each item listed in the **SATNAMES.DAT** data file.

Figure G-1 Data Change Request Form

FIX-FCC Software System Data Change Request

Requested By:_____ Date:_____

Address:_____ Telephone:_____

_____

_____

Problem:_____

_____

_____

_____

_____

Description of Change:_____

_____

_____

_____

_____

Reason for Change:

__ Modified Sampling   __ New Satellite Sampling
__ Satellite Removed
__ Other:_____

Change is: __ Mandatory   __ Improvement

Return the completed form   James E. Hollansworth
and any additional sheets to: Mail Stop 54-2
                                   NASA/Lewis Research Center
                                   21000 Brookpark Road
                                   Cleveland, Ohio 44135
                                   FAX: (216) 433-8705

Appendix H

Abnormal Errors Encountered Report Form

Explanatory Note

An Abnormal Errors Encountered Report Form has been provided in the event that a user encounters an undocumented error or problem during the operation of the FIX-FCC software system.

While a variety of errors occurring in the original FCC data have been taken into consideration by the developers, it is still possible to encounter an error that has not been considered. The FIX-FCC software is able to correct several types of errors during the conversion process, but should an unexpected error occur, the user should send the original FCC data file, the unfinished file containing the converted data, and a completed *Abnormal Errors Encountered Report Form* to the GSOSTATS database manager.

Documenting all software errors will allow the developers of the FIX-FCC software to correct problems and make the necessary modifications to the software code in a timely and useful manner. Refer to *Figure H-1* for a copy of the report form.

Figure H-1 Abnormal Errors Encountered Report Form

Abnormal Errors Encountered Report Form

Discovered By:_____ Date:_____

Address:_____ Telephone:_____

        _____

        _____

Problem:_____

_____

_____

_____

_____

Description of Error Messages (If Any):_____

_____

_____

_____

_____

_____

Type of PC Used to Run the Software:_____

Version of MS-DOS Being Used:_____

Memory Resident Software Being Used (If Any):_____

Other Information:_____

| Return the completed form and all additional sheets to: | James E. Hollansworth
Mail Stop 54-2
NASA/Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135
FAX: (216) 433-8705 |

44

Appendix I

FIX-FCC Software System Source Code

Explanatory Note

The purpose of the FIX-FCC Software System Source Code is to document the actual BASIC source code used to develop the working program. The code was developed using Microsoft QuickBASIC Version 4.5. QuickBASIC was the language of choice due to the rich variety of string processing functions available. Since the FIX-FCC software manipulates a data text file and the format of this data file may be changed, an easy to use development language was chosen to ease future modifications.

```
'*************************** Program Description ****************************
' FIX-FCC.BAS - This program will read the FCC transponder usage data
' as an ASCII file and format it into a new ASCII file containing only the
' pertinent data from the original file.  The FCC transponder usage data is
' received from the FCC on a floppy diskette which must be formatted into a
' INGRES-readable form for loading into the GSOSTATS Database tables.  This
' new ASCII file can then be uploaded to the NASA Hq VAX-Cluster.
'
' WARNING - This program must format an ASCII text file so it relies on the
' present structure of the FCC data on the floppy diskette.  If the FCC
' changes the format of the original ASCII text file, this program MAY ALSO
' have to be modified to reflect those changes.
'
' Written:        September, 1989.        In: MicroSoft QuickBASIC V4.50
'
' Modified:       March, 1990.            By: P.G. Mallasch/Analex Corp.
' Description:    Changed the wording of several screen and error messages.
'
' Modified:       June, 1992.             By: P.G. Mallasch/Analex Corp.
' Description:    Added code in GetSatelliteInfo function to check for SBS 2
'                 satellite and change it to SBS 1.  Added code in
'                 CheckSatellite function to pad satellite numbers with 5
'                 blank spaces so they match the USAF data satellite numbers.
'
' Author:         Paul G. Mallasch/Analex Corporation
' Project:        NASA GSOSTATS Database Enhancement Project
' Manager:        Jim Hollansworth
'                 Space Communications Branch
'                 NASA Lewis Research Center
'                 21000 Brookpark Road, MS 54-2
'                 Cleveland, Ohio  44135
'                 (216) 433-3458 or FTS 297-3458
'
' Input:          Keyboard - User commands for program operation.
'                 Files:
'                     User defined - FCC transponder usage data file.
'                     SATNAMES.DAT - Valid satellite names and transponders file.
'
' Output:         Screen  - Program operation prompts.
'                 Printer - File formatting statistics.
'                 File:
'                     User defined - Formatted FCC transponder usage data file.
'
'********************* Declarations and Definitions ************************

' Resets the default data type from single precision to integer.

DEFINT A-Z

' Miscellaneous symbolic constants for cursor and color control.

CONST False = 0, True = NOT False
CONST CursorOff = 0, CursorOn = 1
CONST Cyan = 3, Black = 0, White = 7, Bright = 8
CONST BackGround = Cyan, Normal = Black, HiLite = White + Bright

' Line numbers for input and output file name positions on screen.

CONST FirstPlace = 14, SecondPlace = 16
```

```
' Maximum number of satellites in SATNAMES.DAT file.

CONST MaxSats = 50

' String constants used as search strings in the FCC transponder usage data
' file.  This is one place that may have to be modified if the FCC changes
' the format of their data file.

CONST PageString$ = "PAGE"
CONST HourString$ = "HOUR"
CONST NameString$ = "NAME:"
CONST DateString$ = "DATE:"
CONST TimeString$ = "TIME:"
CONST PeriodString$ = "PERIOD"

CONST SatelliteNameFile$ = "SATNAMES.DAT"           ' Valid satellite names file.

' Declare a user-defined type to hold information about valid satellites.
' This information is contained in the ASCII file SATNAMES.DAT

TYPE SatelliteType
        SatName AS STRING * 15      ' Satellite family name.
        SatNum AS INTEGER           ' Satellite number
        TypeOfBird AS STRING * 1    ' C = C-Band, K = Ku-Band, H = Hybrid.
        NumSampled AS INTEGER       ' Number of times this bird will be sampled.
        NumC36Band AS INTEGER       ' Number of C-Band/36 Mhz transponders.
        NumC72Band AS INTEGER       ' Number of C-Band/72 Mhz transponders.
        NumKuBand AS INTEGER        ' Number of Ku-Band transponders.
END TYPE

' Declare a user-defined type to deal with the dual sampling problem.  If a
' satellite is sampled twice the flag will be set to true and separate
' transponder processing is performed.

TYPE SatTrackType
        NameSat AS STRING * 15      ' Satellite name.
        NumOfSat AS INTEGER         ' Number of seperate satellite samples.
        Done AS INTEGER             ' Flag true when one sample is done.
END TYPE
```

```
' Declare the variables for the files.   NOTE: These are global variables.

DIM SHARED Inputfile AS STRING          ' Input file name.
DIM SHARED OutputFile AS STRING         ' Output file name.
DIM SHARED NumNames AS INTEGER          ' Number of valid names.
DIM SHARED NumberSats AS INTEGER        ' Number of valid satellites.
DIM SHARED Satellites(1 TO MaxSats) AS SatelliteType    ' Valid name array.
DIM SHARED NumSats(1 TO MaxSats) AS SatTrackType        ' Dual sample array.

' Declare the various SUB procedures and FUNCTIONs.

DECLARE SUB Alarm ()
DECLARE FUNCTION CheckName% (NameToCheck$)
DECLARE SUB CheckSatellite (SatelliteFamily$, SatelliteNumber$,
                            SatelliteName$)
DECLARE SUB DisplayMessage (MessageString$)
DECLARE SUB DrawMenu ()
DECLARE SUB DrawScreen (True)
DECLARE FUNCTION Filter$ (Text$, FilterString$)
DECLARE FUNCTION FixTransponder$ (Counter%, SatNum%, DualFlag)
DECLARE FUNCTION FoundSat% (SatName$, SatNum$)
DECLARE FUNCTION GetDate$ (DateText$)
DECLARE SUB GetFileName (RowNumber%, FileString$, FileNum)
DECLARE SUB GetNames (Num%, TotalSamples%)
DECLARE SUB GetSatelliteInfo ()
DECLARE FUNCTION GetTime$ (TimeText$)
DECLARE SUB GetTransponders (SatelliteFamily$, SatelliteNumber$, StartDate$,
                            StartTime$)
DECLARE SUB LookForString (PageString$)
DECLARE SUB PrintHeader ()
```

48

```
'*************************** Program Mainline ******************************

' Initialize the screen and print border then prompt user for the name of
' the input and output file names.  Also setup error handler and read in the
' set of good satellite names and corresponding numbers for use later.
' Initialize an array used to track dual samples.  Open the printer on
' LPT1: for report printing.

ON ERROR GOTO Handler                              ' Error handler.
LET NumberSats = 0: LET TotalSamples = 0
LET Inputfile = "": LET OutputFile = ""
FOR Increm = 1 TO MaxSats STEP 1                   ' Initialize array.
        LET NumSats(Increm).NameSat = ""
        LET NumSats(Increm).NumOfSat = 0
NEXT Increm
CALL GetNames(NumNames, TotalSamples)              ' Read in valid names.
RESTORE ScreenText
CALL DrawScreen(False)                             ' Draw screen.
CALL DrawMenu
OPEN "LPT1:" FOR OUTPUT AS #4                       ' Open printer for report.

' Infinite loop to keep menu display on screen, includes a label for errors.

TheBigLoop:
DO
        ' Get the user's menu choice.

        DO
            Choice$ = INKEY$
        LOOP WHILE Choice$ = ""

        ' Execute the proper menu choice else error.

        SELECT CASE UCASE$(Choice$)

            ' Convert the input file and write it to the output file if the
            ' file names have both been filled in correctly.

            CASE "C"
               IF FILEATTR(1, 1) = 1 AND FILEATTR(2, 1) = 2 THEN
                     RESTORE StartConversion          ' Display message.
                     READ Text$
                     CALL DisplayMessage(Text$)
                     CALL PrintHeader               ' Printer title to printer.

                     WHILE NOT EOF(1)
                         CALL LookForString(PageString$) ' Look for a page.
                         IF NOT EOF(1) THEN
                                 CALL GetSatelliteInfo  ' Read & convert data.
                                 LET TotalSamples = TotalSamples - 1
                         END IF
                     WEND
```

```
                    ' Print a warning if the number of samples does not
                    ' agree with the file SATNAMES.DAT.

                    IF TotalSamples > 0 THEN
                          PRINT #4,
                          PRINT #4, TAB(8); "WARNING - The FCC is missing";
                                TotalSamples; "satellite sample(s). - WARNING"
                    END IF
                    PRINT #4,
                    PRINT #4, TAB(13);
                       "+++ File conversion and report printing completed +++"
                    PRINT #4,
                    RESTORE EndConversion              ' Display message.
                    CALL Alarm
                    READ Text$
                    CALL DisplayMessage(Text$)
              END IF

         ' Prompt user for the name of the input file.

         CASE "I"
              CALL GetFileName(FirstPlace, Inputfile, 1)

         ' Prompt user for the name of the output file.

         CASE "O"
              CALL GetFileName(SecondPlace, OutputFile, 2)

         ' Display the help screen to the user.

         CASE "H"
              RESTORE HelpText
              CALL DrawScreen(False)              ' Draw help screen.
              DO
                    Choice$ = INKEY$               ' Wait for keystroke.
              LOOP WHILE Choice$ = ""
              RESTORE ScreenText
              CALL DrawScreen(True)               ' Draw original screen.
              CALL DrawMenu

         ' Terminate the program.

         CASE "Q"
              CLOSE
              LOCATE , , CursorOn
              CLS
              END

         ' Wrong choice so display error message and ring alarm.

         CASE ELSE
              CALL Alarm
              RESTORE ErrorText
              READ Text$
              CALL DisplayMessage(Text$)
      END SELECT
LOOP
END
```

```
Handler:
'*******************************************************************************
' This label section will use ERR to determine which error caused the branch
' to this SUB.  An error message will be displayed to the user and the
' execution will proceed from the statement following the error.
'
' Parameters: NONE
'
' Output: The appropriate error message.
'*******************************************************************************

SELECT CASE ERR

     ' Device fault; printer most likely off-line.

     CASE 25
          RESTORE PrinterOff
          READ Text$
          CALL Alarm
          CALL DisplayMessage(Text$)
          RESUME

     ' Bad file name or number; name most likely not entered by user.

     CASE 52
          RESTORE BadFileNumber
          READ Text$
          CALL Alarm
          IF Inputfile = "" THEN
                    CALL DisplayMessage(Text$)
                    CALL GetFileName(FirstPlace, Inputfile, 1)
          ELSEIF OutputFile = "" THEN
                    CALL DisplayMessage(Text$)
                    CALL GetFileName(SecondPlace, OutputFile, 2)
          END IF
          RESUME TheBigLoop

     ' File name not found.

     CASE 53
          RESTORE NoSuchFile
          READ Text$
          CALL Alarm
          CALL DisplayMessage(Text$)
          CALL GetFileName(FirstPlace, Inputfile, 1)
          RESUME NEXT

     ' File already open for use.

     CASE 55
          RESTORE FileOpen
          READ Text$
          CALL Alarm
          CALL DisplayMessage(Text$)
          RESUME NEXT
```

51

```
     ' Bad file name; extra characters or too long.

     CASE 64
          RESTORE BadFileName
          READ Text$
          CALL Alarm
          CALL DisplayMessage(Text$)
          RESUME NEXT

          ' An unanticipated error has occurred; display the message for that
          ' error and stop the program.

     CASE ELSE
          CALL Alarm
          ON ERROR GOTO 0

END SELECT

'*************************************************************************
' This part of the program contains various data statements for errors as
' well as help messages.  The data text is first RESTOREd using the label
' preceding each message, then displayed to the user.
'*************************************************************************

ErrorText:
DATA"          ERROR - Please select from one of the menu choices below."

PrinterOff:
DATA"          ERROR - Please turn printer on-line for report printing."

BadFileNumber:
DATA"       ERROR - Missing file name - please enter missing file name(s)."

BadFileName:
DATA"             ERROR - File name invalid - please retype."

NoSuchFile:
DATA"             ERROR - File does not exist - please retype."

FileOpen:
DATA"      ERROR - One of the files entered is already open for processing."

' Data statements for start and end of processing.

StartConversion:
DATA"   File conversion and report printing in progress - please be patient."

EndConversion:
DATA"             File conversion and report printing completed."
```

```
' Data statements for screen output initialled for 25 rows.  First number
' is the number of data statements following it.

ScreenText:
DATA 17
DATA"           * * *  N A S A   G S O S T A T S   D A T A B A S E  * * *"
DATA" "
DATA"            FCC Transponder Usage Data File Conversion Program"
DATA" "
DATA"                                WARNING"
DATA" This program must format an ASCII text file so it relies on the present"
DATA" structure of the FCC data on the floppy diskette.  If the FCC changes"
DATA" the format of the original ASCII text file, this program MAY ALSO have"
DATA" to be modified to reflect those changes in order to work properly."
DATA" "
DATA" "
DATA"               Input File Name:"
DATA" "
DATA"             Output File Name:"
DATA" "
DATA" Please enter the input and output file names and make sure your printer"
DATA" is on-line as a report will be printed as the conversion takes place."

HelpText:
DATA 20
DATA"           * * *  N A S A   G S O S T A T S   D A T A B A S E  * * *"
DATA" "
DATA"            FCC Transponder Usage Data File Conversion Program"
DATA" "
DATA"                                  HELP"
DATA" You must enter the complete path name of the input file you wish to"
DATA" convert."
DATA"                  Input File Name: A:\FCCAUTO.DOC"
DATA" "
DATA" You will also have to enter the complete path name of the output file"
DATA" where you wish to have the converted data stored for later use."
DATA" "
DATA"                  Output File Name: C:\WORKAREA\NEWFCC.DAT"
DATA" "
DATA" Once you have entered both file names you choose the Convert option and"
DATA" processing will begin.  A report will be printed listing the names of"
DATA" the satellites converted and any problems encountered during the"
DATA" process.  You may also change either file name before choosing Convert."
DATA" "
DATA"                    Press any key to return to the menu"
```

```
SUB Alarm
'*******************************************************************************
' This procedure uses the SOUND statement to send signals to the computer's
' speaker and sound an alarm.
'
' Parameters: NONE
'
' Output: Sends an alarm to the user.
'*******************************************************************************

FOR Tone = 600 TO 2000 STEP 40
        SOUND Tone, Tone / 7000
NEXT Tone
END SUB
```

```
FUNCTION CheckName (NameToCheck$)
'**************************************************************************
' This function will loop through the array of valid satellite names and
' compare those with the name passed to it.  If a match is found (satellite
' name is okay) then the function returns a true otherwise a false (satellite
' name was not found, possible misspelling).
'
' Parameters: NameToCheck$ - String containing name to compare.
'
' Output: A true or false depending upon search.
'**************************************************************************

FOR Increm = 1 TO NumNames STEP 1
        IF NameToCheck$ = RTRIM$(Satellites(Increm).SatName) THEN
                LET CheckName = True
                EXIT FOR
        ELSE
                LET CheckName = False
        END IF
NEXT Increm
END FUNCTION
```

```
SUB CheckSatellite (SatFamilyName$, SatNumber$, SatelliteName$)
'****************************************************************************
' This procedure will examine the satellite's family name and number and
' change it if it is not valid.  If the satellite family name is not among
' those listed in the data file SATNAMES.DAT then the user will be asked to
' enter the correct name.  It will also change the satellite number from a
' Roman numeral to an Arabic one if necessary (e.g. V -> 5).  If the SATCOM
' KU family is found the KU will be appended to the end of the satellite
' family name.  If an invalid satellite number appears the program will
' display an error message and end the program.  Pad the satellite numbers
' spaces so make it 5 characters to match USAF data.  Various items on the
' output report will also be printed depending on the action taking place
' here.
'
' Parameters: SatFamilyName$ - String containing satellite family name.
'             SatNumber$     - String containing satellite number.
'             SatelliteName$ - String containing original satellite name.
'
' Output: The corrected satellite name and number if needed.
'****************************************************************************


LET SatNumHere = False
LET ChangedNum = False
LET WhereSpace = INSTR(SatFamilyName$, " ")
IF WhereSpace > 0 THEN

        ' Two-part satellite family name so check out name and number.

        LET TempName$ = LEFT$(SatFamilyName$, (WhereSpace - 1))
        LET TempNum$ = RIGHT$(SatFamilyName$, (LEN(SatFamilyName$) -
                    LEN(TempName$) - 1))

        ' Fix the satellite number in a two part name if it needs it.

        SELECT CASE TempNum$
                CASE "I"
                        LET SatNumber$ = "1"
                        LET ChangedNum = True
                CASE "II"
                        LET SatNumber$ = "2"
                        LET ChangedNum = True
                CASE "III"
                        LET SatNumber$ = "3"
                        LET ChangedNum = True
                CASE "IV"
                        LET SatNumber$ = "4"
                        LET ChangedNum = True
                CASE "V"
                        LET SatNumber$ = "5"
                        LET ChangedNum = True
                CASE "VI"
                        LET SatNumber$ = "6"
                        LET ChangedNum = True
                CASE "KU"
                        LET TempName$ = TempName$ + " KU"
                CASE ELSE
                        LET SatNumHere = True
        END SELECT
```

56

```
ELSE
        LET TempName$ = SatFamilyName$
END IF
IF ChangedNum = True THEN PRINT #4, TAB(7); "CORRECTION - The satellite number
"; TempNum$; " has been changed to "; SatNumber$; "."

' Now check the name against the list of actual names, prompt the user if
' it's not found and return the new satellite family name.

IF NOT CheckName(TempName$) THEN
        CALL Alarm
        LOCATE 21, 5, CursorOn
        COLOR HiLite, BackGround
        LET Xname$ = TempName$
        PRINT "ERROR - Family "; TempName$; " not found - ";
        INPUT "Enter the correct name: ", TempName$
        LET TempName$ = UCASE$(TempName$)
        LOCATE 21, 4, CursorOff
        COLOR Normal, BackGround
        PRINT STRING$(74, 32)
        PRINT #4, TAB(7); "CORRECTION - The "; Xname$;
                " satellite name has been changed to "; TempName$; "."
        CALL CheckSatellite(TempName$, SatNumber$, SatelliteName$)
END IF
LET SatFamilyName$ = TempName$

' Check the satellite number for validity.  If an improper number appears,
' end the program in failure as a number is required and the user is unable
' to know the correct number without reviewing the original file or possibly
' contacting the FCC.

IF SatNumHere = True THEN
        LET TempNum$ = Filter$(SatNumber$, "0123456789")
        IF LEN(TempNum$) < 1 THEN
                CALL Alarm
                LOCATE 21, 10, CursorOn: COLOR HiLite, BackGround
                PRINT "CATASTROPHIC ERROR - The ";
                        RTRIM$(LTRIM$(SatelliteName$));
                        " satellite does not exist.";
                LOCATE 22, 21
                PRINT "*** Processing has been terminated ***"
                PRINT #4, TAB(7); "CATASTROPHIC ERROR - The ";
                        RTRIM$(LTRIM$(SatelliteName$));
                        " satellite does not exist."
                PRINT #4,
                PRINT #4, TAB(21); "*** Processing has been terminated ***"
                CLOSE
                CALL Alarm
                END
        END IF
END IF
```

```
' Change the Satellite Number to 5 characters (padded with blank spaces)
' so that it will follow the same format as the US Air Force Information.

SELECT CASE LEN(SatNumber$)
        CASE 1
                SatNumber$ = "    " + SatNumber$ + " "
        CASE 2
                SatNumber$ = "   " + SatNumber$ + " "
        CASE 3
                SatNumber$ = " " + SatNumber$ + " "
        CASE 4
                SatNumber$ = SatNumber$ + " "
        CASE ELSE
                SatNumber$ = SatNumber$
END SELECT

END SUB
```

```
SUB DisplayMessage (MessageString$)
'****************************************************************************
' This procedure will display the message text passed to it for several
' seconds.
'
' Parameters: MessageString$ - a string containing the message to print.
'
' Output: A message on the screen.
'****************************************************************************

LOCATE 21, 4
COLOR HiLite, BackGround: PRINT MessageString$
SLEEP 6
LOCATE 21, 4
COLOR Normal, BackGround: PRINT STRING$(74, 32)
END SUB
```

```
SUB DrawMenu
'****************************************************************************
' This procedure will draw the user menu at the bottom of the screen.  The
' first letter of each menu item is highlighted to indicate that the user
' must simply press that letter to choose that menu item.
'
' Parameters: NONE
'
' Output: The user menu on the screen.
'****************************************************************************

LOCATE 23, 6
COLOR HiLite, BackGround: PRINT "C";
COLOR Normal, BackGround: PRINT "onvert File   ";
COLOR HiLite, BackGround: PRINT "I";
COLOR Normal, BackGround: PRINT "nput File Name    ";
COLOR HiLite, BackGround: PRINT "O";
COLOR Normal, BackGround: PRINT "utput File Name   ";
COLOR HiLite, BackGround: PRINT "H";
COLOR Normal, BackGround: PRINT "elp   ";
COLOR HiLite, BackGround: PRINT "Q";
COLOR Normal, BackGround: PRINT "uit Program";
END SUB
```

```
SUB DrawScreen (SeeFiles)
'*********************************************************************
' This procedure will initialize the screen by setting the color, setting the
' width and height, clearing the screen, and hiding the cursor.  Then writes
' the screen that was RESTOREd just before this procedure was called.
'
' Parameters: SeeFiles - Boolean to indicate if file names are to be
'                          displayed.
'
' Output: Text to the screen.
'*********************************************************************

' Clear the screen, set colors, and hide the cursor.

WIDTH 80, 25
COLOR Normal, BackGround
CLS
LOCATE , , CursorOff

' Display the titles and menu choices on the screen.

READ NumLines
FOR Row = 1 TO NumLines
        LOCATE (Row + 3), 2
        READ Tmp$
        PRINT Tmp$
NEXT Row

' Display the border.

LOCATE 3, 3: PRINT CHR$(201); STRING$(74, 205); CHR$(187)
FOR Counter = 4 TO 24
        LOCATE Counter, 3: PRINT CHR$(186);
        LOCATE Counter, 78: PRINT CHR$(186)
NEXT Counter
LOCATE 24, 3: PRINT CHR$(200); STRING$(74, 205); CHR$(188);

' Display the current values of the input and output file names if needed.

IF SeeFiles = True THEN
        COLOR HiLite, BackGround
        LOCATE FirstPlace, 32
        PRINT Inputfile
        LOCATE SecondPlace, 32
        PRINT OutputFile
        COLOR Normal, BackGround
END IF
END SUB
```

```
FUNCTION Filter$ (Text$, FilterString$)
'****************************************************************************
' This function will take unwanted characters out of a string by comparing
' them with a filter string containing only acceptable characters.  The
' function will return the new string with the rightmost blank spaces
' removed following the filtering.
'
' Parameters: Text$          - The original string to filter.
'             FilterString$  - The acceptable characters.
'
' Output: A string with only acceptable characters in it.
'****************************************************************************

LET Temp$ = ""
LET TxtLength = LEN(Text$)

' Isolate each character in the string.

FOR Increm = 1 TO TxtLength
        LET Character$ = MID$(Text$, Increm, 1)

        ' If the character is in the filter string then save it.

        IF INSTR(FilterString$, Character$) <> 0 THEN
                LET Temp$ = Temp$ + Character$
        END IF
NEXT Increm
LET Filter$ = RTRIM$(Temp$)
END FUNCTION
```

```
FUNCTION FixTransponder$ (Counter, SatNum, DualFlag)
'*****************************************************************
' This function will return the correct transponder number based on several
' factors.  First, if the satellite is a C-Band or Ku-Band bird (no hybrids)
' the transponder number count is okay and the transponder number is returned.
' Second, if the satellite is a hybrid bird, and the processing has reached
' the C-Band/72 Mhz transponders, an "A" or "B" is appended to the number
' to indicate whether it's the 1st or 2nd sample.  Thirdly, a satellite is
' a hybrid and the processing has reached the second FCC sample in which
' case an "A" or "B" is appended to the transponder number.  Note that the
' actual transponder number used on the FCC data is calculated and not read
' from the input file.  This is one place where code may have to be modified
' if the FCC changes the format of their data diskette.
'
' Parameters: Counter  - Current count of transponders.
'             SatNum   - Current address place in valid names array.
'             DualFlag - Flagged true if this is the 2nd satellite sample.
'
' Output : The correct transponder number for the output file.
'*****************************************************************

' The value of this variable is preserved over calls to this function.
' It is used as a logical count of transponders for the double sample
' problems of the C-band/72 Mhz transponders.

STATIC TransponderCount

IF ("C" = RTRIM$(Satellites(SatNum).TypeOfBird) OR "K" =
        RTRIM$(Satellites(SatNum).TypeOfBird) OR ("H" =
        RTRIM$(Satellites(SatNum).TypeOfBird) AND Counter <=
        Satellites(SatNum).NumC36Band AND DualFlag = False)) THEN

        ' Satellite is a C-band/36, 72 Mhz or a Ku-band non-hybrid or
        ' hybrid bird.  The transponder numbers are okay in this instance so
        ' append only a blank space to the end of the number.  Add a 0 to the
        ' front if the number is a 1 thru 9.

        LET TempNum$ = LTRIM$(STR$(Counter))
        IF LEN(TempNum$) = 1 THEN LET TempNum$ = "0" + TempNum$
        LET TempNum$ = TempNum$ + " "
        LET TransponderCount = 0
```

```
ELSEIF ("H" = RTRIM$(Satellites(SatNum).TypeOfBird) AND Counter >
        Satellites(SatNum).NumC36Band) THEN

        ' Satellite is a C-band/36 & 72 Mhz and Ku-band hybrid bird.
        ' Do both of the C-band/72 Mhz logical transponders here as the FCC
        ' has double sampled them.  An "A" is appended to the first sample
        ' transponder number while a "B" is appended to the second sample.
        ' Note that this is the first sample of the satellite's transponders.

        IF TransponderCount = 0 THEN
                LET TransponderCount = Counter
        ELSEIF (Counter MOD 2 <> 0) THEN
                LET TransponderCount = TransponderCount + 1
        END IF
        LET TempNum$ = LTRIM$(STR$(TransponderCount))
        IF LEN(TempNum$) = 1 THEN LET TempNum$ = "0" + TempNum$
        IF (Counter MOD 2 <> 0) THEN
                LET TempNum$ = TempNum$ + "A"
        ELSE
                LET TempNum$ = TempNum$ + "B"
        END IF

ELSEIF ("H" = RTRIM$(Satellites(SatNum).TypeOfBird) AND Counter <=
        Satellites(SatNum).NumC36Band AND DualFlag = True) THEN

        ' Satellite is a C-band/36 & 72 Mhz and Ku-band hybrid bird.
        ' Do both of the Ku-band logical transponders here as the FCC
        ' has double sampled them.  An "A" is appended to the first sample
        ' transponder number while a "B" is appended to the second sample.
        ' Note that this is the second sample of the satellite's transponders.

        IF TransponderCount = 0 THEN
                LET TransponderCount = Counter
        ELSEIF (Counter MOD 2 <> 0) THEN
                LET TransponderCount = TransponderCount + 1
        END IF
        IF TransponderCount > Satellites(SatNum).NumKuBand THEN
                TransponderCount = 1
        LET TempNum$ = LTRIM$(STR$(TransponderCount))
        IF LEN(TempNum$) = 1 THEN LET TempNum$ = "0" + TempNum$
        IF (Counter MOD 2 <> 0) THEN
                LET TempNum$ = TempNum$ + "A"
        ELSE
                LET TempNum$ = TempNum$ + "B"
        END IF
END IF
LET FixTransponder$ = TempNum$
END FUNCTION
```

```
FUNCTION FoundSat (SatName$, SatNum$)
'*******************************************************************************
' This function will search through the dual sampling array for a particular
' satellite name and number.  If a satellite is found the function returns
' it's array address otherwise it returns a zero.
'
' Parameters: SatName$ - String containing the satellite name to find.
'             SatNum$  - String containing the satellite number to find.
'
' Output: The array address of the satellite or a zero.
'*******************************************************************************

LET Increm = 0
DO
        LET Increm = Increm + 1
LOOP UNTIL (RTRIM$(NumSats(Increm).NameSat) = SatName$ AND
            NumSats(Increm).NumOfSat = VAL(SatNum$)) OR Increm = MaxSats
IF Increm = MaxSats THEN LET FoundSat = 0 ELSE LET FoundSat = Increm
END FUNCTION
```

```
FUNCTION GetDate$ (DateText$)
'***********************************************************************
' This function will format a date string of the form MM/DD/YY into a new
' date string of the form DD-MMM-19YY.
'
' Parameters: DateText$ - The sting containing the original date string.
'
' Output: The new date string.
'***********************************************************************

LET Month = VAL(DateText$)
LET Day$ = LTRIM$(MID$(DateText$, (INSTR(DateText$, "/") + 1), 2))
LET Year$ = RIGHT$(DateText$, 2)

SELECT CASE Month
        CASE 1
                LET Month$ = "Jan"
        CASE 2
                LET Month$ = "Feb"
        CASE 3
                LET Month$ = "Mar"
        CASE 4
                LET Month$ = "Apr"
        CASE 5
                LET Month$ = "May"
        CASE 6
                LET Month$ = "Jun"
        CASE 7
                LET Month$ = "Jul"
        CASE 8
                LET Month$ = "Aug"
        CASE 9
                LET Month$ = "Sep"
        CASE 10
                LET Month$ = "Oct"
        CASE 11
                LET Month$ = "Nov"
        CASE 12
                LET Month$ = "Dec"
END SELECT
LET GetDate$ = Day$ + "-" + Month$ + "-19" + Year$
END FUNCTION
```

```
SUB GetFileName (RowNumber, FileString$, FileNum)
'*******************************************************************************
' This procedure will blank out the current file name position and then
' prompt the user to enter a file name.  It will then open the corresponding
' input or output file as long as the names are not equal.
'
' Parameters: RowNumber   - Number of the row where cursor should be.
'             FileString$ - File name to be entered by user.
'             FileNum     - File number to use when opening file.
'
' Output: File name entered by user and corresponding file number.
'*******************************************************************************

' Position the cursor and display it and set up error handler just in case.

ON ERROR GOTO Handler
LOCATE RowNumber, 32
PRINT STRING$(43, 32)
LOCATE RowNumber, 32, CursorOn
COLOR HiLite, BackGround
INPUT "", FileString$
LOCATE , , CursorOff
COLOR Normal, BackGround

' Make sure that the file names are not equal and open the right file.

IF (FileNum = 1 AND FileString$ = OutputFile) OR (FileNum = 2 AND FileString$
    = Inputfile) THEN
        Text$ =
        "        ERROR - You may not have the file names equal to one another."
        CALL Alarm
        CALL DisplayMessage(Text$)
ELSEIF FileNum = 1 THEN
        CLOSE #1
        OPEN FileString$ FOR INPUT AS #1
ELSE
        CLOSE #2
        OPEN FileString$ FOR OUTPUT AS #2
END IF
END SUB
```

```
SUB GetNames (Num, TotalSamples)
'*********************************************************************
' This function will open and read the valid satellite names, the type of
' satellite it is (C = C-band/36 Mhz, K = Ku-band, H = Hybrid), the number of
' times that the FCC samples the bird (hybrids = 2, all others = 1) the number
' of C-band 36 Mhz transponders, the number of C-band 72 Mhz transponder, and
' the number of Ku transponders from the satellite name data file.  This file
' is in ASCII format with each of the satellite information on a line by
' itself.  It will also count the number of good names and the number of
' total satellite samples and return these values.
'
' Parameters: NONE
'
' Output: The number of names in the satellite name data file. *********************************************************************

OPEN SatelliteNameFile$ FOR INPUT AS #3
LET Num = 0: LET TotalSamples = 0
WHILE NOT EOF(3)
        LET Num = Num + 1
        INPUT #3, Satellites(Num).SatName, Satellites(Num).SatNum,
            Satellites(Num).TypeOfBird, Satellites(Num).NumSampled,
            Satellites(Num).NumC36Band, Satellites(Num).NumC72Band,
            Satellites(Num).NumKuBand
        LET Satellites(Num).SatName = UCASE$(Satellites(Num).SatName)
        LET Satellites(Num).TypeOfBird = UCASE$(Satellites(Num).TypeOfBird)
        LET TotalSamples = TotalSamples + Satellites(Num).NumSampled
WEND
CLOSE #3
END SUB
```

```
SUB GetSatelliteInfo
'*****************************************************************************
' This procedure will read from the input file to extract the satellite
' information.  It will strip away the input file's titles and headers and
' then call the procedure to retrieve the usage hour and transponder usage
' percentages.  It will also call the SUB to check the satellite family name
' and number as well as add a satellite to the dual sample array.  This is
' one place that might have to be modified if the FCC changes the format of
' their data file.  The satellites SBS 1/2 or SBS 2 are changed to SBS 1
' in this SUB.
'
' Parameters: NONE
'
' Output: NONE
'*****************************************************************************


' Read a line from the input file and extract the satellite family name and
' satellite number.

LINE INPUT #1, NameLine$
LET NameLine$ = UCASE$(NameLine$)
LET SatelliteName$ = MID$(NameLine$, (INSTR(NameLine$, NameString$) + 6), 15)
LET SatelliteFamily$ = Filter$(SatelliteName$, " ABCDEFGHIJKLMNOPQRSTUVWXYZ")
LET SatelliteNumber$ = Filter$(SatelliteName$, "0123456789")
PRINT #4,
PRINT #4, "BEGIN CONVERTING - The "; RTRIM$(LTRIM$(SatelliteName$));
          " satellite is being operated on."

' Check the satellite name and number for validity and return correct ones.

CALL CheckSatellite(SatelliteFamily$, SatelliteNumber$, SatelliteName$)

' Change Satellite SBS 1/2 to SBS 1 as a family name and print to report.
' The number may also be given as 2 which should also be changed to 1.

IF SatelliteFamily$ = "SBS" AND SatelliteNumber$ = "12" THEN
        LET SatelliteNumber$ = "1"
        PRINT #4, TAB(7);
        "CORRECTION - The SBS 1/2 satellite number has been changed to SBS 1."
END IF

IF SatelliteFamily$ = "SBS" AND SatelliteNumber$ = "2" THEN
        LET SatelliteNumber$ = "1"
        PRINT #4, TAB(7);
        "CORRECTION - The SBS 2 satellite number has been changed to SBS 1."
END IF


' Add valid satellite name and number to sampling array if it isn't already.
' This insures that the transponder percentages are numbered correctly.

LET WhereFound = FoundSat(SatelliteFamily$, SatelliteNumber$)
IF WhereFound > 0 THEN
        LET NumSats(WhereFound).Done = True
ELSE
        LET NumberSats = NumberSats + 1
        LET NumSats(NumberSats).NameSat = SatelliteFamily$
        LET NumSats(NumberSats).NumOfSat = VAL(SatelliteNumber$)
        LET NumSats(NumberSats).Done = False
END IF
```

```
' Read a line from the input file and extract the transponder sampling start
' date and start time.

LINE INPUT #1, DateLine$
LET DateLine$ = UCASE$(DateLine$)
LET StartDate$ = MID$(DateLine$, (INSTR(DateLine$, DateString$) + 6), 8)
LET StartDate$ = GetDate$(StartDate$)
LET StartTime$ = MID$(DateLine$, (INSTR(DateLine$, TimeString$) + 6), 5)
LET StartTime$ = GetTime$(StartTime$)

' Continue reading from the input file until the constant HourString$ is
' found to strip off titles and headers.

DO UNTIL (INSTR(OneLine$, HourString$) > 0 AND LEN(OneLine$) < 10) OR EOF(1)
        LINE INPUT #1, OneLine$
        LET OneLine$ = UCASE$(OneLine$)
LOOP

' Retrieve the sampling hour and transponder usage percentages then print a
' message when done with a particular satellite.

CALL GetTransponders(SatelliteFamily$, SatelliteNumber$, StartDate$,
                     StartTime$)
PRINT #4, "DONE  CONVERTING - The "; SatelliteFamily$; " "; SatelliteNumber$;
          " satellite has been successfully converted."
END SUB
```

```
FUNCTION GetTime$ (TimeText$)
'*******************************************************************
' This function will format a 24 hour time string of the form HHMM into a new
' 24 hour time string of the form HH:MM:SS.  The seconds are assumed to be
' zeros.
'
' Parameters: TimeText$ - The sting containing the original time string.
'
' Output: The new time string.
'*******************************************************************

LET HourTime$ = LTRIM$(LEFT$(TimeText$, 2))
IF VAL(HourTime$) = 0 THEN LET HourTime$ = "0"
LET MinuteTime$ = LTRIM$(RIGHT$(RTRIM$(TimeText$), 2))
IF VAL(MinuteTime$) = 0 THEN LET MinuteTime$ = "0"
LET GetTime$ = HourTime$ + ":" + MinuteTime$ + ":0"
END FUNCTION
```

```
SUB GetTransponders (SatelliteFamily$, SatelliteNumber$, StartDate$,
                     StartTime$)
'*****************************************************************************
' This procedure will read through the rows containing the transponder usage
' percentage samples and their corresponding sampling hours.  It will
' retrieve the start time from the file, calculate the transponder number,
' and retrieve the transponder usage percentage.  This SUB will also check
' the number of transponders processed to verify that the data matches the
' satellite's actual physical transponders.  It will then write all pertinent
' data to the output file, including the parameters passed to it.  Print
' a message to the report if the program fails due to bad data or if the
' satellite sample contains less than a 24 hour transponder sampling.  This
' is one place where code may have to be modified if the FCC changes the
' format of their data diskette.
'
' Parameters: SatelliteFamily$ - String containing satellite family name.
'             SatelliteNumber$ - String containing satellite number.
'             StartDate$       - String containing sampling start date.
'             StartTime$       - String containing sampling start time.
'
' Output: The satellite family name, number, transponder number, start date,
'         start time, hour, and percent used to the output file.
'*****************************************************************************

' Get the corresponding valid satellite information to help process the
' transponders.  Transponders are either C-band/36 Mhz, C-band/72 Mhz, or
' Ku-band.  Alert the user and terminate the program if the FCC has provided
' a satellite which is not in the file SATNAMES.DAT.

LET SatNum = 0
DO
        LET SatNum = SatNum + 1
LOOP UNTIL (SatelliteFamily$ = RTRIM$(Satellites(SatNum).SatName) AND
        VAL(SatelliteNumber$) = Satellites(SatNum).SatNum) OR SatNum =
        MaxSats
IF SatNum = MaxSats THEN
        CALL Alarm
        LOCATE 21, 6, CursorOn
        COLOR HiLite, BackGround
        PRINT "CATASTROPHIC ERROR - The "; RTRIM$(LTRIM$(SatelliteFamily$));
            " "; RTRIM$(LTRIM$(SatelliteNumber$));
            " satellite isn't in SATNAMES.DAT.";
        LOCATE 22, 21
        PRINT "*** Processing has been terminated ***"
        PRINT #4, TAB(7); "CATASTROPHIC ERROR - The ";
            RTRIM$(LTRIM$(SatelliteFamily$)); " ";
            RTRIM$(LTRIM$(SatelliteNumber$));
            " satellite isn't in SATNAMES.DAT."
        PRINT #4,
        PRINT #4, TAB(21); "*** Processing has been terminated ***"
        CLOSE
        CALL Alarm
        END
END IF
```

```
' Check to see if satellite has a dual sample and flag if so.

WhereFound = FoundSat(SatelliteFamily$, SatelliteNumber$)
IF WhereFound > 0 AND NumSats(WhereFound).Done = True THEN
        LET DualFlag = True
ELSE
        LET DualFlag = False
END IF

' Read from the input file until the constant PeriodString$ is found and
' initialize the sample hour counter.

LET CountHours = 24
LINE INPUT #1, TransponderLine$
LET TransponderLine$ = UCASE$(TransponderLine$)
DO UNTIL INSTR(TransponderLine$, PeriodString$) > 0 OR EOF(1)

        ' Format the transponder sampling data if there is some.

        IF LEN(TransponderLine$) <> 0 THEN

                ' Start times are actual from input file and decrement hour
                ' sample counter.

                LET CountHours = CountHours - 1
                LET StarttimeNum$ = LTRIM$(MID$(TransponderLine$, 2, 2))
                LET Transponders$ = MID$(TransponderLine$, 6,
                                        (LEN(TransponderLine$) - 5))

                ' Loop thru the row of transponder usage percentages and
                ' write to the output file the complete row.  Transponder
                ' numbers are calculated and are not actual so they may be
                ' modified by the function FixTransponder$.

                LET Counter = 0
                FOR CountTransponders = 1 TO LEN(Transponders$) STEP 4
                        LET Counter = Counter + 1
                        LET TransponderNum$ = FixTransponder$(Counter, SatNum,
                                        DualFlag)
                        LET Percent$ = LTRIM$(MID$(Transponders$,
                                        CountTransponders, 4))
                        PRINT #2, SatelliteFamily$ + "," + SatelliteNumber$ +
                                "," + TransponderNum$ + "," + StartDate$ + " " +
                                StartTime$ + "," + StarttimeNum$ + "," +
                                Percent$
                NEXT CountTransponders
```

73

```
' Check that the number of transponders checked is correct
' for that particular satellite.  If not, abort the program
' as the FCC has supplied incorrect usage data.

IF (DualFlag = False AND Counter <>
        (Satellites(SatNum).NumC36Band +
        Satellites(SatNum).NumC72Band +
        Satellites(SatNum).NumKuBand)) OR (DualFlag = True AND
        Counter <> (2 * Satellites(SatNum).NumKuBand)) THEN
        CALL Alarm
        LOCATE 21, 7, CursorOn
        COLOR HiLite, BackGround
        PRINT "CATASTROPHIC ERROR - The ";
                RTRIM$(LTRIM$(SatelliteFamily$)); " ";
                RTRIM$(LTRIM$(SatelliteNumber$));
                " transponder data is incorrect.";
        LOCATE 22, 21
        PRINT "*** Processing has been terminated ***"
        PRINT #4, TAB(7); "CATASTROPHIC ERROR - The ";
                RTRIM$(LTRIM$(SatelliteFamily$)); " ";
                RTRIM$(LTRIM$(SatelliteNumber$));
                " transponder data is incorrect."
        PRINT #4,
        PRINT #4, TAB(21);
                "*** Processing has been terminated ***"
        CLOSE
        CALL Alarm
        END
    END IF
END IF

' Read another line from the input file.

LINE INPUT #1, TransponderLine$
LET TransponderLine$ = UCASE$(TransponderLine$)
LOOP

' Print a warning to the output file if a satellite has less than a 24 sample.

IF CountHours > 0 THEN PRINT #4, TAB(7);
        "WARNING     - This satellite's transponder sample is missing";
        CountHours; "hour(s)."
END SUB
```

```
SUB LookForString (PageString$)
'*********************************************************************
' This procedure will read from the input file until the constant PageString$
' is found indicating a new set of satellite transponder usage data.  This
' is used to distinguish between satellites.
'
' Parameters: PageString$ - String constant indicating new satellite.
'
' Output: NONE
'*********************************************************************

DO UNTIL INSTR(OneLine$, PageString$) > 0 OR EOF(1)
        LINE INPUT #1, OneLine$
        LET OneLine$ = UCASE$(OneLine$)
LOOP
END SUB
```

```
SUB PrintHeader
'*******************************************************************************
' This procedure is used to print a title and headers to the printer as the
' beginning of the statistics report.  The date, time, input file name, and
' output file name is included for future reference.
'
' Parameters: NONE
'
' Output: The appropriate title and headers to the printer.
'*******************************************************************************

WIDTH #4, 80                          ' Set width of printer to 80 columns.
PRINT #4, : PRINT #4, : PRINT #4, : PRINT #4,
PRINT #4, TAB(12); "* * * N A S A   G S O S T A T S   D A T A B A S E   * * *"
PRINT #4, : PRINT #4,
PRINT #4, TAB(15); "FCC Transponder Usage Data File Conversion Program"
PRINT #4,

'Convert the 24-hour clock used by TIME$ to 12-hour output followed by
' "AM" or "PM".

LET TimeNow$ = TIME$
LET Hour = VAL(TimeNow$)
IF Hour < 12 THEN
        LET AmPm$ = " AM."
ELSE
        LET AmPm$ = " PM."
        LET Hour = Hour - 12
END IF
PRINT #4, TAB(4); "Data file conversion statistics processed on "; DATE$;
                  " at "; STR$(Hour); LTRIM$(RIGHT$(TimeNow$, 6)); AmPm$
PRINT #4,
PRINT #4, TAB(4); "Original input file Name:   "; UCASE$(Inputfile)
PRINT #4,
PRINT #4, TAB(4); "Converted output file Name:   "; UCASE$(OutputFile)
PRINT #4,
PRINT #4, TAB(14); "+++ INFORMATION ON INDIVIDUAL SATELLITES FOLLOWS +++"
END SUB
```

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE January 1993	3. REPORT TYPE AND DATES COVERED Final Contractor Report

4. TITLE AND SUBTITLE
Federal Communications Commission (FCC) Transponder Loading Data
Conversion Software
User's Guide and Software Maintenance Manual
Version 1.2

5. FUNDING NUMBERS

WU-144-10-10
NAS3-25776

6. AUTHOR(S)

Paul G. Mallasch

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Analex Corporation
3001 Aerospace Parkway
Brook Park, Ohio 44142

8. PERFORMING ORGANIZATION REPORT NUMBER

E-7570

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR-191067

11. SUPPLEMENTARY NOTES

Project Manager, James E. Hollansworth, Space Electronics Division, NASA Lewis Research Center, (216) 433-3458.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 61

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This volume contains the complete software system documentation for the Federal Communications Commission (FCC) Transponder Loading Data Conversion Software (FIX-FCC). This software was written to facilitate the formatting and conversion of FCC Transponder Occupancy (Loading) Data before it is loaded into the NASA Geosynchronous Satellite Orbital Statistics Database System (GSOSTATS). The information that FCC supplies NASA is in report form and must be converted into a form readable by the database management software used in the GSOSTATS application. Both the User's Guide and Software Maintenance Manual are contained in this document. This volume of documentation passed an independent quality assurance review and certification by the Product Assurance and Security Office of the Planning Research Corporation (PRC). The manuals were reviewed for format, content, and readability. The Software Management and Assurance Program (SMAP) life cycle and documentation standards were used in the development of this document. Accordingly, these standards were used in the review. Refer to the System/Software Test/Product Assurance Report for the Geosynchronous Satellite Orbital Statistics Database System (GSOSTATS) for additional information.

14. SUBJECT TERMS
FCC; GSOSTATS database; Transponder loading

15. NUMBER OF PAGES
78

16. PRICE CODE
A05

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	