PNL-10414 Ver. 2.1
UC-620

Integrated Baseline System (IBS)
Version 2.1
Data Management Guide

CHEMICAL STOCKPILE
EMERGENCY
PREPAREDNESS
PROGRAM
LOCAL STATE FEDERAL

January 1995

Prepared for the
U.S. Army Nuclear and Chemical Agency
under a Related Services Agreement
with the U.S. Department of Energy
Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Operated for the U.S. Department of Energy
by Battelle Memorial Institute

ACKNOWLEDGMENT

This software and its documentation were produced with Government support under Contract Number DE-AC06-76RLO 1830 awarded by the United States Department of Energy. The Government retains a paid-up non-exclusive, irrevocable worldwide license to reproduce, prepare derivative works, perform publicly and display publicly by or for the Government, including the right to distribute to other Government contractors.

PNL-10414
Ver. 2.1
UC-620

INTEGRATED BASELINE SYSTEM (IBS)
Version 2.1

DATA MANAGEMENT GUIDE

J. C. Bower [a]
M. J. Burford
T. R. Downing
M. C. Moïse
J. R. Williams

January 1995

Prepared for the
U.S. Army Nuclear and Chemical Agency
under a Related Services Agreement
with the U.S. Department of Energy
Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

(a) Bower Software Services
Kennewick, Washington

NOTICE

The Integrated Baseline System (IBS) is an emergency planning and training tool intended to supplement existing resources for emergency planning, training, and management. IBS results must NOT be used as the sole basis for a decision regarding public safety. Although IBS is operational, not all IBS software/models have been verified. The assumptions of the IBS software should be reviewed to ensure its appropriateness for an intended use.

Preface

The Integrated Baseline System (IBS) is an emergency management planning and analysis tool being developed under the direction of the Federal Emergency Management Agency. The following documents were developed to support system users. The audience for each is identified.

This ***IBS Data Management Guide*** provides the information needed to manage the data files and database used to support the administrative, user-environment, database management, and operational capabilities of the IBS. **Audience:** chiefly database administrators and system managers, but also emergency management planners and analysts who want to know details of the emergency management data.

The ***IBS User Guide*** explains how to start and use the ***IBS program***, which is designed to help civilian emergency management personnel to plan for and support their responses to a chemical-releasing *event* at a military chemical stockpile.[a] **Audience:** all users of the IBS, especially emergency management planners and analysts.

The ***IBS Utilities Guide*** explains how you can use IBS utility programs to manage and manipulate various kinds of IBS data. These programs include utilities for creating, editing, and displaying maps and other data that are referenced to geographic location. **Audience:** chiefly data managers, but also system managers and some emergency management planners and analysts.

The ***IBS Models Guide*** summarizes the IBS use of several computer models for predicting the results of emergency situations. These include models for predicting dispersion/doses of airborne contaminants, traffic evacuation, explosion effects, heat radiation from a fire, and siren sound transmission. The guide references additional technical documentation on the models when such documentation is available from other sources. **Audience:** chiefly emergency management planners and analysts, but also data managers and system managers.

The ***IBS System Management Guide*** defines IBS hardware and software requirements and gives instructions for installing, upgrading, or transferring the IBS software package. **Audience:** system managers.

(a) The IBS program was developed as part of the U.S. Army's Chemical Stockpile Emergency Preparedness Program (CSEPP).

About This Guide

Purpose

The Integrated Baseline System (IBS) is an emergency management planning and analysis tool that is being developed under the direction of the U.S. Army Nuclear and Chemical Agency (USANCA)[a]. The ***IBS Data Management Guide*** provides the background, as well as the operations and procedures needed to generate and maintain a site-specific map database.

Scope

Data and system managers use this guide to manage the data files and database that support the administrative, user-environment, database management, and operational capabilities of the IBS. This document provides a description of the data files and structures necessary for running the IBS software and using the site map database.

To ensure that data and system managers have the background to successfully manage the data system, this guide includes a description of the functions supported by the IBS data management system. Also included are descriptions of the IBS data directory structure, the IBS map database, and IBS functional databases. When special utility programs are used during database management, this guide refers you to the ***IBS Utilities Guide*** for specific instructions.

Although the IBS manual set describes basic and advanced operations, it is **not** a complete reference set. Emergency situation modeling software in the IBS is supported by additional technical documents. Some of the IBS software is commercial software for which more complete documentation is available. Where necessary, the IBS manuals reference such documentation.

Direct general questions regarding IBS to the IBS Project Manager for the USANCA:

(a) The IBS is being developed by the Pacific Northwest Laboratory (PNL). PNL is operated for the U.S. Department of Energy by Battelle Memorial Institute under Contract DE-AC06-76RLO 1830.

COL Lawrence Skelly
Chief, Chemical Division
U.S. Army Nuclear and Chemical Agency
7150 Heller Loop, Suite 101
Springfield, VA 22150-3198

Address software use questions and problems to the IBS Help Desk operated by Innovative Emergency Management:

(504) 767-1138
Innovative Emergency Management
7423 Picardy Ave., Suite E
Baton Rouge, LA 70808-4362

All references to IBS documents should be cleared through USANCA. If an emergency contact is necessary, and USANCA cannot be reached, a second source of information is the software developer, Pacific Northwest Laboratory:

Pacific Northwest Laboratory
Attn: Blanche M. Wood
POB 999, MS K7-22
Richland, WA 99352
Telephone (509) 375-2615

Audience

This guide is addressed chiefly to data managers and system managers. It will also be of value to emergency management planners and analysts who want to know details of the emergency management data. Users of this guide are expected to be familiar with the ***IBS User Guide***, and with the IBS itself.

Prerequisites

Users of the IBS software should receive training in the use of IBS capabilities for emergency management planning and operational tasks associated with Chemical Stockpile Emergency Preparedness Program (CSEPP). This document is a reference guide for IBS database administrators (DBAs) and advanced users, describing the IBS files and their structures.

Organization

This guide includes the following four sections, three appendixes, and an index:

1. Introduction

An introduction to the guide.

2. Logical Components of a Map Database

A description of what a Geographic Information System (GIS) does and how the IBS GIS works.

3. IBS Data Architecture

A description of the data files and hierarchy used to support the operational capabilities of IBS.

4. Procedures

This section contains a detailed description of how to perform certain tasks critical to making IBS available to the users. The tasks are described as follows:

- Generating a Site-Specific Map Database - The procedure for designing and setting up a site-specific map database with appropriate topographies.
- Creating User-Defined Icons - The procedure for creating and accessing those icons not supplied with the software.

Appendix A. IBS File Descriptions

IBS file descriptions that include directory, purpose, access, and structure descriptions and specifications.

Appendix B. IBS Icon Dictionary

A list of the attribute codes, names, graphic file names, and types of graphic icons used in the IBS.

Appendix C. DMS and TXT File Formats

IBS descriptions of the Digital Mapping System (DMS) and text (TXT) files.

How to Use This Guide

This guide provides general and specific guidance on managing the data and databases in the IBS. It is divided into two parts, Sections 1 through 4 and Appendixes A, B, and C. Reading the sections will provide a significant amount of information in a reasonable amount of time. The appendixes and index are provided for reference.

For the Data Manager or Systems Manager

By reading Section 2, *Logical Components of a Map Database*, data managers and system managers can become familiar with how the IBS Geographic Information System (GIS) works.

Section 3, *IBS Data Architecture*, describes the purposes of the data files in the IBS system and how they interrelate.

Section 4, *Procedures*, provides details describing how to perform certain complex tasks necessary for IBS operation, specifically creating the map database and site-specific icons.

Visual Conventions

You will find the following conventions in this guide.

Type Style	Meaning
bold	Text that you must type at the keyboard. Example: **NEWUSER** at the $ prompt.
italic	Placeholder for information that you must provide. Example: **DEL *filename*** Here you type the actual name of a file, instead of the word shown in italics.
ALL CAPITALS	Directory names, filenames, program names, and acronyms.

The pathway through menus and submenus is depicted by an arrow symbol (->). IBS -> MESSAGE BOARD -> Create IBS -> ONPOST REPORT indicates that the MESSAGE BOARD submenu is selected from the IBS Main Menu, and that Create IBS -> ONPOST REPORT is selected from the MESSAGE BOARD submenu.

Abbreviations and Acronyms

APA	Accident Planning Area
ASCII	American Standard Code for Information Interchange
CHEMS	Hazardous Chemical Modeling
CSEPP	Chemical Stockpile Emergency Preparedness Program
DBA	Database Administrator
DBMS	database management system
DCL	Digital Command Language
DEM	Digital Elevation Model
DLG	Digital Line Graph
DMS	Digital Mapping System
EMIS	Emergency Management Information System
EOC	Emergency Operations Center
EPZ	Emergency Planning Zone
FEMA	Federal Emergency Management Agency
FGEF	Federal Geographic Exchange Format
FTP	File Transfer Protocol
GIS	Geographic Information System
IBS	Integrated Baseline System
ICD	Interface Control Document
IP	Implementing Procedure
MCE	Maximum Credible Event
MESORAD	Meso-Scale Radiation Model
OSPM	Outdoor Sound Propagation Model
PNL	Pacific Northwest Laboratory
RDBMS	Relational Database Management System
TAC	Terrain Analysis Cell
UIC	User Identification Code
USANCA	U.S. Army Nuclear and Chemical Agency
USGS	United States Geological Survey
UTM	Universal Transverse Mercator

File Suffix List

*.ATT	Ancillary Attribute file for a DLG file
*.BCK	Backup file
*.BIN	Binary file
*.BIS	Batch Input Stream
*.BIX	ASCII version of a BIN file
*.COM	VMS DCL command file
*.DAT	Data file
*.DBG(-OFF)	Debug switch
*.DIR	VMS Directory
*.DLG	Digital Line Graph
*.DMS	Digital Mapping System file (spatial data)
*.DMX	ASCII format of a DMS file
*.DOC	ASCII text used as documents to explain an aspect of the data. Not used by IBS.
*.FIL	Map database control file
*.FMT	Import format file
*.HLT(-OFF)	Switch to enable/disable process
*.LIB	VMS object file library
*.MAI	Mail
*.TAA	Terrain Analysis Attribute (raster data)
*.TAC	Terrain Analysis Cell (raster header data)
*.TMP	Temporary file
*.TXT	Text file used to attach text to points on the map

Contents

Figure

Tables

Section 1
Introduction

The Integrated Baseline System (IBS) was developed by Pacific Northwest Laboratory (PNL) under the direction of the U.S. Army Nuclear and Chemical Agency (USANCA). IBS provides tools to civilian emergency management personnel for use in developing emergency plans and in supporting emergency management activities to cope with a chemical-release event at a military chemical stockpile. During the planning stages, emergency management personnel are able to simulate the effects of an event using the most accurate topographic data available. During the operation stage, IBS exchanges real-time information with the system onpost (EMIS, in this case). In each of these phases, IBS is highly data-dependent.

IBS was developed without the use of any commercial database system. Information and data are conveyed through IBS using binary and ASCII files, file position within the hierarchy, system logicals, and other means. The role of the IBS database administrator (DBA) is to ensure that data necessary to run the IBS software are available and properly installed. The DBA must be prepared to support all of the data aspects of IBS. The intent of this document is to supply the necessary information for the DBAs to accomplish their tasks.

This document is organized as a reference guide. Before reading this document, the user must be familiar with the terminology and use of the IBS system. The user is also expected to be familiar with the VAX/VMS operating system and FORTRAN format statements.

Section 2
Logical Components of a Map Database

Because much of the data in the IBS system is geographic in nature, a Geographic Information System (GIS) has been designed into IBS. That code allows IBS to display maps with certain features, including the output of many of the models and simulations.

Logically, a map

- covers a geographic area
- contains one or more topographies over that area.

(A topography is a group of similar objects [such as roads, rivers, or buildings] displayed in geographic relationship to each other.)

IBS adds some additional constraints:

- a map may cover one or more geographic areas-of-operation or regions
- each area-of-operation can be displayed and accessed using a geographic cover map.

The map database in the IBS GIS is divided into several topographies. (For a list of topographies, refer to the table in Section 3.1.) In addition, a standardized attribute dictionary and a standardized icon dictionary are included.

The Attribute Dictionary provides a standardized method for constructing, assigning, and interpreting characteristic information throughout the map database and the geographic information system.

The Icon Dictionary provides a standardized method for constructing, assigning, and graphically displaying icons throughout the map database and the geographic information system. Icons are symbols that are displayed on the map.

Thus, a map database contains the following logical components:

- Map Database Control Files
- Cover Maps for the Regions in the Map
- Topographies
- Attribute Dictionary
- Icon Dictionary.

2.1 Topographies

A topography can be thought of as a transparent viewgraph with one set of map features (such as railroads) drawn on it. Alone, it offers limited information. However, one of the main functions of the GIS is to allow the user to display multiple topographies simultaneously and in relation to each other. Thus, a user can display railroads (RR), 100K railroads (RL), and 100K roads (RD) together. Two classes of topographies exist in the IBS databases: static and volatile.

Static topographies are not changed by anyone other than the IBS DBA. These topographies are generated by the DBA, placed within the map database, and made available to all users (who cannot modify them). More than 100 static topographies are available to the IBS users. Many of these topographies (or the data used to generate them) are public domain information available from various federal agencies. Other static topographies are site-specific. The information needed to generate the topography will be provided by the appropriate local agency. Although static topographies may experience minor changes throughout a year, they are not dynamic within the time span of an event.

Volatile topographies are modified by the user only through the IBS application software. To create or modify a volatile topography, the DBA must provide the user with the authorization to run that portion of the IBS software. Information stored within an IBS volatile topography can be accessed by other portions of the IBS software. If the IBS user copies the information from the volatile map database to other directories, the transferred files cannot be returned to the volatile database (only one working copy of an IBS volatile geographic topography is permitted). Volatile topographies are generated by the D2 and CHEMS IBS models (D2, EX, FI, VA, and WD), entered by IBS users (DA, RM, RP, and SR), or received from the post (EI, WD, and WI).

2.2 GIS Objects

The IBS GIS supports three types of objects: points, lines, and polygons. For a geographic feature to be represented in IBS, it must be shown as one of the three types of objects. The valid GIS objects and examples of the type of geographic information they represent are:

Points	Dose locations, hospitals, resource centers, population centers, wells, intersections
Lines	Roads, streams, railroads
Polygons	Dispersion tracks, administrative boundaries, emergency planning zones, risk areas, contaminated areas.

These objects may have attributes that characterize the objects. For example, a linear object could be characterized as a road; that is, class 1, two lanes, and Interstate 82. A point object could be characterized as a hospital, named Sacred Heart with 1000 beds, whose unique identifier is 12345, and which is linked to the hospital icon.

An icon is a special type of point. Typically, points are not displayed by IBS, or are displayed as an asterisk. However, if the point is an icon, IBS will look up its symbol in the Icon Dictionary, and display the symbol. This process enables the emergency planner to identify hospitals (displayed as an H), shelters (a pentagon), and police stations (a P in a circle).

GIS objects are located using their latitude and longitude. A point object has a single latitude and longitude. A line is a list of latitudes and longitudes (in the order in which they are connected). A polygon is a line where the two endpoints are the same. (A complex polygon is a polygon with *holes* in it, similar to the hole through a donut.)

Because all geographic objects are physically located on the earth, they exhibit spatial relationships, such as location, length, distance, area, and extent. IBS supports spatial relationships between geographic objects including points-within-polygon, line-intersection, polygon-intersection, inclusion, nearness, and other proximity functions.

2.3 GIS Data Files

The amount of data required by the GIS dramatically complicates the data management task. Although the number of files in the GIS may vary, 2000 files is not considered excessive. In an effort to manage such a large quantity of data, the following paragraphs present some guidelines.

The general guidelines for the construction of IBS files, established to facilitate the identification, generation, and management of files by the IBS software or the DBA, are:

- Each file has a unique filename.
- Files reside within a specified directory corresponding to the functional database that supports the associated IBS functionality.

Within the IBS GIS database are two general types of data files: those that contain geographic (or spatial) information, and those that do not. Spatial data files are used to store mapping and location-oriented information; non-spatial files are used to store tabular, textual, and other supporting information for emergency management.

Most spatial data is stored in special files call Digital Mapping System (DMS) files. These files consist of GIS objects, and contain virtually all the graphics that appear on the IBS system. The three categories of graphics in the IBS GIS are topographies, cover maps, and icon symbols. DMS files are used to store all of them. For each object displayed, the DMS file contains the attributes associated with it and the latitude and longitude of all the points displayed. Icon symbols are always displayed the same size (regardless of the zoom into the map); the latitude and longitude values for an icon symbol are the display coordinates.

Although most spatial data is stored using DMS files, one class of data (raster) is not. Raster data consists of the values for each element of a regularly spaced grid. The most common occurrence of raster data within IBS is the elevation topography (EL). Instead of a single DMS file, raster data is stored in two files, a *.TAC file containing the header data, and a *.TAA file containing the raster values. Data stored as raster are not identified as graphic objects and are not associated with either text or attributes.

Some non-spatial data files within the GIS are binary, and are not readable by the user. Usually a translation program exists to convert these files to and from ASCII. Examples of binary non-spatial GIS files are: limit files, label files, and the attribute dictionary.

The remaining non-spatial files are ASCII. Although the user seldom has reason to access them directly, at times it is advantageous to have human-readable files. Examples of ASCII non-spatial files are T.DAT, the icon dictionary, and the map database control files. In addition, all of the binary GIS files (spatial and non-spatial) can have ASCII versions.

Constraints, in addition to the general file guidelines, levied on ASCII files are:

- Wherever possible, records in the files contain 80 (or fewer) characters, the width of a standard computer screen.
- Where applicable, a header record identifies the type and description of the file.
- If applicable, the second record identifies the type of information or report the file is providing.
- Additional records are defined and documented in Appendix A.

Most GIS files and data are located within topographies. Physically, a topography is a subdirectory under the top-level map database directory. A topography subdirectory name must be two characters long and must be unique. As a minimum, the topography subdirectory must contain two files, a binary limit file and a DMS file. The binary limit file is named LIMITxx.BIN and contains a list of the DMS files and the area each covers, defined in latitude and longitude. Although the LIMITxx.BIN file is not ASCII, it can be converted to the LIMITxx.DAT ASCII file using the utility SEELIMITS. On initially creating the topography, LIMITxx.BIN can be created from a list of the DMS files (usually named T.DAT) using the CREATELIMITS utility. If all of the DMS files in a directory are to be included in the topography, LIMITxx.BIN can be created using the LIMITXX3 utility. (The LIMITXX3 utility is not documented at this time.)

Although the topography must contain at least one DMS file, it is not limited to only one. Especially if the amount of data in the topography is extensive, multiple DMS files may be necessary to provide coverage over mutually exclusive areas (like tiles covering a floor). One of the functions of the GIS is to reassemble the DMS data into a seamless graphics display.

The IBS GIS allows the user to label a graphics object. This process is typically used to add a name to the graphics object. The DMS file does not have the capability to store the textual data. When textual data is present, it is stored in a text file (*.TXT), a binary file not directly readable by the user.

If a user copies a data file from a volatile topography to another directory, the transferred file cannot be returned to the topography (that is, only one working copy of a volatile data file can be present in an IBS functional database). Returning the transferred file could cause inconsistencies in the database.

The map database cover maps are very similar to a normal DMS file in a topography. The principle difference is that instead of the LIMITxx.BIN file in a topography, the cover map uses the TOPOCTL.FIL file. Each region, or area-of-operation, must have only one DMS file listed in the TOPOCTL.FIL file. The TOPOCTL.FIL file contains a list of regions and a single cover map file for each region.

2.4 How the GIS Database Works

Nearly all geographic applications invoke some portion of the map database. Each application differs in the way an area-of-operations is established, how specific topographies are selected, and how the geographic area-of-interest is identified. However, after the topographies and an area-of-interest have been established, each GIS application accesses, reads, and displays geographic information in exactly the same manner. This section provides a general top-down description of how the map database is accessed. When appropriate, comments about database access for individual applications are presented.

2.4.1 Defining the Map Database Logicals

For each map database, the command file MAP$DATA$MAIN:SETUPDB.COM (activated by a *Map*DATABASE symbol) is used to assign top-level logicals for the map database. (Two examples of these symbols are UTDATABASE or ALDATABASE, as defined in IEMIS$SYSF:ASS_MAP_SYM.COM.) Usually, this procedure is initiated during system boot or by the DBA during database configuration to provide system logicals. Lower-level logicals are defined in terms of the top-level logicals in IEMIS$SYSF:ASS_MAP_LOG.COM.

2.4.2 Identifying a Map Database

The most general level of entry into a map database is through the topography control file, TOPOCTL.FIL. This file provides a list of geographic areas-of-operation and graphical cover maps. After the user selects an area-of-operation, the associated cover map is displayed and the user interactively establishes a geographic area-of-interest. This action is automatically performed when a user initiates the MPDISPLAY application. Because the IBS system is based on pre-established site databases for a pre-defined area-of-operations, this information is automatically sent to the map database management system, and the user is not able to select an area.

2.4.3 Identifying Map Layers or Topographies

Upon establishing a map database and an area-of-operation, the topography description file, TOPODESC.FIL, is listed to allow users to select the topographies (map layers) they wish to display. The topography description file identifies all topographies included in the map database.

The topography description file contains one header record and one record for each topography. Table 2.1 is an example of a topography description file for a small database. This database has 17 records (one record per line) for 17 topographies, plus an initial description record. Each topography record shown in the table consists of four columns: 1) a topography descriptor, 2) a topography type, 3) a logical name, and 4) a topographic code.

Table 2.1. Example of a Topography Description File

```
TOPODESC.FIL for my test database
ADMINISTRATIVE BOUNDARIES     MAP$DATA$AB:   AB
POLITICAL BOUNDARIES          MAP$DATA$PB:   PB
CONGRESSIONAL DISTRICTS       MAP$DATA$CD:   CD
CULTURAL FEATURES             MAP$DATA$CF:   CF
RAILROADS                     MAP$DATA$RR:   RR
ELECTRICAL POWER GRIDS        MAP$DATA$EP:   EP
ROADS & TRAILS                MAP$DATA$RT:   RT
STREAMS                       MAP$DATA$ST:   ST
WATER BODIES                  MAP$DATA$WB:   WB
GNIS (Geographic Names)       MAP$DATA$GN:   GN
POPULATED PLACE NAMES         MAP$DATA$PL:   PL
POPULATION                    MAP$DATA$PO:   PO
D2 TRACK                     VMAP$DATA$D2:   D2
FACILITIES & RESOURCES       VMAP$DATA$RM:   RM
RISK AREAS                   VMAP$DATA$RP:   RP
ONPOST D2 TRACK              VMAP$DATA$WD:   WD
ELEVATION                     MAP$DATA$EL:   EL
```

Topography descriptors are placed in the menu, allowing users to select the topographies to be accessed and displayed. The topography type is V for volatile topographies; and is blank for static topographies. The logical name identifies the directory where the individual DMS data files for that topography are stored. The 2-character topographic code is used to identify the limit file that lists the DMS files in the topography.

The MPDISPLAY application always requires the user to identify the topographies to be manipulated. The IBS application uses the default topographies identified in the job environment file (JOB_ENV.DAT). Both applications use the topography description file to identify the directory where the topography's files are stored. Both of these processes also use the topography description file to change the topographies being accessed and interactively displayed.

2.4.4 Accessing Spatial Objects from the Database

After the map layers and an area-of-interest have been established, the DMS files (identified by the topography limit file, LIMITxx.BIN) that intersect or fall within the area-of-interest are opened. Objects whose geographic extent fall within or intersect the area-of-interest are read into the GIS internal data structures.

This data access procedure is followed for each selected topography. On completion, the geographic, characteristic, and graphical display information for all topographic objects within the area-of-interest are directly accessible to the application with the capability provided by the GIS library functions.

2.4.5 Attributes and the Attribute Dictionary

Further description of spatial data objects (points, lines, and polygons) is accomplished through **attributes**. Attributes are numeric values used to describe data items or some characteristic of data items. Attribute codes consist of three numbers (separated by semicolons):

- a major attribute code (to indicate a major topographic group)
- a minor attribute code (to indicate a feature within the group)
- a parameter value (which <u>may</u> give data about the specific feature).

For instance, 1;560;300 indicates a major attribute code of 1, a minor attribute code of 560, and a parameter value of 300. Usually, attribute codes describe specific topographic features: the major code indicates a major topographic group, and the minor code indicates a specific feature within that group. The attribute 1;560;300, for example, specifies

Major code	1	= hydrography	(major topographic group)
Minor code	560	= permanent lake	(hydrographic feature)
Parameter value	300	= lake length in decameters (~10 yd)	(additional information)

For each map database, a special file called an attribute dictionary defines the meaning of each major:minor pair of attribute codes. See the *IBS Utilities Guide* for a listing of the standard IBS attribute dictionary. The ATTXASC utility reads the dictionary and produces a readable ASCII file.

For each defined pairing of major and minor codes, the attribute dictionary contains a text description of the defined topographic feature; for example:

Major	Minor	Param	Description
1	560	300	Lake or pond (Permanent) (decameters)

Notice the parameter value is defined in parentheses at the end of the text description. Attribute codes may be specified without a parameter value; some attribute codes do not even have an associated parameter value. For example, 1;500 indicates a major attribute code of 1 and a minor attribute code of 500:

Major	Minor	Param	Description
1	500	0	Aqueduct

Up to 15 attributes can be assigned to each object in a DMS file. In addition to the normal attributes that characterize the object, three types of special attributes are included. The first type of special attribute is used to define polygon fill characteristics. When the attribute major code is 9 and the minor code is 99, the parameter value defines the fill pattern and color of the polygon.

The second type of special attribute is a reference to an icon. If major and minor codes of the first attribute are listed in the icon dictionary, the GIS point will be identified as an icon. Icons are graphic symbols that can be associated with a particular data point location. They provide a consistent means of representing hospitals, airports, schools, or any topographic feature that occurs repeatedly.

The third type of special attribute is a pointer to a text label. If the major code of the first attribute is either 4 or 10, a label is identified for the object. The parameter value and the minor code of the attribute point to the location of the text label in the TXT file. The parameter value is equal to the record number in the TXT file, and the minor code is equal to the word (2-byte integer) within the specific record in the TXT file where the label is stored. Within the TXT file, the text label contains information including the text, font, rotation, and pen color.

Two types of labels are possible—those that can be displayed (major code equals 4) and those that cannot be displayed (major code equals 10). An example of a displayable label is the name of a road. The information available in a non-displayable label is not useful to the general IBS user. An example of a non-displayable label is a TIGER object ID that could provide a reference back to the same object within a relational database. Any data item can have text associated with it through its attributes. The name of the text data file is stored in the DMS file header, the initial portion of the DMS graphics file.

An icon or label will only be displayed if it is the first attribute of an object. This condition implies an object cannot have an icon and a label at once. IBS does not prevent an object from having multiple icons or labels, but only the first one is displayed. A label can consist of up to 5 lines of text and 255 characters.

This method of managing attributes has been standardized throughout the GIS and its applications. The standardized list of attributes is provided as an appendix to the *IBS Utilities Guide*.

2.4.6 Icons and the Icon Control File

Icons are graphic symbols used to graphically depict the location and availability of features such as facilities, resources, services, and physical landmarks. Each icon is represented by a symbol which can be drawn on the topographic map display. For example, an airplane symbol may be drawn on the topographic map to depict the location of an airport. Icons always appear at a constant viewing size, regardless of the map scale.

For each map database, a special file called an icon control file (or icon dictionary) is used for the proper display of icon symbols. The icon control file contains the following information for each icon listed within the file:

- Major attribute code of the icon
- Minor attribute code of the icon
- Icon DMS file name (used to graphically draw the icon)
- Icon category (see following paragraph)
- Ring position (for display of resource icons).

The icon control file is essentially a list of the icons that are available for display. If the major and minor attribute codes of the first attribute of a point are listed in the icon control file, the program will identify the point as an icon, and display the icon graphically.

Furthermore, the IBS program divides icons into three categories:

1. Facility Icons - represent the location of facilities such as schools, churches or food distribution centers. These icons are identified with an F in the icon category of the icon control file.

2. Resource Icons - depict a major resource type, such as communication equipment, food, and medical equipment. These icons are identified with an R in the icon category of the icon control file.

3. Non-Facility/Resource Icons - represent items that are not within the facility or resource categories. Examples of non-facility/resource icons are the icons for a city, flood, or earthquake. The icon category for such icons is blank.

Facility, resource, and non-facility/resource icons are treated differently in some of the functions of the IBS program, such as within Resource Management. Resource icons are displayed in concentric rings around the facility icon to which they are assigned.

Section 3
IBS Data Architecture

This section identifies the IBS data files and databases used to support the administrative, user-environment, database management, and operational capabilities of IBS. This section will identify the basic structure of the geographic, ASCII, binary, and command files contained within the IBS data management system. It will outline the directories used to store the individual data management files, plus list the name, type, and purpose of the files within each directory. Finally, this section will provide the list of IBS functional databases within the data management system, as well as the name and physical location of their associated data files. File formats (for selected files) are provided in Appendix A.

3.1 IBS Data Directory Structure

The IBS directory structure is a fairly extensive tree of five directories: [INFOMANAGER], [ONSITE], [OFFSITE], [ISYSFILES], and [*Map*].

Figure 3.1 presents the IBS data management structure as graphical tree diagrams. Included are the directories, subdirectories, and all the files within the directories used by the IBS applications. Table 3.1, the directories used in IBS data management, and Table 3.2, representative topographies, support Figure 3.1.

The INFOMANAGER directory contains information about the different sites, the implementing procedures developed for each one, and the information necessary to run the different models available under IBS. This directory structure is the most complex within IBS. The ONSITE directory is used as a depository for incoming information from the onpost system, and the OFFSITE directory contains miscellaneous shared data. The ISYSFILES directory contains the administrative and general database files, as well as template files. Finally, the [*Map*] directory contains static geographic map information.

To better understand the relationships between these different sections, a directory hierarchy and a brief description of each directory are provided (Figure 3.1 and Table 3.1). This information can be used as a map of the system and its components.

[ISYSFILES]

IEMIS$SYSF:
See ISYSFILES
Directory
Structure
(Page 6 of 6)

[INFOMANAGER]

INFO$DIR:
*.DIS
EVENT.LOG
JOB_ENV.DAT
LOGIN.COM
MAIL_C.LOG
SITE.DAT

[.SITES]

[.xxxx][1]

SITE$DIR:
xxxxEL.TAA
xxxxEL.TAC
xxxxELAT.TAA
xxxxELAT.TAC
xxxxEL_UTM.TAA
xxxxEL_UTM.TAC
CASEINDEX.DAT
CASEINDEX.DOC
CASEINDEX.LST
CCS_TRACK.DAT
POPCODES.DAT
SETUP.COM

[.BCK][4]

SITE$BCK:
CHEMS_timestamp.BCK
CHEMS_NDX_timestamp.BCK
D2_timestamp.BCK
D2_DMS_timestamp.BCK
D2_NDX_timestamp.BCK
FROM_timestamp.BCK
IDYNEV_timestamp.BCK
IDYNEV_NDX_timestamp.BCK
IP_timestamp.BCK
IP_NDX_timestamp.BCK
MESORAD_timestamp.BCK
MESORAD_NDX_timestamp.BCK
OFFSITE_timestamp.BCK
OSPM_timestamp.BCK
OSPM_NDX_timestamp.BCK
RA_timestamp.BCK
RA_DMS_timestamp.BCK
RM_timestamp.BCK
RM_DMS_timestamp.BCK
SITE_timestamp.BCK
SEND_timestamp.BCK

[.INPUT]

See INPUT
Directory
Structure
(Pages 2 and 3 of 6)

[.OUTPUT]

See OUTPUT
Directory
Structure
(Page 2 of 6)

[OFFSITE]

OFFSITE$DIR:
See OFFSITE
Directory
Structure
(Page 4 of 6)

[ONSITE]

ONSITE$DIR:
generic.ATT[2]
generic.DAT[2]
generic.DLG[2]
BASE_POP.DAT
CALLLIST.DAT
CAS_SUM.DAT
D2INPnnn.DAT[3]
D2LOGnnn.DAT[3]
DOSAGnnn.DAT[3]
ENVIRON.DAT
KEY.DAT
KNOWNPTS.DAT
MLUPDATE.DAT
NOTIFY.DAT
SIGEVENT.DAT
SLICEnnn.DAT[3]
TOWERSEL.DAT
WEATHER.DAT
WORKPLAN.DAT
XFERLIST.DAT

[*Map*]

MAP$DATA$MAIN:
See Map
Directory
Structure
(Page 5 of 6)

1 The notation **xxxx** indicates a 4-character site name.
2 The notation **generic** indicates the name of a generic data file. Names of generic data files are listed in the XFERLIST.DAT file. At the time of this writing, IBS will accept any filename for a generic data file; however, EMIS always uses the name GENERIC.DAT, and does not send the .DLG or .ATT files.
3 The notation **nnn** indicates a model scenario case number between 000 and 999.
4 The notation **timestamp** indicates a 14-character time stamp with the format YYMMDDHHMMSSHH.

Figure 3.1. IBS Data Management Structure (Page 1 of 6)

[.INPUT]

hierarchy continued

[.CHEMS]

SITEICHEMS:
CnnnCHEMS.GEN
CRASHFLG.DAT

[.D2]

SITEID2:
CnnnD2INP.DAT
D2_MASTER_FILE.DAT

[.DYNEV]

SITEIDYNEV:
CnnnEVAC.DMS
CnnnEVACPARAM.DAT
CnnnINP.DAT
CnnnLNK.DAT
EVAC_SEARCH_HEADERS.DAT

[.MESORAD]

SITEIMESORAD:
CnnnCHK.DAT
CnnnOBS.DAT
CnnnPOL.DAT
CnnnRSP.DAT
CnnnSnm.DAT[1]
CnnnSTA.DAT
CnnnTER.DAT
CnnnTOP.DAT
CnnnV.DAT

[.OSPM]

SITEIOSPM:
CnnnOSPM.DAT
CnnnOSPM.DMS

[.OUTPUT]

[.CHEMS]

SITEOCHEMS:
Temporary files only

[.D2]

SITEOD2:
CnnnD2LOG.DAT
CnnnDOSAGE.DAT
CnnnPLOT.DAT
CnnnPOSINP.DAT
CnnnSCRATCH.DAT
CnnnSCRIPT.DAT
CnnnSLICE.DAT
CnnnSTEP.DAT

[.DYNEV]

SITEODYNEV:
CnnnBIN.U36
CnnnCOD.RPT
CnnnCOD.U37
CnnnLNK.DAT
CnnnNOD.DAT
CnnnSUM.RPT
CnnnTV.DAT
NODE_POS.DAT
NODE_POS.SRT

[.MESORAD][2]

SITEOMESORAD:
CnnnAIRDPL.typ
CnnnAIRNDP.typ
CnnnCONTOUR.DMS
CnnnDEPOST.typ
CnnnEXTGRD.typ
CnnnEXTSIC.typ
CnnnFLD.RPT
CnnnGBRADV.BIN
CnnnGDRADV.BIN
CnnnING.RPT
CnnnINHLNG.typ
CnnnINHTHY.typ
CnnnINHWBD.typ
CnnnIODADV.BIN
CnnnLOG.RPT
CnnnPRM.RPT
CnnnPUFFTR.BIN
CnnnSTATIN.BIN
CnnnTOTWBD.typ
CnnnWINDFD.BIN
CnnnXmmDAT[3]

[.OSPM]

SITEOOSPM:
CnnnCONTOUR.DMS
CnnnDBA.DAT
CnnnDBC.DAT
CnnnOCTm.DAT[4]

The notation nnn indicates a model scenario case number between 000 and 999.
1 The notation nm indicates n releases (1 thru 6) of m sources (1 thru 4).
2 The notation typ indicated a file suffix of both ADV and CUM.
3 The notation mm indicates a checkpoint number between 1 and 50.
4 The notation m indicates the octave bands between 1 and 8 (64Hz to 8KHz).

Figure 3.1. IBS Data Management Structure (Page 2 of 6)

[.INPUT]

continued

- [.OFF]
  - **SITE$OFF:**
    - CR_RISK_AREA.DAT
    - CR_RISK_AREA.DMS
    - LOCATE_PLA.TMP
    - LOCATE_PLU.TMP
    - P_T_UPDATE.DAT
    - PEOPLE_TRACK.DAT
- [.MAPS]
  - **SITE$MAPDIR:**
  - [.D2]
    - **MAP$DATA$D2:**
      - CnnnD2.TXT
      - CnnnPLUME.DMS
      - LIMITD2.BIN[2]
  - [.EX]
    - **MAP$DATA$EX:**
      - CnnnEXP.DMS
      - LIMITEX.BIN[2]
  - [.FI]
    - **MAP$DATA$FI:**
      - CnnnFI.DMS
      - LIMITFI.BIN[2]
  - [.VA]
    - **MAP$DATA$VA:**
      - CnnnVA.DMS
      - LIMITVA.BIN[2]
- [.IP]
  - **SITE$IP:**
    - CnnnACTIONS.DAT
    - CnnnACTION_NOTES.DAT
    - CnnnMASTER_CHECKLIST.DAT
    - CnnnMASTER_NOTES.DAT
    - CnnnRESOURCE_CHECKLIST.DAT
    - CnnnRESOURCE_NOTES.DAT
    - CnnnTASK_LINKS.DAT
    - CnnnTASK_NOTES.DAT
    - SCEN_INDEX.DAT
    - SCEN_TABLE.DAT
- [.POP]
  - **SITE$POP:**
    - pppp.DAT[1]

The notation **nnn** indicates a model scenario case number between 000 and 999.

1 Population files are referenced in [INFOMANAGER.SITES.xxxx]POPCODES.DAT.

2 The LIMITxx.BIN file references the currently selected model case.

Figure 3.1. IBS Data Management Structure (Page 3 of 6)

[OFFSITE]

OFFSITE$DIR:

[.xxxx]

OFF$MAPDIR:

[.SEND] [.FROM] [.xx][3]

OFF$DIR:
AGENCY_LOOKUP.DAT
COMBINED.LOG
EVR_route.DMS[1]
FACILITY.DAT
FACILITY_TRACK.DAT
IMPMAIL.LOG
KNOWN_POINTS.DAT
KNOWN_POLYGONS.DAT
KNOWN_ROUTES.DAT
PERSONNEL.DAT
POSITION_LOOKUP.DAT
RESOURCE.DAT
SITE_EF.DAT
SPECIAL_POPULATION.DAT
SPOCC_ACTIVITY.LOG
SPOCC_ACTIVITY.LOG_DETAIL
SPOCC_EVENT.LOG
SPOCC_EVENT.LOG_DETAIL
WATCH_EVENT.LOG
WATCH_EVENT.LOG_DETAIL
WORK_PLAN.DAT

OFF$SENDDIR:
generic.ATT[2]
generic.DAT[2]
generic.DLG[2]
D2INPnnn.DAT
D2LOGnnn.DAT
DOSAGnnn.DAT
KEY.DAT
MLUPDATE.DAT
OFF_CAS.DAT
OFF_EVR.DAT
OFF_KP.ATT
OFF_KP.DAT
OFF_KP.DLG
OFF_KP.DMS
OFF_KPL.DAT
OFF_PA.DAT
OFF_PEVR.DAT
OFF_PIMR.DAT
OFF_RCD.DAT
OFF_RCR.DAT
OFF_SHLT.DAT
OFF_STAT.DAT
OFF_TC.DAT
SLICEnnn.DAT
XFERLIST.DAT

OFF$FROMDIR:
Contents of this subdirectory are the same as the [ONSITE] directory (Page 1 of 6)

MAP$DATA$xx:
Contents of this subdirectory are the same as the [*Map*.xx] subdirectory (Page 5 of 6)

The notation **xxxx** indicates a 4-character site name. The notation **nnn** indicates a model scenario case number between 000 and 999.
1 The notation **route** indicates a evacuation route name. The evacuation route names are listed in the KNOWN_ROUTES.DAT file.
2 The notation **generic** indicates the name of a generic data file. Generic file names are listed in the OFF$SENDDIR:XFERLIST.DAT file. At the time of this writing, IBS will accept any name for a generic data file; however, EMIS requires the name GENERIC.DAT, and ignores .DLG and .ATT files.
3 The notation **xx** indicates the 2-character shared-volatile topography names (DA, EI, RM, RP, SR, WD, and WI). See the Input directory structure, pages 2 and 3 of 6, for the private volatile topographies, and the Map directory, page 5 of 6, for the non-volatile topographies.

Figure 3.1. IBS Data Management Structure (Page 4 of 6)

[*Map*]

MAP$DATA$MAIN:
SETUPDB.COM
TOPOCTL.FIL[1]
TOPODESC.FIL[2]
*.DMS[3]

[.xx][4]

MAP$DATA$xx:
*.DMS[5]
*.TXT[6]
LIMITxx.BIN[7]
T.DAT

[.EL][8]

MAP$DATA$EL:
*.TAA
*.TAC
LIMITEL.BIN
T.DAT

[.ATT]

MAP$ATTRIB:
ATTRIBDIC.BIN[9]
ATTRIBDIC.BIX

[.ICON]

MAP$ICONS:
ICON.CTL[10]
*.DMS[11]

[.LOG]

MAP$MEMLOG:
MEM_LOG.DAT

1 TOPOCTL.FIL can be accessed with the **MAP$CONTROL:** logical.
2 TOPODESC.FIL can be accessed with the **MAP$TOPOGRAPHY:** logical.
3 DMS files in the map directory are cover maps of regions. Cover maps are listed in the TOPOCTL.FIL file.
4 The notation **xx** signifies multiple 2-character topography names. The topographies are listed in TOPODESC.FIL and may include, but are not limited to, all of the non-volatile topographies listed in Table 3.2. See the Input directory structure, pages 2 and 3 of 6, for the private volatile topographies, and the Offsite directory, page 4 of 6, for the Shared volatile topographies.
5 The DMS files are listed in the LIMITxx.BIN file.
6 The TXT files are present only if text labels were added to an item.
7 It is common to see other LIMITxx files in a topography subdirectory. They are output files from various IBS utilities, are not required to use IBS, and can be deleted. Some examples are:

LIMITxx.DAT	ASCII version of .BIN, created by SEELIMITS
LIMITxx.STAT	ASCII report, created by STATDMS
LIMITxx.FRQ	ASCII report, created by ATTFREQ
LIMITxx.DMS	coverage map, created by SHOWCOVER.

8 The elevation topography contains raster data in the TAC and TAA files, instead of vector data in a DMS file. The filename prefixes of the TAC and TAA files must be the same and are listed in the LIMITEL.BIN file.
9 ATTRIBDIC.BIN can be accessed with the **MAP$ATTRIBUTE$DICTIONARY:** logical.
10 ICON.CTL can be accessed with the **MAP$ICON$CTL:** logical.
11 The icon names, types, and the name of the DMS files used to draw the icons are listed in the ICON.CTL file.

Figure 3.1. IBS Data Management Structure (Page 5 of 6)

[ISYSFILES]

IEMIS$SYSF:

System
- ALLSITE.DAT
- COLORTBL.DAT
- MAILLIST.DAT
- MAIL_SYSTEMS.DAT
- MASTERDIR.DAT
- OFF_TO_ON.DAT
- OTHER_SYSTEMS.DAT
- PERMITI.DAT
- PERMITI.DFT
- POST_SYSTEM.DAT

Setup
- ASS_DISK_LOG.COM
- ASS_IBS_LOG.COM
- ASS_IBS_SYM.COM
- ASS_MAP_LOG.COM
- ASS_MAP_SYM.COM
- ASS_SITE_LOG.COM
- PNLBATCH.COM
- PNLEXECUT.COM
- PNLINSTALL.COM
- PNLLOGDEF.COM
- PNLSYMDEF.COM

Switches[1]
- EVENT.HLT-OFF
- METSIM.HLT-OFF

Debug Switches[2]
- EVENT.DBG
- FIXKPL.DBG
- MAIL_C.DBG
- MAPDBGEN.DBG
- MAPDBUPDT.DBG

Templates
- BLANKICON.DMS
- CASEINDEX.DAT_TEMPLATE
- FTP.COM_TCPWARE
- FTP.COM_TEMPLATE
- FTP.COM_WINTCP
- JOB_ENV.DAT_TEMPLATE
- LOGIN.COM_TEMPLATE
- POPCODES.DAT_TEMPLATE
- SETUP.COM_TEMPLATE
- SETUPDB.COM_TEMPLATE
- SITE.DAT_TEMPLATE
- SITE_EF.DAT_TEMPLATE
- TOPODESC.FIL_TEMPLATE

Import Format Files
- AG.FMT
- FAC.FMT
- KP.FMT
- PER.FMT
- POS.FMT
- RES.FMT

Miscellaneous
- COLORTEST.DMS
- DLGXDMS.ATT
- PROBLEM.RPT
- RPTSUBJ.DAT
- SIREN.LIB
- SITE.DAT
- UPDATE.DOC

Sample programs
- FTPCHECK.COM
- FTPSTART.COM
- FTPSTOP.COM
- SELECTDB.COM

Chems Model
- C000CHEMS.GEN
- CRASHFLG.DAT

D2 Model
- C000D2INP.DAT

IDYNEV Model
- C000EVACPARAM.DAT
- C000INP.DAT

OSPM Model
- C000OSPM.DAT

MESORAD Model
- C000CHK.DAT
- C000OBS.DAT
- C000POL.DAT
- C000RSP.DAT
- C000Snm.DAT[3]
- C000STA.DAT
- C000TER.DAT
- C000TOP.DAT
- C000V.DAT

D2 Igloo Files
- IGL_AAD.DAT
- IGL_DPG.DAT
- IGL_EWA.DAT
- IGL_LBG.DAT
- IGL_NAP.DAT
- IGL_PAD.DAT
- IGL_PBA.DAT
- IGL_UAD.DAT

[.SIM]

IEMIS$SIMF:

Contents of this sub-directory are the same as the [ONSITE] directory (Page 1 of 6)

[.nn][4]

- generic.ATT[5]
- generic.DAT[5]
- generic.DLG[5]
- BASE_POP.DAT
- CALLLIST.DAT
- CAS_SUM.DAT
- D2INP001.DAT
- D2LOG001.DAT
- DOSAG001.DAT
- ENVIRON.DAT
- KEY.DAT
- KNOWNPTS.DAT
- MLUPDATE.DAT
- NOTIFY.DAT
- SIGEVENT.DAT
- SLICE001.DAT
- TOWERSEL.DAT
- WEATHER.DAT
- WORKPLAN.DAT
- XFERLIST.DAT

[.nn][4]

- C000CHEMS.GEN
- C000D2INP.DAT
- C000INP.DAT
- C000OSPM.DAT

[.ATT]

Contents of this sub-directory are the same as the [*Map*.ATT] sub-directory (Page 5 of 6)

[.ICON]

Contents of this sub-directory are the same as the [*Map*.ICON] sub-directory (Page 5 of 6)

[.RELEASE]

Miscellaneous files used for new version releases

1 Allowable file states are *.HLT and *.HLT-OFF
2 Allowable file states are *.DBG and *.DBG-OFF
3 The notation **nm** signifies n releases (1 thru 6) of m sources (1 thru 4).
4 The notation **nn** signifies the 2-character state code.
5 The notation **generic** indicates the name of a generic data file. Names of generic data files are listed in the XFERLIST.DAT file. At the time of this writing, IBS will accept any file name for a generic data file; however, EMIS always uses the name GENERIC.DAT.

Figure 3.1. IBS Data Management Structure (Page 6 of 6)

Table 3.1. List and Description of Directories

Directory	Description
EVNDB:[INFOMANAGER]	**INFO$DIR:** Contains some operational data for all the sites. At the top level, it contains user environment and control files. It also contains mail distribution lists and other user-specific information. This directory contains a SITES directory that, in turn, contains information for each specific site.
EVNDB:[INFOMANAGER.SITES]	This directory is used only as a branching point to other sites. No files, other than subdirectories for the different sites, are found here.
EVNDB:[INFOMANAGER.SITES.xxxx]	**SITE$DIR:** The notation [.xxxx] signifies that many different directories are possible here. Each directory must have a 4-letter name (such as, BPNL) and contain information about a specific site. The types of files found at this level are site control files and model data files, such as elevation. Also found at this level are three subdirectories: [.INPUT], [.OUTPUT] and [.BCK].
EVNDB:[INFOMANAGER.SITES.xxxx.BCK]	**SITE$BCK:** Contains files that are archives of the current site. The file names are the date on which the data was archived, together with the data type.
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT]	Contains only directory files, as described following.
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT.IP]	**SITE$IP:** Contains the implementing procedures. The files contain the actions or tasks performed in response to a CSEPP event and its specific conditions. A scenario table linking model data to the IPs also resides here.
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT.OFF]	**SITE$OFF:** Contains Current Risk Area files and People Locate files.
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT.POP]	**SITE$POP:** Contains data associating raw population data with evacuation centroids of areas and sites.
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT.CHEMS]	**SITEICHEMS:** Contains input data for the three chemical models known as CHEMS (fire, vapor, and explosion).
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT.D2]	**SITEID2:** Contains input files for the D2 dispersion model.

Table 3.1 (contd)

Directory	Description
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT.DYNEV]	**SITEIDYNEV:** Contains input files for the IDYNEV evacuation model.
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT.MESORAD]	**SITEIMESORAD:** Contains input files for the radiological MESORAD model.
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT.OSPM]	**SITEIOSPM:** Contains input files for the Outdoor Sound Propagation Model (OSPM).
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT.MAPS]	**SITE$MAPDIR:** Includes subdirectories that contain volatile GIS topographies whose information is created and modified from within the IBS application. These topographies are associated with IBS models.
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT.MAPS.D2]	**MAP$DATA$D2:** Contains volatile GIS data files created by the D2 dispersion model. These files consist of dispersion tracks.
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT.MAPS.EX]	**MAP$DATA$EX:** Contains volatile GIS data files generated by the CHEMS application. These files consist of explosion contours.
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT.MAPS.FI]	**MAP$DATA$FI:** Contains volatile GIS data files created by the CHEMS application. These files consist of fire contours.
EVNDB:[INFOMANAGER.SITES.xxxx.INPUT.MAPS.VA]	**MAP$DATA$VA:** Contains volatile GIS data files created by the CHEMS application. These files contain vapor dispersion contours.
EVNDB:[INFOMANAGER.SITES.xxxx.OUTPUT]	Contains directories for storing the output from the different IBS models.
EVNDB:[INFOMANAGER.SITES.xxxx.OUTPUT.CHEMS]	**SITEOCHEMS:** Contains intermediate results from the chemical models known as CHEMS.
EVNDB:[INFOMANAGER.SITES.xxxx.OUTPUT.D2]	**SITEOD2:** Contains results from the D2 dispersion model.
EVNDB:[INFOMANAGER.SITES.xxxx.OUTPUT.DYNEV]	**SITEODYNEV:** Contains results from IDYNEV evacuation model.

Table 3.1 (contd)

Directory	Description
EVNDB:[INFOMANAGER.SITES.xxxx.OUTPUT.MESORAD]	**SITEOMESORAD:** Contains results from the radiological MESORAD model.
EVNDB:[INFOMANAGER.SITES.xxxx.OUTPUT.OSPM]	**SITEOOSPM:** Contains results from the Outdoor Sound Propogation Model (OSPM).
EVNDB:[ONSITE]	**ONSITE$DIR:** Contains information received from onpost or from the Onpost Simulator program.
EVNDB:[OFFSITE]	**OFFSITE$DIR:** Contains site directories allowing data sharing between users.
EVNDB:[OFFSITE.xxxx]	**OFF$DIR:** Contains journal files, facility/resource data, and personnel data.
EVNDB:[OFFSITE.xxxx.FROM]	**OFF$FROMDIR:** Contains files that have been sent from onpost or from the Onpost Simulator program.
EVNDB:[OFFSITE.xxxx.SEND]	**OFF$SENDDIR:** Contains files that have been sent from IBS to onpost.
EVNDB:[OFFSITE.xxxx.DA]	**MAP$DATA$DA:** Contains a volatile GIS data file created by the Damage Assessment application. This file consists of national defense boundaries and contaminated areas.
EVNDB:[OFFSITE.xxxx.EI]	**MAP$DATA$EI:** Contains a volatile GIS data file generated by the Event program based on the information sent from onpost. This file consists of environment icons.
EVNDB:[OFFSITE.xxxx.RM]	**MAP$DATA$RM:** Contains a volatile GIS data file created by the Resource Management application. This file consists of facility and resource icons.
EVNDB:[OFFSITE.xxxx.RP]	**MAP$DATA$RP:** Contains a volatile GIS data file created by the Risk Area Analysis application. This file consists of risk polygons.

Table 3.1 (contd)

Directory	Description
EVNDB:[OFFSITE.xxxx.SR]	**MAP$DATA$SR** Contains a volatile GIS data file created by the Search and Rescue application. This file consists of Search and Rescue areas.
EVNDB:[OFFSITE.xxxx.WD]	**MAP$DATA$WD:** Contains a volatile GIS data file created by the Event program based on information sent from onpost for the D2 dispersion model. This file consists of dispersion tracks.
EVNDB:[OFFSITE.xxxx.WI]	**MAP$DATA$WI:** Contains a volatile GIS data file created by the Event program based on information sent from onpost. This file consists of weather icons.
PNLDEVDB:[ISYSFILES]	**IEMIS$SYSF:** A general potpourri of files. First, it contains system administration files. Second, it contains ASCII data files that do not belong in a site database or modeling directory. Third, it contains template sites used by the Onpost Simulator Program. Finally, it contains the majority of the template files used by the DBA.
*Map*DB:[*Map*]	**MAP$DATA$MAIN:** Contains the subdirectories for the spatial data, attribute and icon dictionaries, and control files used to store, access, and manage the map database.
*Map*DB:[*Map*.ATT]	**MAP$ATTRIB:** Contains the attribute dictionary. This information is used to categorize data (for example, a line can have associated attributes indicating it is a road or a river).
*Map*DB:[*Map*.ICON]	**MAP$ICONS:** Contains the icon control file and the icon DMS files.
*Map*DB:[*Map*.LOG]	**MAP$MEMLOG:** Contains a statistical report generated by some programs with such information as the number of icons used in maps. This data can be used by the DBA to tailor the IBS system to make better use of the computer on which it is running.
*Map*DB:[*Map*.xx]	**MAP$DATA$xx:** Contain topographic data files. The directories shown in Table 3.2 represent topographies that may be present in the IBS system. Individual sites could have some or all, in addition to others not listed; for instance, *Map*DB:[*Map*.EL], accessed by logical MAP$DATA$EL, contains cell elevation data in grid format.

Table 3.2. Codes and Names of Representative Topographies

Topographies:

AB	ADMINISTRATIVE BOUNDARY
AG	AGRICULTURE
BM	BASE MAP
CD	CONGRESSIONAL DISTRICTS
CF	CULTURAL FEATURES
CN	RECOVERY AREAS
CT	CENSUS TRACT/ED
EC	ELEVATION CONTOUR
EL	ELEVATION
EP	ELECTRICAL POWER GRID
EZ	EMERGENCY PLANNING ZONE
FQ	FAULT LINES
FT	FEATURES (Icons)
GN	GNIS (Geographic Names)
HD	HYDROLOGY
HG	100K HYDROGRAPHY
HY	HYPSOGRAPHY
HZ	HAZMAT SITES
LS	LIVESTOCK
LU	LAND USES
MT	100K MISC TRANSPORT
NA	NAPB TARGETS
NF	NEW FEATURES (Icons)
NG	PIPE LINES
NP	NUC POWER PLANTS
PB	POLITICAL BOUNDARIES
PL	POPULATED PLACE NAMES
PO	POPULATION
R5	500K ROADS
RD	100K ROADS
RL	100K RAILROADS
RR	RAILROADS
RT	ROADS
SC	APA SECTORS
ST	STREAMS
TC	TRAFFIC CONTROL
TP	TRACT POPULATION
WB	WATER BODIES
WE	WELLS
WL	WILDLIFE
WS	WATERSHEDS

Private Volatile Topographies:

D2	D2 TRACK
EX	EXPLOSION CONTOURS
FI	FIRE CONTOURS
VA	VAPOR CONTOURS

Shared Volatile Topographies:

DA	DAMAGE ASSESSMENT AREAS
EI	ENVIRONMENT ICONS
RM	FACILITIES & RESOURCES
RP	RISK AREAS
SR	SEARCH/RESCUE AREAS
WD	ONPOST D2 TRACK
WI	WEATHER ICONS

TIGER Boundary Topographies:

BB	BOUNDARY	- BLOCK
CB	BOUNDARY	- COUNTY
LB	BOUNDARY	- PLACE
TB	BOUNDARY	- TRACT
VB	BOUNDARY	- SUBDIVISION

TIGER Line Topographies:

SN	STREAM NAMES
TA	ADDRESSES
TF	LANDMARKS
TM	MISC TRANSPORTATION
TN	STREET NAMES
TR	RAILROADS
TS	STREAMS
TT	ROADS
TW	WATER BODIES
WN	WATER BODY NAMES

TIGER Census Topographies:

B1	POPULATION	- BLOCK
B2	DEMOGRAPHIC	- BLOCK
B3	HOUSE SIZE	- BLOCK
B4	SPECIAL POP	- BLOCK
B5	HOUSING	- BLOCK
B6	HOUSE STRUCT	- BLOCK
B7	HOUSE VALUE	- BLOCK
B8	HOUSE TYPE	- BLOCK
C1	POPULATION	- COUNTY
C2	DEMOGRAPHIC	- COUNTY
C3	HOUSE SIZE	- COUNTY
C4	SPECIAL POP	- COUNTY
C5	HOUSING	- COUNTY
C6	HOUSE STRUCT	- COUNTY
C7	HOUSE VALUE	- COUNTY
C8	HOUSE TYPE	- COUNTY
G1	POPULATION	- GROUP
G2	DEMOGRAPHIC	- GROUP
G3	HOUSE SIZE	- GROUP
G4	SPECIAL POP	- GROUP
G5	HOUSING	- GROUP
G6	HOUSE STRUCT	- GROUP
G7	HOUSE VALUE	- GROUP
G8	HOUSE TYPE	- GROUP
P1	POPULATION	- PLACE
P2	DEMOGRAPHIC	- PLACE
P3	HOUSE SIZE	- PLACE
P4	SPECIAL POP	- PLACE
P5	HOUSING	- PLACE
P6	HOUSE STRUCT	- PLACE
P7	HOUSE VALUE	- PLACE
P8	HOUSE TYPE	- PLACE
S1	POPULATION	- STATE
S2	DEMOGRAPHIC	- STATE
S3	HOUSE SIZE	- STATE
S4	SPECIAL POP	- STATE
S5	HOUSING	- STATE
S6	HOUSE STRUCT	- STATE
S7	HOUSE VALUE	- STATE
S8	HOUSE TYPE	- STATE
T1	POPULATION	- TRACT
T2	DEMOGRAPHIC	- TRACT
T3	HOUSE SIZE	- TRACT
T4	SPECIAL POP	- TRACT
T5	HOUSING	- TRACT
T6	HOUSE STRUCT	- TRACT
T7	HOUSE VALUE	- TRACT
T8	HOUSE TYPE	- TRACT
V1	POPULATION	- SUBDIV
V2	DEMOGRAPHIC	- SUBDIV
V3	HOUSE SIZE	- SUBDIV
V4	SPECIAL POP	- SUBDIV
V5	HOUSING	- SUBDIV
V6	HOUSE STRUCT	- SUBDIV
V7	HOUSE VALUE	- SUBDIV
V8	HOUSE TYPE	- SUBDIV

The order, from largest to smallest, of TIGER topographies is State, County, Subdivision, Place, Track, Group, and Block.

3.1.1 Map Database [*Map*]

The map database is a single, well-defined source of geographic information that is available to all IBS users. The database is generated by consolidating geographic information from various national, state, and local sources. Selected information from each source is converted to individual geographic or spatial data objects and then placed into a common topography. Each spatial data object within a topography contains geographic location, characteristic (attribute), and graphical display information that can be directly accessed, manipulated, and displayed by the GIS. By selecting a desired topography and identifying a specific area-of-interest, you can directly access, in a single, well-defined manner, geographic information acquired from many sources.

Section 4 presents the steps a DBA takes to generate the site-specific IBS map database. A map database consists of special support files and data files distributed within various subdirectories of an overall database directory. Components of a map database are shown in the [*Map*] directory in Figure 3.1. They are described in Table 3.3. Specific file names are shown in uppercase letters; variable parts of file names and of directory names are shown in lowercase letters.

Table 3.3. Components of a Map Database in the [*Map*] Directory (Figure 3.1)

[*Map*]	<u>Map Database Directory</u> - Main (top-level) directory of the database (identified by the logical MAP$DATA$MAIN:).
*.DMS	<u>Cover Map File</u> - Is a high-level map of some region of the map database and is used by the MPDISPLAY program to allow the user to select an area-of-interest. This file is referenced from the Topography Control File.
SETUPDB.COM	<u>Setup Control File</u> - Assigns logical names so the files of the map database can be accessed by IBS software. It is invoked with the *Map*DATABASE or SYS*Map*DATABASE symbol.
TOPOCTL.FIL	<u>Topography Control File</u> - Contains the number and names of DMS file cover maps that depict broad geographic areas within the database. It is identified by the logical MAP$CONTROL:.
TOPODESC.FIL	<u>Topography Description File</u> - Lists the different topographies and the database directories where the topographic data reside. It is identified by the logical MAP$TOPOGRAPHY:.
[*Map*.xx]	<u>Topography Subdirectory</u> - Contains all the topographic data for one of the topographies listed in the topography control file. The name of the directory is a 2-letter **topographic code** assigned to that topography (see Table 3.2). Each topography is uniquely identified by a logical MAP$DATA$xx: where xx is the 2-character topography code.
*.DMS	<u>Topography Data File</u> - Binary file of the display symbols of the icons.

Table 3.3 (contd)

*.TXT	Topography Text File - Binary file containing textual data (for example, street or river names). Graphic objects in DMS files can reference these strings. More than one DMS file can reference the same TXT file. This file is necessary only if text strings are referenced.
LIMITxx.BIN	Topography Limit File - Specifies the names and geographic extent of all the DMS or TAC files that make up the topography in that subdirectory.
LIMITxx.DAT	ASCII Topography Limit File - Optional; an ASCII version of the .BIN file generated by the SEELIMITS utility.
T.DAT	Topography File List File - Optional; an ASCII file that specifies the names of all the DMS files that make up the topography in that subdirectory. It is created by the LIMITXX utility. The CREATLIMITS utility uses it to create the LIMITxx.BIN file.
[*Map*.ATT]	Attribute Dictionary Directory - Contains the GIS attribute dictionary. It is identified by the logical MAP$ATTRIB:.
ATTRIBDIC.BIN	Attribute Dictionary - Binary file containing the interpretations for each attribute that can be stored in a DMS data file. It is identified by the logical MAP$ATTRIBUTE$DICTIONARY:.
ATTRIBDIC.BIX	ASCII Attribute Dictionary - Optional; an ASCII version of the binary attribute dictionary that can be edited or printed. It is created by ATTXASC and used by ASCXATT.
[*Map*.EL]	Elevation Topography - Unique topography in that the data is not stored using DMS format. Instead, data is stored as data values in a raster (TAC, TAA) format. It is identified by the logical MAP$DATA$EL:. See [*Map*.xx] for a description of the other files which may be present in the EL topography.
*.TAA	Elevation Raster Data File - Stores the elevation data as elements. Elevation is stored as 2-byte integer values.
*.TAC	Elevation Raster Data Header File - Stores the elevation header information used to identify the geographic area, the size of the grids, and the compression method.
[*Map*.ICON]	Icon Dictionary Directory - Contains the GIS Icon dictionary. It is identified by the logical MAP$ICONS:.
*.DMS	Icon Symbol File - Graphic symbol that is displayed for the icon.
ICON.CTL	Icon Control File - ASCII file specifying information about each icon, including the attribute code, description, symbol file name, and icon type. It is identified by the logical MAP$ICON$CTL:.
[*Map*.LOG]	Log Subdirectory - Contains the GIS Usage Log file. It is identified by the logical MAP$MEMLOG:.
MEM_LOG.DAT	GIS Usage Log File - Used to store the IBS run-time memory usage statistics for later assessment by the DBA.

3.1.2 Site Database - [INFOMANAGER] + [user]

Site databases can be divided into two classes: planning and operational. Planning site databases are located under individual user accounts. Operational site databases are located under the Information Manager (INFOMANAGER) account, and only one is active (current) at a time. All IBS users can access the current operational site database. When they do, they are said to be in operational mode.

When not using the current operational site database, users (including the INFOMANAGER) are said to be in planning mode. The real difference between the current operational site database and all other site databases is that several users can use the current operational site database at once, in essence, sharing data. Changes made to the data by one user are immediately reflected in the data available to the other users in that mode. However, data in planning site databases or non-current operational site databases is not shared: only one user at a time (the user under whose account it is located) can use the data.

Planners require their own private site databases so that they may run model simulations and prepare implementing procedures without affecting data in operational site databases or other user site databases. Moreover, if a planner intends to modify topographic data, a private copy is necessary for the same reasons. After making changes to the data, the planner may, depending on privileges, copy the data to one of the operational site databases or copy the topographic data to the map database.

Operational sites are further subdivided into real or exercise sites (designated in INFO$DIR:SITE.DAT). A real site contains data that describes the current situation onpost. An exercise site is linked to a real site, but may contain fictitious data. Exercise sites are used in the simulation of an event, to test onpost and/or offpost Emergency Operation Centers.

A site can contain input to and output from the CHEMS, D2, IDYNEV, MESORAD, and OSPM models. A site also contains population data, risk area analysis data, people tracking data, implementing procedures, and private volatile topographies. During planning, analysts will develop scenarios, and run the models to predict the results. The pertinent scenarios, IPs, and other data will be copied from the planning site to an operational site to make it available to other users.

A site database consists of special support files and data files distributed within various subdirectories of an analyst (or INFOMANAGER) user directory. Components of a site database are shown in the INFOMANAGER directory in Figure 3.1. They are described in Table 3.4. Specific file names are shown in uppercase letters; variable parts of file names and of directory names are shown in lowercase letters.

Table 3.4. Physical Components of a User Home Directory

[user][a]	User Home Directory - Main (top-level) directory, an IBS user account, under which site databases are located. At the top level, it contains user environment and control files. It is identified by the logical INFO$DIR: that points to the INFOMANGER home directory, and SYS$LOGIN: that points to the user home directory.
*.DIS	Mail Distribution File - Mail distribution list. The file in INFOMANAGER is the system distribution list, the file in a users account is their personal distribution list.
EVENT.LOG	Event Log File - A communications log between IBS and the onpost system. Within IBS it can be viewed under STATUS BORAD -> VIEW BATCH LOG. It is created by the EVENT batch program and only exists in INFO$DIR:.
JOB_ENV.DAT	Job Environment File - Contains parameters that determine the user environment, (such as color, model numbers, or topographies). New users are given a default file that they can change within IBS.
LOGIN.COM	Login Command File - Executed automatically when a user logs onto the system. It sets system parameters (such as terminal type, protection, or editor). New IBS users are given a default file in which certain commands must not be changed or IBS does not run properly.
MAIL_C.LOG	Mail Check Log File - Log file for the Mail_C program.
SITE.DAT	Site Description File - Lists the different sites available in this user account. It includes such information as site name, description, type, or location. By default, normal IBS users have an empty site description file because they have no site databases.
[user.SITES]	Sites Subdirectory - Used as a branching point to site databases, and contains no files except the individual site subdirectories. The NEWSITE and DELSITE utilities are the primary means by which site databases are added to or deleted from a user account.
[user.SITES.xxxx]	Site Subdirectories - The notation [.xxxx] indicates several possible subdirectories, each having a 4-letter name (such as BPNL) designating a specific site. The currently selected site is identified by the logical SITE$DIR:.

(a) *user* is either the planner's/analyst's or the INFOMANAGER's home directory.

Table 3.4 (contd)

xxxxEL.TAA xxxxEL.TAC	OSPM Elevation Files - Required for OSPM operation only. Standard IBS elevation data set aside for the OSPM model.
xxxxELAT.TAA xxxxELAT.TAC	OSPM Attenuation Files - Required for OSPM operation only. Attenuation values (0 - soft ground, high attenuation; 1 - hard ground, low attenuation) used in the OSPM model.
xxxxEL_UTM.TAA xxxxEL_UTM.TAC	OSPM UTM Elevation Files - Required for OSPM operation only. UTM projection of elevation for use by the OSPM model.
CASEINDEX.DAT	Model Case Index File - Determines which model cases exist. It consists of a flag (Y or N) for the existence of all of the 1000 possible scenario case numbers (000 through 999) for each of the five models (MESORAD, DYNEV, CHEMS, OSPM, and D2) and the IPs.
CASEINDEX.DOC	Model Case Index Document File - Optional; contains the same data as the model case index file, but in an format that is more easily readable by the users. It is generated by the SHOWCNX utility.
CASEINDEX.LST	Model Case Index List File - Optional; contains the same data as the model case index file but in an format that is more easily readable by the users. It is generated by the LISTCNX utility.
CCS_TRACK.DAT	Checkout File - Maintains an ASCII list of who has which model or IP cases checked out for modification.
POPCODES.DAT	Population File - Contains an ASCII list of population databases stored in the SITE$POP: directory.
SETUP.COM	Site Setup Command File - Sets the various system logicals to point to this site. Many IBS programs run this file as part of the initialization phase.
[user.SITES.xxxx.BCK]	Backup Subdirectory - Contains backups of the data. Backups are created by using IBS -> SETUP -> BACKUP/RESTORE SITE DATA. The savesets are generated with the VAX/VMS Backup utility.
CHEMS_timestamp.BCK CHEMS_NDX_timestamp.BCK	CHEMS Case Backup Files - Generated when **Backup CHEMS Case data** is selected. These files are a backup of the CHEMS model scenarios and their indexes.
D2_timestamp.BCK D2_DMS_timestamp.BCK D2_NDX_timestamp.BCK	D2 Case Backup Files - Generated when **Backup D2 Case data** is selected. The files are a backup of the D2 model scenarios, the D2 topography and the D2 model scenario indexes.

Table 3.4 (contd)

FROM_timestamp.BCK	FROM Directory Backup File - Generated when **Backup data Received** is selected. It is a backup of the FROM directory.
IDYNEV_timestamp.BCK IDYNEV_NDX_timestamp.BCK	IDYNEV Case Backup Files - Generated when **Backup EVAC Case data** is selected. They are a backup of the IDYNEV model scenarios and their indexes.
IP_timestamp.BCK IP_NDX_timestamp.BCK	IPs Backup Files - Generated when **Backup IP Case data** is selected. They are a backup of the IPs and their indexes.
MESORAD_timestamp.BCK MESORAD_NDX_timestamp.BCK	MESORAD Case Backup Files - Generated when **Backup MESORAD Case data** is selected. They are a backup of the MESORAD model scenarios and their indexes.
OFFSITE_timestamp.BCK SITE_timestamp.BCK	Total Site Backup Files - Generated when **Backup All Site data** is selected. They are a backup of the OFF$MAPDIR: and SITE$TOPDIR: directories.
OSPM_timestamp.BCK OSPM_NDX_timestamp.BCK	OPSM Case Backup Files - Generated when Backup **OPSM Case data** is selected. They are a backup of the OPSM model scenarios and their indexes.
RA_timestamp.BCK RA_DMS_timestamp.BCK	Risk Area Backup Files - Generated when **Backup Risk Area data** is selected. They are a backup of the Known Polygons and the RP, SR, and DA topographies.
RM_timestamp.BCK RM_DMS_timestamp.BCK	Resource Management Backup Files - Generated when **Backup Resource Management data** is selected. They are a backup of the resource, facility, personnel, and special population data, and the RM topography.
SEND_timestamp.BCK	SEND Directory Backup File - Generated when **Backup data Sent** is selected. It is a backup of the SEND directory.
[user.SITES.xxxx.INPUT]	Model Input Subdirectory - Contains subdirectories that contain input for models, IPs, and population data.
[user.SITES.xxxx.INPUT.CHEMS]	CHEMS Model Input Subdirectory - Stores the different scenarios used as input to the CHEMS model. It is identified by the logical SITEICHEMS:.
CnnnCHEMS.GEN	CHEMS Model Input File - Input to the CHEMS model.
CRASHFLG.DAT	CHEMS Control File - Controls the way the CHEMS model behaves in abnormal situations.
[user.SITES.xxxx.INPUT.D2]	D2 Model Input Subdirectory - Stores the different scenarios used as input to the D2 model. It is identified by the logical SITEID2:.
CnnnD2INP.DAT	D2 Model Input File - Input to the D2 model.

Table 3.4 (contd)

D2_MASTER_FILE.DAT	D2 Case Index File - An index to all the D2 input files in the directory.
[user.SITES.xxxx.INPUT.DYNEV]	IDYNEV Model Input Subdirectory - Stores the different scenarios used as input to the IDYNEV model. It is identified by the logical SITEIDYNEV:.
CnnnEVAC.DMS	First Cut Network File - Contains a graphical representation of evacuation boundary or shelter point data created and maintained using the First-Cut approximation software (BUILDNET).
CnnnEVACPARAM.DAT	First Cut Control File - Controls how the first cut utility (BUILDNET) interprets road data.
CnnnINP.DAT	IDYNEV Model Input File - Input to the IDYNEV model.
CnnnLNK.DAT	Link Node Display File - Generated by the system when displaying an evacuation network. It can be deleted after displaying the network.
EVAC_SEARCH_HEADERS.DAT	IDYNEV Case Index File - An index to all the IDYNEV input files in the directory.
[user.SITES.xxxx.INPUT.IP]	Implementing Procedures Subdirectory - Stores the different implementing procedure scenarios. It is identified by the logical SITE$IP:.
CnnnACTIONS.DAT	IP Actions File - Contains a list of actions performed in the IP.
CnnnACTION_NOTES.DAT	IP Action Notes File - Contains notes relating to the actions to be performed.
CnnnMASTER_CHECKLIST.DAT	IP Master Checklist File - Contains the list of all tasks in the IP.
CnnnMASTER_NOTES.DAT	IP Master Checklist Notes File - Contains high level notes for the IP.
CnnnRESOURCE_CHECKLIST.DAT	IP Resource Checklist File - Contains a list of resources required to implement the IP.
CnnnRESOURCE_NOTES.DAT	IP Resource Notes File - Contains resource notes for the IP.
CnnnTASK_LINKS.DAT	IP Task Dependency File - Contains task dependency information for the IP.
CnnnTASK_NOTES.DAT	IP Task Notes File - Contains general task notes for the IP.
SCEN_INDEX.DAT	Scenario Index File - An index to the scenario table.
SCEN_TABLE.DAT	Scenario Table File - Links IP, D2, and IDYNEV cases with population datasets.

Table 3.4 (contd)

[user.SITES.xxxx.INPUT.MAPS]	Private Volatile Topographies - Logically part of the map database, contains the four private volatile topographies; D2, EX, FI, and VA. Unlike map topographies, these change and are generated by the users. It is identified by the logical SITE$MAPDIR:. These topographies are the output of the D2 and CHEMS models.
[user.SITES.xxxx.INPUT.MAPS.D2]	D2 Dispersion Plume Topography - Stores the D2 plumes generated from the different input scenarios by the D2 Dispersion model. See [*Map*.xx] for a description of the other files which may be present in the D2 topography. It is identified by the logical MAP$DATA$D2:.
CnnnD2.TXT	D2 Text File - Contains text data referenced by graphics objects in the associated DMS file.
CnnnPLUME.DMS	D2 Plume File - Contain the topographic data of the plume derived from the input scenarios.
LIMITD2.BIN	Geographic Limit File - Specifies the name and geographic limit of the DMS file that is the currently selected case.
[user.SITES.xxxx.INPUT.MAPS.EX]	Explosion Contour Topography - Stores the explosion contours generated from the different input scenarios by the CHEMS model. See [*Map*.xx] for a description of the other files which may be present in the EX topography. It is identified by the logical MAP$DATA$EX:.
CnnnEXP.DMS	CHEMS Explosion Contour - Contain the topographic data of the explosion contour derived from the input scenarios.
LIMITEX.BIN	Geographic Limit File - Specifies the name and geographic limit of the DMS file that is the currently selected case.
[user.SITES.xxxx.INPUT.MAPS.FI]	Fire Contour Topography - Stores the fire contours generated by the CHEMS model. See [*Map*.xx] for a description of the other files which may be present in the FI topography. It is identified by the logical MAP$DATA$FI:.
CnnnFI.DMS	CHEMS Fire Contour - Contains the topographic data of the fire contours derived from the input scenarios.
LIMITFI.BIN	Geographic Limit File - Specifies the name and geographic limit of the DMS file that is the currently selected case.
[INFOMANAGER.SITES.xxxx. INPUT.MAPS.VA]	Vapor Contour Topography - Stores the vapor contours generated by the CHEMS model. See [*Map*.xx] for a description of the other files which may be present in the VA topography. It is identified by the logical MAP$DATA$VA:.

Table 3.4 (contd)

CnnnVA.DMS	CHEMS Vapor Contour - Contains the topographic data of the vapor contour derived from the input scenarios.
LIMITVA.BIN	Geographic Limit File - Specifies the name and geographic limit of the DMS file that is the currently selected case.
[user.SITES.xxxx.INPUT.MESORAD]	MESORAD Model Input Subdirectory - Stores the different scenarios used as input to the MESORAD model. It is identified by the logical SITEIMESORAD:.
CnnnCHK.DAT	Dose Levels Checkpoint File - Contains checkpoints (locations where dose levels will be monitored) for the MESORAD model.
CnnnOBS.DAT	Weather Observations File - Contains weather observations for the MESORAD model.
CnnnPOL.DAT	Elevation File (polar coordinate) - Contains elevations for sectors of a polar coordinate grid used for close in calculations by MESORAD.
CnnnRSP.DAT	Run Specification File - Contains run specification data for MESORAD.
CnnnSnm.DAT	Release Specification File - Contains release specification data for MESORAD. Up to 24 files exist for each case (6 for each of 4 sources).
CnnnSTA.DAT	Weather Data File - Contains weather station definition data for MESORAD.
CnnnTER.DAT	Terrain File - Contains terrain data to be used in modifying the wind field.
CnnnTOP.DAT	Elevation File (Cartesian coordinate) - Contains ground level elevation to describe Cartesian elevation data.
CnnnV.DAT	Evacuation File - Contains evacuation data (sector statistics) produced by IDYNEV and used to compute dose to population.
[user.SITES.xxxx.INPUT.OFF]	Miscellaneous Subdirectory - Contains current risk area files and people location files. It is identified by the logical SITE$OFF:.
CR_RISK_AREA.DAT	Current Risk Area File - Contains a ASCII list of the polygons that make up the current risk area.
CR_RISK_AREA.DMS	Current Risk Area - A visual representation of the current risk area.
P_T_UPDATE.DAT	People Track Update Files - Distributes people track updates to other sites. Not generally used to preserve confidentiality.

Table 3.4 (contd)

PEOPLE_TRACK.DAT	People Location File - Contains people tracking information as entered by the LOCATE program.
[user.SITES.xxxx.INPUT.OSPM]	OSPM Model Input Subdirectory - Stores the different scenarios used as input to the OSPM model. It is identified by the logical SITEIOSPM:.
CnnnOSPM.DAT	OSPM Model Input File - Input to the OSPM model.
CnnnOSPM.DMS	OSPM Model Graphics File - Graphical location of siren icons.
[user.SITES.xxxx.INPUT.POP]	Population Subdirectory - Contains files associating raw population data with evacuation centroids. It is identified by the logical SITE$POP:.
pppp.DAT	Population Files - Associated raw population data with evacuation centroids. It is referenced in SITE$DIR:POPCODES.DAT:.
[user.SITES.xxxx.OUTPUT]	Private Model Output Subdirectory - Contains the directories that contain output from the models.
[user.SITES.xxxx.OUTPUT.CHEMS]	CHEMS Model Output Subdirectory - Originally used to store the non-topographic output from the CHEMS model. Currently, there is only topographic output from CHEMS, hence this directory is empty (or contains temporary work files). It is identified by the logical SITEOCHEMS:. Topographic output is stored in the [-.-.INPUT.MAP.EX], [-.-.INPUT.MAP.FI], and [-.-.INPUT.MAP.VA] subdirectories.
[user.SITES.xxxx.OUTPUT.D2]	D2 Model Output Subdirectory - Stores the output from the D2 model. It is identified by the logical SITEOD2:. Topographic output is stored in the [-.-.INPUT.MAP.D2] subdirectory.
CnnnD2LOG.DAT	D2 Log File - The complete execution of the D2 model as a user would see running D2PC from the command line.
CnnnDOSAGE.DAT	D2 Dosage File - Tabular output generated from CnnnPLOT.DAT and CnnnD2INP.DAT. This file contains all the tabular input required to create a D2 track on the map.
CnnnPLOT.DAT	D2 Plume Plot File - Tabular report listing plume half-widths for various distances from the dispersion source.
CnnnPOSINP.DAT	D2 / PARDOS Interface File - D2 output file used as input to the PARDOS model.
CnnnSCRATCH.DAT	D2 Scratch File - Temporary work file.

Table 3.4 (contd)

CnnnSCRIPT.DAT	D2 Input Script File - Script of D2 variables in the order they were fed into the model.
CnnnSLICE.DAT	D2 Time Slice File - PARDOS output file containing impact times for track tip, tail, and individual dosage levels.
CnnnSTEP.DAT	D2 Time Step File - Time based interpolation of information taken from PARDOS used to animate the plume's progress.
[user.SITES.xxxx.OUTPUT.DYNEV]	IDYNEV Model Output Subdirectory - Stores the output from the IDYNEV model. It is identified by the logical SITEODYNEV:. For more detail on the individual file contents, see Appendix A.
CnnnBIN.U36	Cumulative Sector Statistics File - Contains the cumulative sector statistics in both polar and Cartesian coordinates for the MESORAD models.
CnnnCOD.RPT	Report File - Contains tabular reports and diagnostics from users of the IDYNEV evacuation model and used input to IBS evacuation graphic functions for case nnn.
CnnnCOD.U37	Turn Percentages File - Contains turn movement percentages to the simulation model as calculated by the traffic assignment model.
CnnnLNK.DAT	Road Links File - Contains link statistics extracted from IDYNEV output to support rapid graphical display and animation of evacuation result.
CnnnNOD.DAT	Sign and Signal File - Contains sign and signal information extracted from the IDYNEV report CnnnCOD.RPT for quicker access during response to user picks of nodes from graphic display of evacuation results.
CnnnSUM.RPT	Summary File - Contains a one screen summary of IDYNEV simulation results extracted from CnnnCOD.RPT and reformatted for screen display.
CnnnTV.DAT	Time and Volume File - Contains time stamps and link volume statistics extracted from CnnnCOD.RPT.
NODE_POS.DAT	Node Position File - Contains position data and link identification information of nodes extracted from Tiger Trails data by utility TTXTTDAT.
NODE_POS.SRT	Sorted Node Position File - Sorted version of NODE_POS.DAT using the entire record as the key in ascending order.
[user.SITES.xxxx.OUTPUT.MESORAD]	MESORAD Model Output Subdirectory - Stores the output from the MESORAD model. It is identified by the logical SITEOMESORAD:.

Table 3.4 (contd)

CnnnAIRDPL.typ	<u>Air Deposition Files</u> - Provides Cartesian and polar matrix of time integrated ground-level air concentrations with deposition, washout, decay, and ingrowth for each advection (time) step (.ADV), and cumulative from Start to Current time step (.CUM).
CnnnAIRNDP.typ	<u>Air Non-Deposition Files</u> - Provides Cartesian and polar matrix of time integrated ground-level air concentrations assuming no depletion and no radioactive decay for each advection (time) step (.ADV), and cumulative from Start to Current time step (.CUM).
CnnnCONTOUR.DMS	<u>Contour Graphics File</u> - Contains a graphic representation of dose/concentration isopleths, created using the graphic display module from MESORAD output.
CnnnDEPOST.typ	<u>Surface Concentrations Files</u> - Provides Cartesian and polar matrix of surface concentrations resulting from depletion process - deposition and washout for each advection (time) step (.ADV), and cumulative from Start to Current time step (.CUM).
CnnnEXTGRD.typ	<u>Ground Radation Files</u> - Provides Cartesian and polar matrix of external dose to whole body from exposure to radionuclides deposited on the ground for each advection (time) step (.ADV), and cumulative from Start to Current time step (.CUM).
CnnnEXTSIC.typ	<u>Semi-Infinite Cloud Files</u> - Provides Cartesian and polar matrix of external dose to whole body from exposure to radionuclides in semi-infinite cloud for each advection (time) step (.ADV), and cumulative from Start to Current time step (.CUM).
CnnnFLD.RPT	<u>Field Teams Report File</u> - Provides a report of MESORAD output covering Iodine air concentrations, Gamma and Gamma + Beta simulated readings for use by field teams during REPSS exercises.
CnnnGBRADV.BIN	<u>Gamma/Beta Radiation File</u> - Provides Cartesian matrix of Gamma + Beta dose rate for each advection (time) step.
CnnnGDRADV.BIN	<u>Gamma Dosage Rate File</u> - Provides Cartesian matrix of Gamma dose rate for each advection (time) step.
CnnnING.RPT	<u>Ingestion Report File</u> - Provides a report of MESORAD output covering ingestion pathway doses.
CnnnINHLNG.typ	<u>Inhalation Lung Files</u> - Provides Cartesian and polar matrix of 50-year dose commitment to the lung from inhaled radionuclides for each advection (time) step (.ADV), and cumulative from Start to Current time step (.CUM).

Table 3.4 (contd)

CnnnINHTHY.typ	Inhalation Thyroid Files - Provides Cartesian and polar matrix of dose commitment to thyroid from inhaled radionuclides for each advection (time) step (.ADV), and cumulative from Start to Current time step (.CUM).
CnnnINHWBD.typ	Inhalation Whole Body Files - Provides Cartesian and polar matrix of inhalation dose to the whole body for each advection (time) step (.ADV), and cumulative from Start to Current time step (.CUM).
CnnnIODADV.BIN	Iodine File - Provides a 31X31 Cartesian matrix of Radio Iodine concentrations for each advection (time) step.
CnnnLOG.RPT	Log File - Provides a textual report of input, diagnostic messages, and some output results of the MESORAD model.
CnnnPRM.RPT	Population Report File - Provides a report of total whole body and thyroid dose to population for each advection (time) step (current and cumulative).
CnnnPUFFTR.BIN	Puff Trace File - Provides coordinate information on puffs released from MESORAD sources for each advection (time) step.
CnnnSTATIN.BIN	Weather Station File - Provides wind speed and direction information on weather stations for each advection (time) step.
CnnnTOTWBD.typ	Total Source Whole Body Files - Provides Cartesian and polar matrix of total whole body dose from inhalation, ground, and cloud shine for each advection (time) step (.ADV), and cumulative from Start to Current time step (.CUM).
CnnnWINDFD.BIN	Wind File - Provides 16X16 matrices of wind speed and direction (wind fields) for each advection (time) step.
CnnnXmm.DAT	Checkpoint Count File - Provides a report of thyroid and total whole body doses projected at specified checkpoints for each advection (time) step.
[user.SITES.xxxx.OUTPUT.OSPM]	OSPM Model Output Subdirectory - Stores the output from the OSPM model. It is identified by the logical SITEOOSPM:. For more detail on the individual file contents, see Appendix A.
CnnnCONTOUR.DMS	Sound Pressure Contours File - Contains sound pressure contours generated by OSPM.
CnnnDBA.DAT	DbA File - contains sound pressure values (DBA) generated by OSPM for each cell in the OSPM model grid.

Table 3.4 (contd)

CnnnDBC.DAT	DbC File - Contains sound pressure values (DBC) generated by OSPM for each cell in the OSPM model grid.
CnnnOCTm.DAT	Db by Octave File - Contains sound pressure values for octave band m generated by OSPM for each cell in the OSPM model grid.

3.1.3 ONSITE Directory

The ONSITE directory is used to receive data from either the onpost system or the Onpost Simulator program. The EVENT program checks each data file. If valid, EVENT processes and copies the file to the OFF$FROMDIR: directory. Invalid files are deleted and the onpost is notified. Consequently, the ONSITE directory is usually empty.

The ONSITE directory consists of data files transferred from the onpost (or simulated with the Onpost Simulator program). The files are tested for validity by the EVENT program and copied to the offsite database. Components of the ONSITE directory are shown in Figure 3.1, and are described in Table 3.5. Specific file names are shown in uppercase letters; variable parts of file names and of directory names are shown in lowercase letters. Refer to the ICD for more information about individual files.

Table 3.5. Components of the ONSITE Directory

[ONSITE]	Onpost Communications Directory - Contains files that have been sent either from onpost or from the Onpost Simulator program. It is identified by the logical ONSITE$DIR:. If a file is valid and can be processed by EVENT, it will be copied to the [OFFSITE.xxxx.FROM] directory. EVENT deletes each file from ONSITE$DIR: as the file is processed; therefore this directory is typically empty.
generic.ATT	Generic Geographic Attribute File - Each geographic object in a Digital Line Graph (DLG) file can have attributes (tower height or road width, for instance) that are stored in this file.
generic.DAT	Generic Text File - Any ASCII file that the onpost chooses to send. Although onpost can send anything, the EVENT program uses only the first 80 columns and 1000 rows.
generic.DLG	Generic Geographic Map File - The common transfer format of geographic information between IBS and EMIS is via USGS DLG files. A DLG file will be accompanied by an ATT file. In reality, the onpost cannot send this set of files, although IBS can send them to EMIS; they simply are ignored.
BASE_POP.DAT	Base Population File - The various types of population of the onpost.
CALLLIST.DAT	Emergency Contact List File - List of people onpost to contact, and their responsibilities.

Table 3.5 (contd)

CAS_SUM.DAT	Casualty Summary File - Summary of the casualties that have been reported onpost.
D2INPnnn.DAT	D2 Input File - D2 input file from the onpost. Used with D2LOGnnn.DAT, DOSAGnnn.DAT, and SLICEnnn.DAT. Upon receipt, the plume is calculated and made available as the WD topography.
D2LOGnnn.DAT	D2 Log File - D2 log from the onpost. Used with D2INPnnn.DAT, DOSAGnnn.DAT, and SLICEnnn.DAT. Upon receipt, the plume is calculated and made available as the WD topography.
DOSAGnnn.DAT	D2 Dosage File - D2 results from the onpost. Used with D2INPnnn.DAT, D2LOGnnn.DAT, and SLICEnnn.DAT. Upon receipt, the plume is calculated and made available as the WD topography.
ENVIRON.DAT	Environmental Icon Update File - Environmental observations used to update the EI topography.
KEY.DAT	Keyed Data File - Simple handshaking between IBS and EMIS occurs through key files that tell each other about the validity of the last files sent.
KNOWNPTS.DAT	Known Points File - Supplies known points to IBS from the onpost. It is used to update the [OFFSITE.xxxx]KNOWN_POINTS.DAT files. It contains latitude, longitude, description, and attributes.
MLUPDATE.DAT	Mail List Update - Used to add, delete, or modify one or more entries in the master IBS mail list that keeps track of all users, both onpost and offpost, to whom IBS can send mail. This file is unique in this directory in that it can be created by IBS applications (NEWUSER, MLMGR) and placed here. EVENT uses this file to update the system mail list.
NOTIFY.DAT	Notification File - Used to notify IBS and the civilian Emergency Operation Centers of an event.
SIGEVENT.DAT	Significant Event File - Contains a list an onpost user considered as significant activities during the current event.
SLICEnnn.DAT	D2 Time Slice File - PARDOS results from the onpost. Used with D2INPnnn.DAT, D2LOGnnn.DAT, and DOSAGnnn.DAT. Upon receipt, the plume is calculated and made available as the WD topography.
TOWERSEL.DAT	Weather Tower Selection File - Used in conjunction with the weather file to select a specific set of weather parameters for use in D2.
WEATHER.DAT	Weather File - Generated by onpost meteorological towers and includes wind speed and direction, temperature, and humidity. This data is used with TOWERSEL.DAT to update the WI topography.
WORKPLAN.DAT	Onpost Work Plan File - Contains the activities the onpost intends to pursue during some period of time, usually one day.
XFERLIST.DAT	Transfer List File - The list of files being transferred from onpost to IBS.

3.1.4 OFFSITE Directory

The OFFSITE directory is divided into offsite databases that are used to share data, either between IBS users, or with the onpost system. During planning, users can access and share resource, facility, and personnel data. The users can also access and share DA (damage assessment areas), RM (facilities and resources), RP (risk areas), and SR (search and rescue areas) topographies. In operational mode, current data and reports are sent to and received from the onpost. In addition, the onpost is capable of updating the EI (environmental icons), WI (weather icons), and WD (onpost D2 track) topographies.

The OFFSITE directory is divided into offsite databases. Each offsite database contains data shared between IBS users, data sent to the onpost, data received from the onpost, and shared volatile topographies. Each operational site (see Section 3.1.2) will point to an offsite database with the same name. Each planning site will point to an offsite database, but not necessarily with the same name. Components of the OFFSITE directory are shown in Figure 3.1 and described in Table 3.6. Specific file names are shown in uppercase letters; variable parts of file names and directory names are shown in lowercase letters. Refer to the ICD for more information about individual files.

Table 3.6. Components of the OFFSITE Directory

[OFFSITE]	<u>Offsite Directory</u> - This directory is the main (top-level) directory of the Offsite directory (identified by the logical OFFSITE$DIR:).
[OFFSITE.xxxx]	<u>Offsite Database Subdirectory</u> - The notation [.xxxx] indicates that many different subdirectories can be here, each with a 4-letter name (for example, BPNL) and designating a specific offsite database. An offsite database contains data shared between IBS users, subdirectories to exchange data with the onpost, and shared volatile topographies. The currently selected offsite database is identified by the logical OFF$DIR:. For each site database under INFOMANAGER, an offsite database with the same name is included. Each planning site database also has an associated offsite database; however, it does not need to have the same name.
AGENCY_LOOKUP.DAT	<u>Agency Lookup File</u> - A list of agencies and their abbreviations.
COMBINED.LOG	<u>Combined Log File</u> - Log containing selected data from journals [SPOCC_ACTIVITY.LOG*, SPOCC_EVENT.LOG*, and WATCH_EVENT.LOG*, and *important* mail (IMPMAIL.LOG)]. Users can have a private version of this file in SYS$LOGIN: also.

Table 3.6 (contd)

EVR_route.DMS	Evacuation Routes Topography File - Contains the different routes that are available for evacuation, listed in KNOWN_ROUTES.DAT.
FACILITY.DAT	Facility File - List of the facilities, their type, address, point of contact, and information on the existence of mutual aid agreements.
FACILITY_TRACK.DAT	Facility Track File - Used by RESOURCE MANAGEMENT and LOCATE, contains a list of the facilities, whether or not they are active, and the number of people who have come through them.
IMPMAIL.LOG	Important Mail File - Contains selected mail messages the DBA thought were important.
KNOWN_POINTS.DAT	Known Points File - Known points on the map. Contains the name of the point and its location (latitude and longitude).
KNOWN_POLYGONS.DAT	Known Polygons File - Known polygons on the map. Contains polygon id and name.
KNOWN_ROUTES.DAT	Known Routes File - List of the available routes. Contains route id and name. The name of the EVR_route.DMS file is calculated based on the name.
PERSONNEL.DAT	Emergency Response Personnel File - List of emergency response personnel, their titles, telephone numbers, addresses, availability, and work schedules.
POSITION_LOOKUP.DAT	Agency Position File - A list of agency abbreviations and the positions at that agency.
RESOURCE.DAT	Resource File - A list of available resources (including quantity) and their location (facility name).
SITE_EF.DAT	Site Emergency Function File - List of the emergency functions that can be used during the creation of implementing procedures.
SPECIAL_POPULATION.DAT	Special Populations File - Contains facilities with special needs concerning their populations (such as, transportation dependent).
SPOCC_ACTIVITY.LOG	IBS Journal Index File - Log file for database changes, including record changed, time stamp, and pointers to detail records.
SPOCC_ACTIVITY.LOG_DETAIL	IBS Journal File - Free-format text description for database changes.
SPOCC_EVENT.LOG	IBS to Onpost Journal Index File - Log of reports sent to the onpost system from the local IBS node, including type of report, time stamp, and pointers to detail records.

Table 3.6 (contd)

SPOCC_EVENT.LOG_DETAIL	IBS to Onpost Journal File - Free format text description for reports sent to onpost.
WATCH_EVENT.LOG	Onpost to IBS Journal Index File - Log of incoming reports sent from the onpost system to the local IBS node, including type of report, time stamp, and pointers to detail records.
WATCH_EVENT.LOG_DETAIL	Onpost to IBS Journal File - Free format text description for reports sent from onpost.
WORK_PLAN.DAT	Work Plan File - Contains information on a single activity selected from the onpost work plan.
[OFFSITE.xxxx.FROM]	From Onpost Subdirectory - Contains data that has been sent from onpost (or simulated data generated by the Onpost Simulator program), and has been processed by EVENT and copied here. It is identified by the logical OFF$FROMDIR:. For a list and description of the files, see Table 3.5.
[OFFSITE.xxxx.SEND]	To Onpost Subdirectory - Contains files that an IBS user has sent to the onpost. For more detail on any file, see the ICD.
generic.ATT	Generic Ancillary Attribute File - Ancillary attribute file created in this directory whenever a DLG file is created. It will have the same prefix as the associated DLG file.
generic.DAT	Generic Text File - Any ASCII file that the user chooses to send. At this time, IBS can send anything, but EMIS will only process GENERIC.DAT.
generic.DLG	Generic Digital Line Graph File - DLG files are created when graphic data is to the onpost. Evacuation route files begin with EVR_.
D2INPnnn.DAT	D2 Input File - D2 input file sent to the onpost.
D2LOGnnn.DAT	D2 Log File - D2 log file sent to the onpost.
DOSAGnnn.DAT	D2 Dosage File - D2 dosage file sent to the onpost.
KEY.DAT	Keyed Data File - Simple handshaking between IBS and EMIS occurs through key files that tell each other about the validity of the last files sent.
MLUPDATE.DAT	Mail List Update - Used to add, delete, or modify one or more entries in the master mail list that keeps track of all users, both onpost and offpost, to whom IBS can send mail.
OFF_CAS.DAT	Offpost Casualties File - Summary of the casualties that have reported to civilian facilities.
OFF_EVR.DAT	Offpost Evacuation Routes File - Lists the planned evacuation routes, and their graphic file (EVR_route.DLG).

Table 3.6 (contd)

OFF_KP.ATT OFF_KP.DAT OFF_KP.DLG OFF_KP.DMS	Known Points Files - Known points are converted from IBS Digital Mapping Format (*.DMS) and supplied to the onpost in Digital Line Graph (*.DLG and *.ATT), and ASCII text (*.DAT) formats.
OFF_KPL.DAT	Known Polygons File - Known polygons in IBS to be sent onpost (typically Emergency Planning Zone and Risk Area).
OFF_PA.DAT	Protective Action File - Report to the onpost of protective action, including who ordered it, evacuation route names, EPZs moved to shelters, EPZs evacuated, and the reception center to use.
OFF_PEVR.DAT	Proposed Evacuation Routes File - Lists the proposed evacuation routes, and their graphic file (EVR_route.DLG).
OFF_PIMR.DAT	Proposed Military Routes File - Lists the proposed routes for the military response teams, and their graphic file (EVR_route.DLG).
OFF_RCD.DAT	Reception Center Directory File - Lists the reception center, its address, and responsible contacts.
OFF_RCR.DAT	Reception Center Report File - Summary of the current load at reception centers and the number of people who have passed through them.
OFF_SHLT.DAT	Offpost Shelter Report File - Summary of health-related statistics for people at shelters, reception centers, hospitals, etc..
OFF_STAT.DAT	Offpost Status Report File - Reports the current offpost status, including event status, local ID and authority, proposed protective action decision, and hospitals for onpost casualties.
OFF_TC.DAT	Traffic Control Report File - Status report on traffic control points.
SLICEnnn.DAT	D2 Time Slice File - PARDOS time slice file sent to the onpost.
XFERLIST.DAT	Transfer List File - List of the files to be transferred to the onpost.

Table 3.6 (contd)

[OFFSITE.xxxx.xx]	Shared Volatile Topography - These subdirectories store the shared volatile topographies. They are identified by the logical MAP$DATA$xx:. EI, WD, and WI are generate from data sent from the onpost; DA, RM, RP, and SR are generated from data originating on IBS. With the exception of WD, each of the following directories will contain a xx.DMS, xx.TXT (optional), and a LIMITxx.BIN file. Contents of a topography subdirectory are described in Table 3.3, [*Map*.xx]. DA - Damage Assessment Areas Topography - Created by the Damage Assessment application in IBS (IBS -> EMERGENCY ACTIVITIES -> DAMAGE ASSESSMENT). EI - Environmental Icon Topography - Created and updated base on environment information sent from the onpost (ONSITE$DIR:ENVIRON.DAT). RM - Facilities and Resources Topography - Created by the Resource Management application in IBS, it is not actually used there, but can be used in other parts of IBS. RP - Risk Areas Topography - Created by the Risk Area Analysis application in IBS (IBS -> EMERGENCY ACTIVITIES -> DIRECTION/CONTROL). SR - Search and Rescue Areas Topography - Created by the Search and Rescue application in IBS (IBS -> EMERGENCY ACTIVITIES -> SEARCH/RESCUE). WD - Onpost D2 Track Topography - Created based on D2 output information sent from the onpost. The DMS and TXT file names have the format CnnnWD.DMS and CnnnWD.TXT, and will have the same case number as their input (ONSITE$DIR: D2INPnnn.DAT, D2LOGnnn.DAT, DOSAGnnn.DAT, and SLICEnnn.DAT). The most recently sent input data will be the current selected model case. WI - Weather Icon Topography - Created and updated based on weather information sent from the onpost (ONSITE$DIR:TOWERSEL.DAT and WEATHER.DAT).

3.1.5 ISYSFILES Directory

The ISYSFILES directory contains system command files, templates, data for the Onpost Simulator program, and other configuration data. ISYSFILES is used in both planning and operational modes.

Components of the ISYSFILES directory are shown in Figure 3.1 and described in Table 3.7. Specific file names are shown in uppercase letters while variable parts of file names and directory names are shown in lowercase letters.

Table 3.7. Components of the ISYSFILES Directory

[ISYSFILES]	ISYSFILES Directory - Main (top-level) directory of the ISYSFILES directory (identified by the logical IEMIS$SYSF:). The ISYSFILES directory contains administrative, general database, and template files. Because of the large number of files in the ISYSFILES directory, the files have been divided based upon function.
System: Provides IBS with data necessary for its execution.	
ALLSITE.DAT	All Sites File - List of all of the available sites on the system. It is created by the utility ALLSITE. It is also created by the utility FIXMSTRDIR, prior to creating the file MASTERDIR.DAT.
COLORTBL.DAT	Color Table File - Defines the red-green-blue intensities for each color.
MAILLIST.DAT	Mail List File - Directory of user names and mail addresses across all IBS systems.
MAIL_SYSTEMS.DAT	IBS Mail Systems File - List of all IBS systems (node, password, and directory names). Allows the EVENT program to keep MAILLIST.DAT on the other IBS systems current. EVENT will route MLUPDATE.DAT to each of the system listed.
MASTERDIR.DAT	Site Master Directory File - List of the valid disk, user and site names on the system. This file is created by the utility FIXMSTRDIR.
OFF_TO_ON.DAT	Offpost/Onpost Site Name Cross Reference File - List of offpost site names and their corresponding onpost site name.
OTHER_SYSTEMS.DAT	Other IBS Systems File - List of other IBS offpost systems (node, password, and directory names). Allows the onpost system to communicate with a single offpost system, and that offpost system will route the data to the other offpost systems.
PERMITI.DAT	IBS User Privileges File - A list of users and their IBS execution privileges.
PERMITI.DFT	IBS GroupPrivileges File - A list of IBS groups and their privileges.
POST_SYSTEM.DAT	Post System File - Provides IBS with the information necessary to communicate with the onpost system (node, user name, and password).

Table 3.7 (contd)

Setup: Executed to establish the environment at bootup or login. Typically, they define the appropriate logicals and symbols.

ASS_DISK_LOG.COM	Assign Disk Logicals Command File - Assigns most logicals concerning the disk.
ASS_IBS_LOG.COM	Assign IBS Logicals Command File - Assigns the logicals concerning IBS.
ASS_IBS_SYM.COM	Assign IBS Symbols Command File - Defines symbols to allow the user to execute the different programs within IBS without specifying the executable location.
ASS_MAP_LOG.COM	Assign Map Logicals Command File - Assigns the logicals concerning the Map database.
ASS_MAP_SYM.COM	Assign Map Symbols Command File - Defines symbols to enable the user to change from one map data base to another.
ASS_SITE_LOG.COM	Assign Site Logicals Command File - Assigns the logicals specific to the site.
PNLBATCH.COM	Start IBS Background Processes Command File - Starts the IBS background processes (e.g., Event and Mail Check).
PNLEXECUT.COM	Execute Command File - Installs IBS and starts the IBS background processes.
PNLINSTALL.COM	Install Command File - Installs IBS (e.g. CHGOSITE, XMAIL, and Shared Memory).
PNLLOGDEF.COM	Define Logicals Command File - Executes ASS_DISK_LOG.COM, ASS_IBS_LOG.COM, ASS_MAP_LOG.COM, and ASS_SITE_LOG.COM in correct order; assigning all necessary logicals needed to run IBS.
PNLSYMDEF.COM	Assign Symbols Command File - Executes ASS_IBS_SYM and ASS_MAP_SYM, assigning all necessary symbols needed to run IBS.

Switches : The content of these files is undefined. IBS is concerned only with their existence. The file can be named either *.HLT to halt the process or *.HLT-OFF to enable the process. IBS is not operational unless the state of these switches is *.HLT-OFF.

EVENT.HLT-OFF	EVENT Switch - Enables/disables the execution of the EVENT process.
METSIM.HLT-OFF	METSIM Switch - Enables/disables the execution of the METSIM process.

Debug Switches : The content of these files is undefined. IBS is concerned only with their existence. The file can be named either *.DBG to provide debugging data or *.DBG-OFF not to. The normal state of these switches is *.DBG.

Table 3.7 (contd)

EVENT.DBG	EVENT Debug Switch - Enables the output of debugging data from the EVENT program.
FIXKPL.DBG	Fix Known Polygons Debug Switch - Enables the output of debugging data from the FIXKPL utility.
MAIL_C.DBG	Mail Check Debug Switch - Enables the output of debugging data from the MAIL_C program.
MAPDBGEN.DBG	Map Database Generate Debug Switch - Enables the output of debugging data from the MAPDBGEN program.
MAPDBUPDT.DBG	Map Database Update Debug Switch - Enables the output of debugging data from the MAPDBUPDT program.

Templates : Used as a starting point to create data with the correct file formats. Some need to be modified by the DBA before use. For a list of the templates, their target files, whether or not they require modification, and the application that references them, see Table 3.8.

BLANKICON.DMS	Blank Icon Template - Used by the GEDIT utility to create a new icon. The new icon file is then placed in the icon directory (MAP$ICONS:) of the current map database.
CASEINDEX.DAT_TEMPLATE	Case Index Template - A Case Index file, SITE$DIR:CASEINDEX.DAT, with no valid case numbers.
FTP.COM_TCPWARE FTP.COM_TEMPLATE FTP.COM_WINTCP	FTP Templates - Two different commercial software packages can be used to communicate with the onpost system using File Transfer Protocol (FTP): Pathway Access, formerly called Win/TCP, and TCP ware. Each uses a slightly different syntax, as specified in FTP.COM_WINTCP and FTP.COM_TCPWARE, respectively. The appropriate file should be copied to FTP.COM_TEMPLATE, where IBS can access it.
JOB_ENV.DAT_TEMPLATE	Job Environment File Template -Used by the utility NEWUSER. When a new user account is created, the template is used to create the JOB_ENV.DAT file for the new user.
LOGIN.COM_TEMPLATE	Login Command File Template - Used by the utility NEWUSER. When a new user account is created, the template is used to create the LOGIN.COM file for the new user.
POPCODES.DAT_TEMPLATE	Population File Template - Used by the utility NEWUSER. When a new user account is created, the template is used to create the POPCODES.DAT file for the new user.
SETUP.COM_TEMPLATE	Site Setup Command File Template - Used by the utility NEWUSER. When a new user account is created, the template is used to create the SETUP.COM file for the new user.
SETUPDB.COM_TEMPLATE	SetupDB Command File Template - Used by the utility FIXMAPDB. When a new Map database is created, it is used to create the SETUPDB.COM file.

Table 3.7 (contd)

SITE.DAT_TEMPLATE	<u>Site Template</u> - Used by the utility NEWUSER. When a new user account is created, the template is used to create the SITE.DAT file for the new user.
SITE_EF.DAT_TEMPLATE	<u>Emergency Functions Template</u> - Template of default emergency functions used by NEWSITE to create (if necessary) the OFF$DIR:SITE_EF.DAT file.
TOPODESC.FIL_TEMPLATE	<u>Topography Description File Template</u> - Used by the utility FIXMAPDB. When a new Map database is created, it is used to create the TOPODESC.FIL file.
Import Format Files : Provide the format necessary to import each field of the specified file from an import file. The import file is typically generated by another IBS.	
AG.FMT	<u>Agency Format File</u> - Format to update the OFF$DIR:AGENCY_LOOKUP.DAT file.
FAC.FMT	<u>Facility Format File</u> - Format to update the OFF$DIR:FACILITY.DAT file.
KP.FMT	<u>Known Points Format File</u> - Format to update the OFF$DIR:KNOWN_POINTS.DAT file.
PER.FMT	<u>Personnel Format File</u> - Format to update the OFF$DIR:PERSONNEL.DAT file.
POS.FMT	<u>Position Format File</u> - Format to update the OFF$DIR:POSITION_LOOKUP.DAT file.
RES.FMT	<u>Resource Format File</u> - Format to update the OFF$DIR:RESOURCE.DAT file.
Miscellaneous files	
COLORTEST.DMS	<u>Color Test File</u> - A diagram that allows for a quick test of the color settings.
DLGXDMS.ATT	<u>DLG/DMS Cross Reference File</u> - Cross reference used to convert DLG attributes to DMS attributes.
PROBLEM.RPT	<u>Problem Report Form</u> - Form for filing a problem report. This file is for user information only, and is not used by IBS.
RPTSUBJ.DAT	<u>Report Subject File</u> - Allows the user to create search criteria for producing custom reports from the journal and important mail files (IBS -> MESSAGE BOARD -> CREATE COMMON LOG REPORT or CREATE PRIVATE LOG REPORT).
SIREN.LIB	<u>Siren Library</u> - A list of sirens and their output (in decibels) at specific frequencies. Used by the OSPM model.

Table 3.7 (contd)

SITE.DAT	Depot Sites File - List of depot sites. Used to create new map databases or by the user when creating new sites.
UPDATE.DOC	Changes in this Version of IBS - A description of what has been changed in the [ISYSFILES...] directory for this version of IBS. This file is for user information only, and is not used by IBS.

Sample programs: Because different FTP packages can be used, these programs may need to be changed or replaced. These programs are for WIN/TCP, now called PATHWAY ACCESS from Wollongang.

FTPCHECK.COM	Check FTP Command File - Check on whether or not communications have been established with the post.
FTPSTART.COM	Start FTP Command File - Establish communications with the post.
FTPSTOP.COM	Stop FTP Command File - Stop communications with the post.
SELECTDB.COM	Select Map DB Command File - Allows the user to select the appropriate map database.

Model Input templates: These files are templates for the input files to the associated model. For a description of the individual files, see model subdirectories under [user.SITES.xxxx.INPUT.*], Table 3.4.

C000CHEMS.GEN CRASHFLG.DAT	Chems Model Templates
C000D2INP.DAT	D2 Model Template
C000EVACPARAM.DAT C000INP.DAT	IDYNEV Model Templates
C000OSPM.DAT	OSPM Model Template
C000CHK.DAT C000OBS.DAT C000POL.DAT C000RSP.DAT C000Snm.DAT C000STA.DAT C000TER.DAT C000TOP.DAT C000V.DAT	MESORAD Model Templates

D2 Igloo Files: Chemicals weapons are stored in underground bunkers called igloos. These files give the location and unclassified contents of the igloos at each of the eight depots.

IGL_AAD.DAT IGL_DPG.DAT IGL_EWA.DAT IGL_LBG.DAT IGL_NAP.DAT	D2 Igloo Files - Location and unclassified contents of the igloos at the depot.

Table 3.7 (contd)

IGL_PAD.DAT IGL_PBA.DAT IGL_UAD.DAT	D2 Igloo Files (cont.) - Location and unclassified contents of the igloos at the depot.
[ISYSFILES.ATT]	Attributes Subdirectory - Contains the latest release of the GIS attributes. The GIS programs do not access this subdirectory, but access the [*Map*.ATT] subdirectory. To incorporate the latest attributes, the files in this directory must be copied to the [*Map*.ATT] subdirectory. For a list of the contents of this subdirectory, see the [*Map*.ATT] subdirectory.
[ISYSFILES.ICON]	Icon Subdirectory - Contains the latest release of the icons available to IBS. The GIS programs do not access this subdirectory, but access the [*Map*.ICON] subdirectory. To incorporate the latest icons, the files in this directory must be copied to the [*Map*.ICON] subdirectory. For a list of the contents of this subdirectory, see the [*Map*.ICON] subdirectory.
[ISYSFILES.RELEASE]	Release Subdirectory - Contains miscellaneous files for creating releases of IBS.
[ISYSFILES.SIM]	Onpost Simulator Subdirectory - Contains files which the Onpost Simulator program will copy into the ONSITE directory. These files can be edited to insert in the appropriate data. For a list of the contents of this subdirectory, see the [ONSITE] directory.
[ISYSFILES.SIM.nn]	Onpost Simulator Templates Subdirectories - The notation [.nn] indicates that many different subdirectories can be located here, each having a 2-letter name (such as, WA) and designating the 2-character state code for the CSEPP site. A site subdirectory contains templates for the Onpost Simulator program. Because these files are site dependent, the files in the appropriate subdirectory should be copied to [ISYSFILES.SIM]. For a list of the contents of this subdirectory, see the [ONSITE] directory.
[ISYSFILES.nn]	Site-Specific Model Templates Subdirectories - The notation [.nn] indicates that there can be many different subdirectories here, each having a 2-letter name (such as, WA) and designating the 2-character state code for the CSEPP site. A site subdirectory contains templates for some of the models. The files in the appropriate subdirectory should be copied to [ISYSFILES].
C000CHEMS.GEN C000D2INP.DAT C000INP.DAT C000OSPM.DAT	Site Specific Model Templates File - Used for model input, but contain some data that is site-specific.

The following template files are configured by the DBA for the particular system. Table 3.8 identifies the template files stored in the IEMIS$SYSF: directory, their respective target file, together with a Y/N flag indicating whether the DBA should modify it for the particular system, and the primary application that uses it. Complete descriptions and formats for each template target file are presented in Appendix A.

Table 3.8. Template Files Stored in IEMIS$SYS:

Filename	Target File (Directory and Name)	Modify	Application
BLANKICON.DMS	MAP$ICONS:*.DMS	N	GEDIT
TOPODESC.FIL_TEMPLATE	MAP$DATA$MAIN:TOPODESC.FIL	Y	FIXMAPDB
SETUPDB.COM_TEMPLATE	MAP$DATA$MAIN:SETUPDB.COM	N	MAPDBGEN
JOB_ENV.DAT_TEMPLATE	SYS$LOGIN:JOB_ENV.DAT	Y	NEWUSER
LOGIN.COM_TEMPLATE	SYS$LOGIN:LOGIN.COM	Y	NEWUSER
CASEINDEX.DAT_TEMPLATE	SITE$DIR:CASEINDEX.DAT	N	NEWSITE
POPCODES.DAT_TEMPLATE	SITE$DIR:POPCODES.DAT	N	NEWSITE
SETUP.COM_TEMPLATE	SITE$DIR:SETUP.COM	N	NEWSITE
SITE.DAT_TEMPLATE	SYS$LOGIN:SITE.DAT	N	NEWSITE
SITE_EF.DAT_TEMPLATE	OFF$DIR:SITE_EF.DAT	Y	NEWSITE
CRASHFLG.DAT	SITEICHEMS:CRASHFLG.DAT	Y	NEWSITE
C000CHEMS.GEN	SITEICHEMS:C000CHEMS.GEN	Y	NEWSITE
C000D2INP.DAT	SITEID2:C000D2INP.DAT	Y	NEWSITE
C000INP.DAT	SITEIDYNEV:C000INP.DAT	Y	NEWSITE
C000EVACPARAM.DAT	SITEIDYNEV:C000EVACPARAM.DAT	Y	NEWSITE
C000CHK.DAT	SITEIMESORAD:C000CHK.DAT	Y	NEWSITE
C000OBS.DAT	SITEIMESORAD:C000OBS.DAT	Y	NEWSITE
C000POL.DAT	SITEIMESORAD:C000POL.DAT	Y	NEWSITE
C000RSP.DAT	SITEIMESORAD:C000RSP.DAT	Y	NEWSITE
C000Snm.DAT	SITEIMESORAD:C000Snm.DAT	Y	NEWSITE
C000STA.DAT	SITEIMESORAD:C000STA.DAT	Y	NEWSITE
C000TER.DAT	SITEIMESORAD:C000TER.DAT	Y	NEWSITE
C000TOP.DAT	SITEIMESORAD:C000TOP.DAT	Y	NEWSITE
C000V.DAT	SITEIMESORAD:C000V.DAT	Y	NEWSITE
C000OSPM.DAT	SITEIOSPM:C000OSPM.DAT	Y	NEWSITE
FTP.COM_TCPWARE	IEMIS$SYSF:FTP.COM_TEMPLATE	N	N/A
FTP.COM_WINTCP	IEMIS$SYSF:FTP.COM_TEMPLATE	N	N/A
FTP.COM_TEMPLATE	SYS$LOGIN:FTP.COM	N	N/A

3.2 IBS Functional Databases

This section provides a general overview of the functional databases within the IBS. These databases contain the geographic data files and ASCII and binary control files necessary to support an emergency management capability within the software. The IBS functional databases supported by the data management system include:

D2 Dispersion Model
Damage Assessment
Hazardous Chemical Modeling (CHEMS)
IDYNEV Evacuation Model
Implementing Procedures
Journaling
MESORAD Model
Offpost Integration
Onpost Integration
Outdoor Sound Propagation Model (OSPM)
People Location
Personnel Management
Resource Management
Risk Analysis
Search and Rescue

These databases are identified in the following sections. Each section includes a database description, a list of applications, and a list of files that can be accessed or manipulated.

3.2.1 Database: D2 Dispersion Model

Description

The D2 database allows for dispersion modeling by planners, as well as accepting and displaying D2 output from the onpost system. In the planning stages, CnnnD2INP.DAT in the SITEID2: directory is used as input to the D2 model. The output from the model is placed in SITEOD2:. These output files include information on the dosage track and summaries on the dispersion used to create graphical representations of the track in the MAP$DATA$D2: directory.

Output from the D2 model can also be sent from onpost and placed in ONSITE$DIR:. The EVENT program uses this information to create a graphical representation of the dosage track that is placed in MAP$DATA$WD:.

Use

1. D2_(NO)GR
2. IBS -> EMERGENCY ACTIVITIES -> HAZARD ANALYSIS -> EXECUTE D2
3. EVENT

File Name	Directory
CASEINDEX.DAT	SITE$DIR:
CnnnD2INP.DAT	SITEID2:
D2_MASTER_FILE.DAT	SITEID2:
CnnnD2LOG.DAT	SITEOD2:
CnnnnDOSAGE.DAT	SITEOD2:
CnnnPLOT.DAT	SITEOD2:
CnnnPOSINP.DAT	SITEOD2:
CnnnSCRIPT.DAT	SITEOD2:
CnnnSLICE.DAT	SITEOD2:
CnnnD2.TXT	MAP$DATA$D2:
CnnnPLUME.DMS	MAP$DATA$D2:
LIMITD2.BIN	MAP$DATA$D2:
CnnnWD.DMS	MAP$DATA$WD:
CnnnWD.TXT	MAP$DATA$WD:
LIMITWD.BIN	MAP$DATA$WD:
D2LOGnnn.DAT	ONSITE$DIR:, OFF$FROMDIR:
DOSAGnnn.DAT	ONSITE$DIR:, OFF$FROMDIR:
SLICEnnn.DAT	ONSITE$DIR:, OFF$FROMDIR:
D2INPnnn.DAT	ONSITE$DIR:, OFF$FROMDIR:

3.2.2 Database: Damage Assessment

Description This database is used to define national defense boundaries and contaminated areas. This information is not currently used by the rest of the IBS system. .

Use

1. IBS -> EMERGENCY ACTIVITIES -> DAMAGE ASSESSMENT

File Name	Directory
KNOWN_POLYGONS.DAT	OFF$DIR:
LIMITEZ.BIN	MAP$DATA$EZ:
*.DMS	MAP$DATA$EZ:
*.TXT	MAP$DATA$EZ:
LIMITDA.BIN	MAP$DATA$DA:
DA.DMS	MAP$DATA$DA:

3.2.3 Database: Hazardous Chemical Modeling (CHEMS)

Description This database is used in hazardous chemical modeling to plan for small-scale events involving a variety of hazardous chemicals. These emergencies may include, for example, truck spills and explosions. Three different situations involving chemicals can be modeled: fire, vapor, and explosion.

Use 1. CHEMS_(NO)GR

File Name	Directory
CASEINDEX.DAT	SITE$DIR:
CnnnCHEMS.GEN	SITEICHEMS:
CRASHFLG.DAT	SITEICHEMS:
LIMITEX.BIN	MAPS$DATA$EX:
CnnnEXP.DMS	MAPS$DATA$EX:
LIMITFI.BIN	MAPS$DATA$FI:
CnnnFI.DMS	MAPS$DATA$FI:
LIMITVA.BIN	MAPS$DATA$VA:
CnnnVA.DMS	MAPS$DATA$VA:
xxxPROP.DAT	IEMIS$CHEMS:
CHEM.DAT	IEMIS$CHEMS:
LABEL.DAT	IEMIS$CHEMS:
PAT.DAT	IEMIS$CHEMS:
UNITS.DAT	IEMIS$CHEMS:
UNITSI.DAT	IEMIS$CHEMS:

3.2.4 Database: IDYNEV Evacuation Model

Description This database supports the IDYNEV evacuation model. It is used to simulate traffic patterns and to plan evacuation routes in the case of an event. This model can be used in conjunction with the MESORAD model to provide information on radiological dosages during an evacuation. Inputs for the IDYNEV model are found in SITEIDYNEV:; its outputs are placed in SITEODYNEV:.

Use

1. IDYNEV_(NO)GR
2. IBS -> EMERGENCY ACTIVITIES -> EVACUATION

File Name	Directory
CASEINDEX.DAT	SITE$DIR:
CnnnEVACPARAM.DAT	SITEIDYNEV:
CnnnINP.DAT	SITEIDYNEV:
CnnnLNK.DAT	SITEIDYNEV:
EVAC_SEARCH_HEADERS.DAT	SITEIDYNEV:
CnnnBIN.U36	SITEODYNEV:
CnnnCOD.RPT	SITEODYNEV:
CnnnCOD.U37	SITEODYNEV:
CnnnLNK.DAT	SITEODYNEV:
CnnnNOD.DAT	SITEODYNEV:
CnnnSUM.DAT	SITEODYNEV:
CnnnTV.DAT	SITEODYNEV:

3.2.5 Database: Implementing Procedures

Description This database consists of Implementing Procedures for emergency response and recovery. These procedures are defined before an event takes place and consist of tasks performed when certain situations arise. Tasks are composed of resources, actions, action notes, resource notes, and task notes. They can be used to step through an event and resolve it in an orderly fashion.

Use

1. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES

File Name	Directory
CASEINDEX.DAT	SITE$DIR:
SCEN_TABLE.DAT	SITE$IP:
SCEN_INDEX.DAT	SITE$IP:
CnnnACTIONS.DAT	SITE$IP:
CnnnACTION_NOTES.DAT	SITE$IP:
CnnnMASTER_CHECKLIST.DAT	SITE$IP:
CnnnMASTER_NOTES.DAT	SITE$IP:
CnnnRESOURCE_CHECKLIST.DAT	SITE$IP:
CnnnRESOURCE_NOTES.DAT	SITE$IP:
CnnnTASK_NOTES.DAT	SITE$IP:
CnnnTASK_LINKS.DAT	SITE$IP:

3.2.6 Database: Journaling

Description This database is used to maintain a history of messages arriving at and departing from the system, as well as to track changes in the operational database. Numerous reports can be created from this data.

Use 1. IBS -> MESSAGE BAORD

File Name	Directory
COMBINED.LOG	OFF$DIR:, SYS$LOGIN:
IMPMAIL.LOG	OFF$DIR:
IMPMAIL.TMP	OFF$DIR:
SPOCC_ACTIVITY.LOG	OFF$DIR:
SPOCC_ACTIVITY.LOG_DETAIL	OFF$DIR:
SPOCC_EVENT.LOG	OFF$DIR:
SPOCC_EVENT.LOG_DETAIL	OFF$DIR:
WATCH_EVENT.LOG	OFF$DIR:
WATCH_EVENT.LOG_DETAIL	OFF$DIR:
ANNOTATE.TMP	SYS$LOGIN:
RPTSUBJ.DAT	IEMIS$SYSF:

3.2.7 Database: MESORAD Model

Description This database is used in the MESORAD model for predicting radiological dosage levels over time and their location in the area of an event. It can be used to track cumulative dosages in a number of different fashions, including dosages in the lungs or over the whole body. The MESORAD model can be used in conjunction with the evacuation model, IDYNEV, to predict dosages received during an evacuation.

Use 1. MES_(NO)GR

File Name	Directory
CASEINDEX.DAT	SITE$DIR:
CnnnCHK.DAT	SITEIMESORAD:
CnnnOBS.DAT	SITEIMESORAD:
CnnnPOL.DAT	SITEIMESORAD:
CnnnRSP.DAT	SITEIMESORAD:
CnnnSnm.DAT	SITEIMESORAD:
CnnnSTA.DAT	SITEIMESORAD:
CnnnTER.DAT	SITEIMESORAD:
CnnnTOP.DAT	SITEIMESORAD:
CnnnV.DAT	SITEIMESORAD:
NUC.DAT	IEMIS$MESORAD:
PLUMELABS.DAT	IEMIS$MESORAD:
CnnnAIRDPL.ADV	SITEOMESORAD:
CnnnAIRDPL.CUM	SITEOMESORAD:
CnnnAIRNDP.ADV	SITEOMESORAD:
CnnnAIRNDP.CUM	SITEOMESORAD:
CnnnCONTOUR.DMS	SITEOMESORAD:
CnnnDEPOST.ADV	SITEOMESORAD:
CnnnDEPOST.CUM	SITEOMESORAD:
CnnnEXTGRD.ADV	SITEOMESORAD:
CnnnEXTGRD.CUM	SITEOMESORAD:
CnnnEXTSIC.ADV	SITEOMESORAD:
CnnnEXTSIC.CUM	SITEOMESORAD:
CnnnFLD.RPT	SITEOMESORAD:
CnnnGBRADV.BIN	SITEOMESORAD:

File Name	Directory
CnnnGDRADV.BIN	SITEOMESORAD:
CnnnING.RPT	SITEOMESORAD:
CnnnINHLNG.ADV	SITEOMESORAD:
CnnnINHLNG.CUM	SITEOMESORAD:
CnnnINHTHY.ADV	SITEOMESORAD;
CnnnINHTHY.CUM	SITEOMESORAD:
CnnnIODADV.BIN	SITEOMESORAD:
CnnnLOG.RPT	SITEOMESORAD:
CnnnPRM.RPT	SITEOMESORAD:
CnnnPUFFTR.BIN	SITEOMESORAD:
CnnnSTATIN.BIN	SITEOMESORAD:
CnnnTOTWBD.ADV	SITEOMESORAD:
CnnnTOTWBD.CUM	SITEOMESORAD:
CnnnWINDFD.BIN	SITEOMESORAD:
CnnnXmm.DAT	SITEOMESORAD:

3.2.8 Database: Offpost Integration

Description This database includes reports and maps sent from the offpost to the onpost system. It represents what the EOC believes to be taking place offpost. It includes such information as military and evacuation routes.

Use 1. IBS -> MESSAGE BOARD -> VIEW/CREATE/RESEND IBS -> ONPOST REPORT

File Name	Directory
generic.ATT	OFF$SENDDIR:
generic.DAT	OFF$SENDDIR:
generic.DLG	OFF$SENDDIR:
xx.ATT	MAP$DATA$xx:
xx.DLG	MAP$DATA$xx:
xx.DMS	MAP$DATA$xx:
KEY.DAT	OFF$SENDDIR:
XFERLIST.DAT	OFF$SENDDIR:
D2INPnnn.DAT	OFF$SENDDIR:
D2LOGnnn.DAT	OFF$SENDDIR:
DOSAGnnn.DAT	OFF$SENDDIR:
SLICEnnn.DAT	OFF$SENDDIR:
OFF_CAS.DAT	OFF$SENDDIR:
OFF_EVR.DAT	OFF$SENDDIR:
OFF_KP.ATT	OFF$SENDDIR:
OFF_KP.DAT	OFF$SENDDIR:
OFF_KP.DLG	OFF$SENDDIR:
OFF_KP.DMS	OFF$SENDDIR:
OFF_KPL.DAT	OFF$SENDDIR:
OFF_PA.DAT	OFF$SENDDIR:
OFF_PEVR.DAT	OFF$SENDDIR:
OFF_PIMR.DAT	OFF$SENDDIR:
OFF_RCD.DAT	OFF$SENDDIR:
OFF_RCR.DAT	OFF$SENDDIR:
OFF_SHLT.DAT	OFF$SENDDIR:
OFF_STAT.DAT	OFF$SENDDIR:
OFF_TC.DAT	OFF$SENDDIR:

3.2.9 Database: Onpost Integration

Description This database contains files sent from the onpost system to the offpost system and includes meteorological data, the current onpost work plan, casualty summary information, and data indicating what onpost personnel believe is happening. This information is accumulated and managed by the EVENT program which distributes the information appropriately. Although users do not typically manage these files, because they are distributed automatically, they can view the reports. All valid files are copied from the ONSITE$DIR: directory to the OFF$FROMDIR: directory by the EVENT program.

Use

1. EVENT
2. ONPOSTSIM

File Name	Directory
generic.ATT	ONSITE$DIR:, OFF$FROMDIR:
generic.DAT	ONSITE$DIR:, OFF$FROMDIR:
generic.DLG	ONSITE$DIR:, OFF$FROMDIR:
KEY.DAT	ONSITE$DIR:, OFF$FROMDIR:
MLUPDATE.DAT	ONSITE$DIR:, OFF$FROMDIR:
XFERLIST.DAT	ONSITE$DIR:, OFF$FROMDIR:
D2INPnnn.DAT	ONSITE$DIR:, OFF$FROMDIR:
D2LOGnnn.DAT	ONSITE$DIR:, OFF$FROMDIR:
DOSAGnnn.DAT	ONSITE$DIR:, OFF$FROMDIR:
SLICEnnn.DAT	ONSITE$DIR:, OFF$FROMDIR:
BASE_POP.DAT	ONSITE$DIR:, OFF$FROMDIR:
CALLLIST.DAT	ONSITE$DIR:, OFF$FROMDIR:
CAS_SUM.DAT	ONSITE$DIR:, OFF$FROMDIR:
ENVIRON.DAT	ONSITE$DIR:, OFF$FROMDIR:
KNOWNPTS.DAT	ONSITE$DIR:, OFF$FROMDIR:
NOTIFY.DAT	ONSITE$DIR:, OFF$FROMDIR:
SIGEVENT.DAT	ONSITE$DIR:, OFF$FROMDIR:
TOWERSEL.DAT	ONSITE$DIR:, OFF$FROMDIR:
WEATHER.DAT	ONSITE$DIR:, OFF$FROMDIR:
WORKPLAN.DAT	ONSITE$DIR:, OFF$FROMDIR:

3.2.10 Database: Outdoor Sound Propagation Model (OSPM)

Description This database is used to model the propagation of sound over terrain. Its primary purpose is to determine where warning sirens and other such devices can be placed to warn the population.

Use 1. OSPM_(NO)GR

File Name	Directory
xxxxEL.TAA	SITE$DIR:
xxxxEL.TAC	SITE$DIR:
xxxxELAT.TAA	SITE$DIR:
xxxxELAT.TAC	SITE$DIR:
xxxxEL_UTM.TAA	SITE$DIR:
xxxxEL_UTM.TAC	SITE$DIR:
CASEINDEX.DAT	SITE$DIR:
CnnnOSPM.DAT	SITEIOSPM:
CnnnOSPM.DMS	SITEIOSPM:
CnnnCONTOUR.DMS	SITEOOSPM:
CnnnDBA.DAT	SITEOOSPM:
CnnnDBC.DAT	SITEOOSPM:
CnnnOCTm.DAT	SITEOOSPM:

3.2.11 Database: People Location

Description This database is used for tracking the whereabouts of civilians during an event. During or after an event, civilians can check into hospitals or other facilities to receive assistance. The time they check in, their names, condition, and the name of the facility they enter are stored in this database. The database can then be used to determine where civilians are located and their condition. This database is primarily used by the Red Cross.

Use 1. LOCATE

File Name	Directory
FACILITY_TRACK.DAT	OFF$DIR:
PEOPLE_TRACK.DAT	SITE$OFF:
P_T_UPDATE.DAT	SITE$OFF:
LOCATE_PLA.TMP	SITE$OFF:
LOCATE_PLU.TMP	SITE$OFF:

3.2.12 Database: Personnel Management

Description This database contains information about emergency response personnel. During an event, this data can be used to designate the people to perform specific tasks during an event. It also provides a means of tracking emergency response personnel, based on their schedule and the date/time. Information on emergency response personnel must include their addresses, telephone numbers, responsibilities, and schedules.

Use

1. IBS -> PERSONAL MANAGEMENT
2. RMMGR

File Name	Directory
AGENCY_LOOKUP.DAT	OFF$DIR:
PERSONNEL.DAT	OFF$DIR:
POSITION_LOOKUP.DAT	OFF$DIR:

3.2.13 Database: Resource Management

Description This database is used to allocate and control equipment and material to be used in case of an event. Records of available resources and their locations are maintained in this database.

Use

1. IBS -> RESOURCE MANAGEMENT
2. RMMGR

File Name	Directory
FACILITY.DAT	OFF$DIR:
FACILITY_TRACK.DAT	OFF$DIR:
RESOURCE.DAT	OFF$DIR:
LIMITRM.BIN	MAP$DATA$RM:
RM.DMS	MAP$DATA$RM:

3.2.14 Database: Risk Analysis

Description

This database is used to define risk areas and describe the characteristics of populations in those areas. Such areas can be designated as either immediate response or protective action, depending on the distance from the event site, the evacuation routes, and the topography. Those areas designated as being in the immediate response category require immediate protective action in case of an event. After a risk area is identified and defined, population summary information for the risk area can be provided (such as estimates of total population within the risk area or special populations residing within that area).

Use

1. IBS -> EMERGENCY ACTIVITIES -> DIRECTION/CONTROL -> RISK AREA ANALYSIS

File Name	Directory
CR_RISK_AREA.DAT	SITE$OFF:
CR_RISK_AREA.DMS	SITE$OFF:
KNOWN_POLYGONS.DAT	OFF$DIR:
RP.DMS	MAP$DATA$RP:
LIMITRP.BIN	MAP$DATA$RP:
*.DMS	MAP$DATA$EZ:
LIMITEZ.BIN	MAP$DATA$EZ:

3.2.15 Database: Search and Rescue

Description This database is used to define areas in which search and rescue operations are taking place. This information is not currently used by the rest of the IBS system.

Use 1. IBS -> EMERGENCY ACTIVITIES -> SEARCH/RESCUE

File Name	Directory
KNOWN_POLYGONS.DAT	OFF$DIR:
*.DMS	MAP$DATA$EZ:
LIMITEZ.BIN	MAP$DATA$EZ:
SR.DMS	MAP$DATA$SR:
LIMITSR.BIN	MAP$DATA$SR:

Section 4
Procedures

Much of the general information needed to perform the following tasks has been presented previously in this manual. However, due to the complexity of the tasks, this information is detailed here in a specific procedural manner. This section describes certain critical tasks in an effort to simplify them for the user.

4.1 Generating a Site-Specific Map Database Design

For the IBS software to meet the specific emergency management needs for each site, it is usually necessary to generate a new site-specific map database. Although the map database delivered with the IBS software will be somewhat tailored for the site, some modifications and additions will be necessary. However, it probably will not be necessary to create an entirely new map database. The steps in creating a new map database are not difficult, but they do require planning and a systematic approach. The approach used in the following sections is top-down, comprising three major steps, to design, set up, and populate the map database:

Design Map Database

1. Determine need for a new map database
2. Identify areas of operation
3. Determine necessary topographies
4. Determine use of current map database topographies
5. Quantify need for raw data
6. Determine methodology for converting raw data.

Set Up Map Database

1. Create the new map database directory
2. Create the topography subdirectories
3. Generate the map database control files
4. Transfer attribute dictionary and icon files
5. Test the new map database structure.

Populate Map Database

1. Transfer topographies from the current map database
2. Acquire complete raw data set
3. Partition raw data into topographies

4. Partition large raw data sets into smaller ones
5. Convert raw data into DMS format
6. Place DMS files in appropriate topography directory
7. Partition large DMS files into smaller ones.
8. Create a limit file for each topography
9. Copy/create overview maps
10. Test the new map database.

The procedures for each step are described in Sections 4.2.3, 4.2.4, and 4.2.5.

4.1.1 Background Information

Before the DBAs can begin to design the map database, they must first know what data is available and what data is required. Section 4.1.1.1 (Obtaining Map Database Information) explains the sources of data and how to convert the data to DMS format. Section 4.1.1.2 (Required Map Database Topographies) explains why and which topographies are required and the sources of their data.

4.1.1.1 Obtaining Map Database Information

Two questions must be considered when identifying sources of map database information. The first is: "What resources (data formats and methods) can be used to generate or transfer geographic information to the system?" The second is: "What are the actual sources of data that can be transferred to a map database and that are available to each specific site?" With the knowledge of what sources and resources are available, plus the knowledge of which topographies are mandatory to provide IBS functionality, the DBA can develop a strategy for generating a map database.

Currently, seven sources of spatial data are available for a map database:

1. Existing IBS Geographic Topographies
2. USGS DLG Format
3. TIGER Line Format
4. 1990 Census Demographic and Housing Tables
5. ASCII Files
6. Vector Editing and DMS File Manipulation
7. USGS Digital Elevation Model (DEM).

Each method is described in the following subparagraphs. The operations and procedures for generating a map database are presented later in this chapter. The procedures for using the utilities and applications are described in the *IBS Utilities Guide*.

4.1.1.1.1 Existing IBS Geographic Topographies

The Federal Emergency Management Agency (FEMA) maintains a national map database consisting of data for all 50 states. This data is derived primarily from the

1:2,000,000-scale sectional maps of the national atlas distributed by the United States Geological Survey (USGS). The database also includes some 1:250,000-scale elevation (raster, not DMS) data.

The IBS topographies available through the national map database are shown in Table 4.1.

Table 4.1. Geographic Topographies and Topographic Codes

Topographies	Code
Administrative Boundaries	AB
Congressional Boundaries	CD
Cultural Features	CF
Electric Power Grid	EP
Features (Icons)	FT
GNIS (Geographic Names)	GN
HAZMAT Sites	HZ
Hypsography	HY
NAPB Targets	NA
Nuc Power Plants	NP
Political Boundaries	PB
Populated Place Names	PL
Population	PO
Railroads	RR
Roads & Trails	RT
Streams	ST
Water Bodies	WB

4.1.1.1.2 USGS DLG Format

USGS DLG format is currently the most effective method for importing geographic information into the map database. It is a *defacto* data transfer standard for many public domain and commercially available GIS.

The DLGXDMS import utility converts data from optional DLG format to DMS files.

The USGSXDMS utility converts 1:2,000,000-scale data from standard DLG format to DMS format.

The USGS100KXDMS import utility converts 1:100,000-scale standard DLG format transportation and hydrography data to the following DMS topographies:

100K	Roads	RD
100K	Railroads	RL
100K	Miscellaneous Transportation	MT
100K	Hydrography	HG

USGS 1:100,000-scale information is distributed in 1-degree geographic areas. Lists of the areas covered and their names are available at any USGS service center.

4.1.1.1.3 TIGER/Line Format

The TIGERXDMS utility converts data from TIGER Line format to DMS format. It provides 1:100,000-scale resolution topographies for roads, railroads, miscellaneous transportation, and hydrography (rivers and streams). This data is organized and distributed by counties, requiring the DBA to acquire data from all counties that intersect the desired area-of-operation. This utility is used to create the following topographies:

Addresses	(TIGER)	TA
Landmarks	(TIGER)	TF
Misc. Transportation	(TIGER)	TM
Railroads	(TIGER)	TR
Roads	(TIGER)	TT
Streams	(TIGER)	TS
Stream Names	(TIGER)	SN
Street Names	(TIGER)	TN
Water Bodies	(TIGER)	TW
Water Body Names	(TIGER)	WN

The TBXDMS utility creates DMS boundary data files from TIGER Line data files provided by the U.S. Department of Commerce, Bureau of the Census. The Bureau of the Census makes TIGER Line data available for individual counties. TBXDMS is designed to manage data from one county at a time, converting boundary data from the Basic Data Record (Record Type 1) and Shape Coordinate Points (Record Type 2) files. Other data records of the TIGER Line data files are ignored. This utility is used to create the following topographies, depending on the type of boundary data selected by the user:

Block	BB
Census Tract	TB
Place	LB
County Subdivision	VB
County	CB

4.1.1.1.4 1990 Census Demographic and Housing Tables

The POPXDMS utility converts 1990 Census Demographic and Housing Tables data to DMS point files. This utility generates topographies that provide population and housing information at the state, county, county subdivision, place, tract, group, and block levels.

4.1.1.1.5 ASCII Files

A geographic utility, ASCIIXDMS, will convert an ASCII DMX file to DMS format. It is the converse of the DMSXASCII utility. The utility has often been used to create DMS point files by editing output tables generated by DBMSs. The format of the DMX file is presented in Appendix A.

4.1.1.1.6 Vector Editing and DMS File Manipulation

The GIS supports an interactive vector editing capability. It also provides utilities that manipulate existing DMS data files. The vector editing capability in GEDIT can add, delete, or modify the geographic location, attributes, and graphical display of a spatial data object. Additionally, it will support the manipulation of text and icons associated with point objects. The use of GEDIT is described in detail in the *IBS Utilities Guide*. Although the vector editing capabilities of GEDIT can be used to digitize new topographies, this utility is used mainly to edit existing DMS files and convert lines into polygons to create boundary topographies. Topographies, such as Emergency Planning Zones (EZ) and APA Sectors (SC), are normally created in this manner.

DMS manipulation utilities also examine and modify attribute or graphical display information within an entire file. For example, COLORDMS will change the color of every object in a DMS file to a new user-selected color. ADDATTDMS is used to add, delete, or modify an attribute to every object in a DMS file. Additional utilities exist to merge two DMS files (APPENDDMS), to join lines that have the same endpoints and attributes (LINEAR_JOIN), and to reduce the number of geographic data points from high resolution objects (SMOOTHER and THINNER). These utilities and other DMS data management utilities are presented in the *IBS Utilities Guide*.

4.1.1.1.7 USGS Digital Elevation Model (DEM)

The DMS data items (for example, points, line segments, and polygons) are point-vector structures that are best used to represent 2-dimensional data, such as map features. DMS databases are not designed to contain cell data that represent regularly spaced values, such as elevation.

To provide the map display and modeling software with modest capabilities for dealing with elevation data, IBS supports limited use of Terrain Analysis Cell (TAC) data sets, which generally contain 1:250,000-scale elevation data. TAC data sets are further defined in Appendix A.

Geographic support for elevation data sets is usually limited to the following:

- The DEMXTAC utility program converts USGS Digital Elevation Model data to TAC format for 1:250,000-scale, 1-degree by 1-degree map sheets. (See *IBS Utilities Guide*.)
- The CONTOUR utility program converts TAC elevation data to a DMS file containing elevation contours (polygons). (See the *IBS Utilities Guide*.)
- The MPDISPLAY and GEDIT utilities can read a TAC elevation data set and plot the grid cells, which the user can pick to display or modify the corresponding elevation values. (See the *IBS Utilities Guide*.)
- The Map Display and GEDIT software libraries include subroutines that IBS modeling programs can use to read TAC data into a 2-dimensional array for program use. (For example, both the MESORAD model and the OSPM can utilize elevation data derived from TAC data sets.)

4.1.1.2 Required Map Database Topographies

Three classes of topographies are stored within the map database: mandatory, primary, and supportive. Mandatory topographies are required in order to perform the basic functionality of the IBS software. These topographies include transportation, elevation, EZs, and population. Primary topographies are those that directly impact the effectiveness and productivity of emergency managers. Such topographies include higher resolution transportation and population information via TIGER Line files and 1990 Census Demographic and Housing Tables, respectively.

Supportive topographies are those that provide additional information to the planning and response of emergency management, but are not critical in the development and use of an implementing procedure. These supporting topographies are generated by identifying a topography that would be helpful in supporting automated planning and response for the site.

4.1.1.2.1 Transportation

A transportation topography is mandatory to provide a first-cut approximation network for the evacuation model IDYNEV. Without this topography, interactive evacuation modeling cannot be performed. A default transportation topography is provided in the map database. Although this topography is 1:2,000,000-scale resolution and a bit outdated, it still will work. If possible, use 1:100,000-scale data provided by the U.S. Census TIGER Line files. The next best source is the USGS 1:100,000-scale DLG transportation files. (See Sections 4.1.1.1.2 and 4.1.1.1.3.)

4.1.1.2.2 Elevation

The elevation topography is required to perform radiological dispersion (MESORAD) and OSPM modeling and analysis. Converting the elevation data from USGS DEM format to the data structure (TAA, TAC) support by the IBS GIS is the only method for importing elevation data into the map database. (See Section 4.1.1.1.7.)

4.1.1.2.3 Emergency Planning Zones

The EZ topography containing the EPZ is mandatory. Without an EZ topography, the IBS cannot establish an emergency risk area or identify the appropriate Implementation Procedure (IP) for the event. Also, without the EPZs, users cannot identify the population at risk during an event. Editing boundary information provided with the map database with GEDIT is the most effective method for generating initial EPZs.

4.1.1.2.4 Population

A population topography is needed to provide input to the evacuation and other planning models. Without this topography, users cannot establish the population at risk during event notification and risk area analysis. The population provided with the map database contains total population information generated from the 1990 Census Demographic and Housing Tables. The census data conversion utilities can generate population and housing topographies that contain population, special population, and housing information. Because this data is based on census information, these population topographies best reflect nighttime and weekend population levels.

The planning community is still uncertain on how weekday population information will be acquired and stored. Because of this, the DBA will probably be responsible for acquiring and generating weekday population databases. Fortunately, (if sufficient source of population information is provided) it is easy to generate population topographies from tabular information using the ASCIIXDMS utility.

4.1.2 Design Map Database

Because creating a new map database is fairly complicated, users should have a good reason for creating one. With proper design, a map database will reflect its intended use and its information. Considering the proper design will help you clarify exactly what new information is needed for a map database. Therefore, before creating a new map database, take the following steps:

1. Determine need for a new map database.
2. Identify areas of operation.
3. Determine necessary topographies.
4. Decide use of current map database topographies.

5. Quantify need for raw data.
6. Determine methodology for converting raw data.

Each step is described in the following list:

1. Determine need for a new map database: Each site will require a map database that is tailored to contain site-specific information for emergency planning and response. A generic map database will be delivered with the IBS software, but some tailoring will probably be necessary.

2. Identify areas of operation: Determine the longitude-latitude extent of the new map database. This determination will allow a user to use map utilities to generate as much of the map database as possible. It will also be the area of interest that can be used as the base map when running the IBS.

3. Determine necessary topographies: Most often a new map database does not require 100% new data. A new map database generally consists of some preexisting topographies and a few new topographies. For many reasons, it is important to know what topographies should be included and how they will be used. First, the user wants the map database to contain applicable topographic information from existing map databases. Second, the user wants the map database to contain new site-specific topographic information. Third, the user does not want to include topographies that have no use or information appropriate for the site; they just take up space, waste processing time, and cause overhead charges. Fourth, considering the use of the topographies will allow the user to change an available topography's textual descriptor to fit the intended use of the new map database. Finally, the user must be sure the new map database topographies will contain all the information necessary to make the map database meet the emergency management needs of the specific site.

   To run the IBS applications, Elevation, Transportation, Population, and the Emergency Zone topographies must be included in the map database. (See Section 4.1.1.2.)

   To ensure that adequate background maps can be displayed, the following information is included in the map database: administrative and political boundaries and congressional districts; streams and water bodies; geographic names; populated place names; and features topographies.

   To facilitate effective display and analysis, the 1:100,000-scale roads, railroads, and hydrography data generated from TIGER line files should also be included. To support effective statistical analysis, all 56 population and housing topographies (Table 2.3), based on the 1990 Census Demographic and Housing Tables, should be included, as well.

4. Decide use of current map database topographies: The DBA determines which topographies in the current map database pertain to the new map database. During the population phase, the identified topographies will be transferred from the map database (using the MAPDBGEN utility) to the new map database.

5. Quantify need for raw data: Very rarely will a user construct a new map database where all the information is previously stored in an existing map database. At this time, the DBA must identify the source and transfer method to be used to populate an existing or new topography. Sources and data import utilities exist for all of the mandatory and primary topographies for the IBS. Presumably, most of the information to be placed in the new map database will be available in a supported transfer format. However, sometimes information for support topographies will not be in a form that is directly convertible into the DMS format. It is very important to know what information the raw data represents, where it is located, and how much data exists. It is also important to know how each new topography will be used to ensure that all of the necessary information (attributes) are included in the DMS files when they are created.

6. Determine methodology for converting raw data: Before proceding, the user must know how to turn raw data into DMS files. Converting the raw data to DMS format is probably the most difficult and tedious part of creating a site-specific map database.

   Many utilities exist for converting raw data to/from DMS format. The utilities include conversions for USGS DLG, TIGER Line, 1990 Census Demographic and Housing Tables, and ASCII files. For further information about converting data to DMS format, refer to Section 4.1.4, paragraph 5.

4.1.3 Set Up Map Database

When you have described the map database and the data conversion path, you are ready to set up the initial map database structure—that is, the database without the data. The steps involved in setting up the database structure are as follows:

1. Create the new map database directory.
2. Create the topography subdirectories.
3. Generate the map database control files.
4. Transfer attribute dictionary and icon files.
5. Test the new map database structure.

Each step is explained in the following paragraphs.

1. Create the new map database directory: This directory is the top-level database directory that will contain all the site's map database.[a]

   a. Enter $ **CREATE/DIRECTORY** *Disk:*[*newmap*]

      Substitute for *newmap* the new directory name established for your site specific map database. The DBA chooses a name that relates to the depot or site to be supported. For example, suppose that you want to create your new map database directory relating to **TOOE** on **DISK1**:

      $ CREATE/DIRECTORY DISK1:[TOOE]

   b. Enter $ **SET DEFAULT** *Disk:*[*newmap*]

      Again, substitute the name you used previously for *newmap*. This action sets the default directory to your new map database directory. All map database files will reside in or under this directory. All further steps in this map database setup procedure are performed with the new map database directory as the default directory.

2. Create the topography subdirectories: For each topography you want in your new map database, you must create a subdirectory. For each topography,

   $ CREATE/DIRECTORY [.*xx*]

   where *xx* is a 2-character code for the topography. (For example, to create a directory for a Mountain Range topography,
   $ CREATE/DIRECTORY [.MR].) As a minimum, create directories for the mandatory and primary topographies.

3. Generate the map database control files: Three control files are required for each map database: SETUPDB.COM which assigns logical variables to point to the map database; TOPOCTL.FIL which establishes the areas-of-operation; and TOPODESC.FIL which lists all the topographies in the map database. Use the FIXMAPDB utility to create these three files. For more information, refer to the *IBS Utilities Guide*.

4. Transfer attribute dictionary and icon files: Because the IBS applications and geographic databases use standardized attributes and icons, the DBA does not have to generate these files. Instead, the DBA can copy the dictionaries from an

(a) MAPDBGEN can be used to create the top-level directory and all subdirectories for topographies you intend to copy to the new map databases from the old map database. Just specify the database name and extent of 0,0,0,0.

existing IBS map database. The DBA creates the subdirectories needed to hold the attribute dictionary, the icon dictionary, and the log files.[a]

$ CREATE/DIRECTORY [.ATT]

$ CREATE/DIRECTORY [.ICON]

$ CREATE/DIRECTORY [.LOG]

The DBA copies the attribute dictionary from the current map database.

$ COPY MAP$ATTRIBUTE$DICTIONARY [.ATT]ATTRIBDIC.BIN

It is important to remember the entire icon directory needs to be copied to provide the icon dictionary and its support DMS icon files.

$ COPY/LOG MAP$ICONS:*.* [.ICON]*.*

When this is complete, you have an initial map database structure ready to be tested.

5. <u>Test the new map database structure</u>: As the final step in setting up the database structure, you must test to make sure your map database structure is correct. Executing SETUPDB.COM will define the access paths to your database. Follow these steps:

   a. Enter $ **@SETUPDB** or *newmap***DATABASE** to execute SETUPDB.COM.

   If you see no screen messages, proceed to Step (b). You may see messages telling you the definitions of some system logical names (the ones included in SETUPDB.COM) have been superseded. This fact indicates that MPDISPLAY-based programs will now access your new map database, rather than the original map database. If you get error messages instead of superseded messages, you need to examine SETUPDB.COM for errors. If you see a message indicating a *logical name table* is full, contact the system manager.

   b. Enter **DIR MAP$DATA$MAIN:**

   MAP$DATA$MAIN is the logical name that now refers to your top-level database directory. You will see a directory listing on the screen of all the files in your map database top-level directory. These files include

(a) If you use MAPDBGEN to create the database, the attribute dictionary, icon files, and log directory are copies or created for you.

TOPOCTL.FIL, TOPODESC.FIL, SETUPDB.COM, and the DIR (directory) files for each topography.

c. Enter **TYPE MAP$CONTROL:**

MAP$CONTROL is the logical name that now refers to your map database control file (TOPOCTL.FIL). You will see a listing of the contents of TOPOCTL.FIL on the screen.

d. Enter **TYPE MAP$TOPOGRAPHY:**

MAP$TOPOGRAPHY is the logical name that will be assigned to your topography description file (TOPODESC.FIL). You will see a listing of the contents of TOPODESC.FIL on the screen.

e. For each topography listed in TOPODESC.FIL, enter **DIR MAP$DATA$*xx:*** where *xx* is replaced by the 2-character topography code. This action causes an attempt to display directory listings of each topography subdirectory. You will see a message, indicating no files found in each directory because the directories are empty at this time. As long as you do not get an error, everything is O.K. If you get a *directory not found* error message, then either the logical variable is incorrect or the topography subdirectory was not created correctly: correct ASS_MAP_LOG.COM or create the correct subdirectory as needed.

f. After the map database structure checks out correctly, enter ***oldmap*DATABASE** where *oldmap* is the name of your original map database. The system is then reset to use the original map database, instead of your new map database.

If you were able to see or list the files and directories of the new database, then the structure of the site-specific map database was correctly defined. It is now time to populate the empty topographies in the new map database.

4.1.4 Populate Map Database

After the structure of the new map database has been created, your next step is to produce the proper DMS files to fill each new topography. This process usually involves conversion of raw (non-DMS) data files into DMS files.

Use the following steps to manipulate the topography data and put it into your map database:

1. Transfer topographies from the current map database.
2. Acquire complete raw data set.
3. Partition raw data into topographies.
4. Partition large raw data sets into smaller ones.

5. Convert raw data into DMS format.
6. Place DMS files in appropriate topography directory.
7. Partition large DMS files into smaller ones.
8. Create a limit file for each topography.
9. Copy/create overview maps.
10. Test the new map database.

These steps are explained in more detail in the following procedure. Before you start, remember to reset the pointers to the original map database by entering
$ *oldmap*DATABASE.

1. Transfer topographies from the current map database: The current map database will contain some of the topographies you want in the new map database. These include some of the mandatory and primary topographies. Use the MAPDBGEN utility to transfer, as a minimum, the mandatory and primary topographies, together with any others you want. For more information on MAPDBGEN, refer to the *IBS Utilities Guide*.

2. Acquire complete raw data set: Make sure that all the non-DMS data you need are available. This includes both the geographic data needed to locate graphic items and the information needed to describe those data items. Much of the power of any map database is in the attributes associated with each graphic item. (Attributes are discussed in Section 2.) Be sure this data exists and is obtainable. Adding data in later stages may require extra work.

3. Partition raw data into topographies: By splitting the raw data into multiple files and removing unwanted or unused data, the IBS GIS will execute more efficiently and faster. Depending on what form your data takes, you may have to manipulate files or create programs that will break your raw data files into appropriate topographic files or sets of files. If you are familiar with the format of your data, this action can save time and trouble by not requiring you to use IBS map utilities later to partition and recreate topographies.

4. Partition large raw data sets into smaller ones: This process reduces each file to a size that is manageable for data conversion and appropriate for a DMS file. Then, if conversion errors occur, smaller files will save reprocessing time.

5. Convert raw data into DMS format: This action involves converting the raw data files to DMS format. For each data file, run the appropriate map utility to convert the data file into a DMS file(s).

   Some raw data file formats do not have a map utility and will need conversion to an intermediate format before they can be converted into a DMS file. Typically, raw data file formats are converted to USGS Digital Line Graph (DLG) format or DMX (a special ASCII version of DMS).

The utilities that support conversion of data to DMS format are described in the *IBS Utilities Guide* and include the following:

Utility	Raw Data File Format
USGSXDMS	• USGS 1:200,000,000-scale standard DLG
USGS100KXDMS	• USGS 1:100,000-scale standard DLG
TIGERXDMS	• U.S. Bureau of Census TIGER Line Format
POPXDMS	• U.S. Bureau of Census 1990 Census Demographic and Housing Tables
DLGXDMS	• USGS 1:24,000-scale optional DLG USGS 1:100,000-scale optional DLG USGS 1:2,000,000-scale optional DLG
ASCIIXDMS	• ASCII (DMX)

6. <u>Place DMS files in appropriate topography directory</u>: If the conversion utility did not place the new DMS files in the appropriate topography directory, you must do so by using the VMS RENAME or COPY commands. For example, if you have created an airline routes file with the topographic code AR, place it into the [.AR] subdirectory that you prepared when you set up the new map database structure.

7. <u>Partition large DMS files into smaller ones</u>: Use the SET DEFAULT command to set the default directory to the topography subdirectory and enter **DIR/SIZE**.

   The screen will display the size (in blocks) of each file in your topography subdirectory.

   If any newly created DMS files is over 500 blocks, it is a strong candidate for further partition. If that is the case, then complete the following:

   a. Thinking of the file size as a 2-dimensional array of blocks, determine how the array could be split to yield smaller files of a more manageable size. For example, if the DMS file size is 1200 blocks, it could be split into a 2x2 array containing four DMS files of 300 blocks or perhaps a 2x3 array containing six blocks of 200.

b. Use the DMSXASCII utility to list the file header information from the too-large DMS file. This header information includes the longitude-latitude extent (maximum long-lat, minimum long-lat) of the topography area. Record these extent carefully.

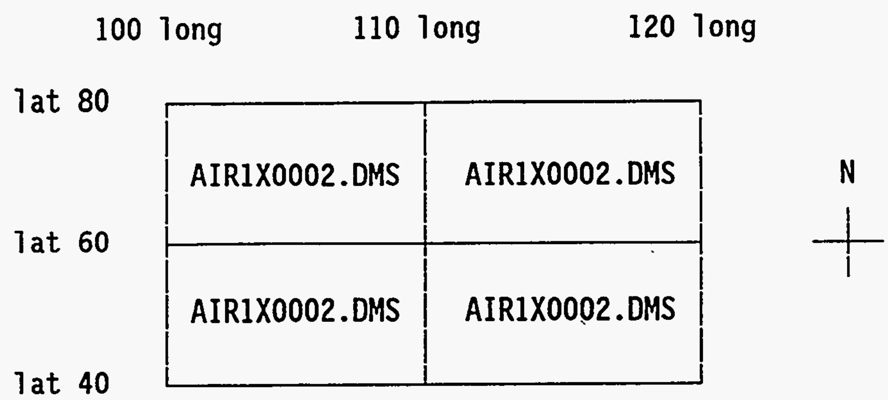
c. Run the CLIPDMS utility to divide the DMS file into a set of smaller DMS files.

For example, suppose that you have a 1200-block topography file named AIR1AR.DMS whose extent are 100 to 120 degrees longitude and 40 to 80 degrees latitude. You might use CLIPDMS to split AIR1AR.DMS into four smaller files, as follows:

$ CLIPDMS
Is a single area to be selected > **NO**
Enter Input filename > **AIR1AR.DMS**
Enter 5-character File Designator > **AIR1X**
Enter Starting Longitude, Latitude > **100, 40**
Enter Longitude, Latitude step sizes > **10, 20**

Your first NO response tells CLIPDMS that you want to split a file into several smaller-area files. CLIPDMS will use the 5-character File Designator you enter as the basis for naming the smaller files. Based on this input, CLIPDMS will start at 100 degrees longitude, 40 degrees latitude (the southwest corner of the region represented by the data). The first region clipped would be from 100 to 110 degrees longitude (a longitude step size of 10) and from 40 to 60 degrees latitude (a latitude step size of 20). The data for this region is stored in a file named AIR1X0001.DMS.

For each longitude step (west to east), CLIPDMS processes each latitude step (south to north) before moving on to the next longitude step. The second region clipped would be from 100 to 110 degrees longitude and from 60 to 80 degrees latitude. It is stored in a file named AIR1X0002.DMS. In this case, CLIPDMS has created four files named AIR1X0001.DMS, AIR1X0002.DMS, AIR1X0003.DMS, and AIRX10004.DMS. On a straightforward coordinate map, the region represented by data in AIR1.DMS could be represented like this:



Note these files do not actually represent equal areas because the latitudinal distances between two longitudinal lines decrease with distance from the equator. The size of the files may also vary because portions of the original file are denser (more data rich) than others.

8. <u>Create a limit file for each topography</u>: For every topography subdirectory you must create a limit file, named LIMIT*xx*.BIN, where *xx* is the 2-character code of the topography. This binary file specifies the names and geographic extent of all topography files that make up the topography in that subdirectory.

   For each topography subdirectory of your database, complete the following:

   a. Use the SET DEFAULT command to move to the directory above the topography subdirectory.

   b. Use the LIMITXX utility to create the LIMIT file. Refer to the *IBS Utilities Guide* for information on LIMITXX.

   You may also use the DOLIMITXX utility to do the entire map database at one time: $ *newmap*DATABASE; $ DOLIMITXX M; and $ *oldmap*DATABASE.

9. <u>Copy/create overview maps</u>: An overview map is displayed when you pick a database area to access at the beginning of a session with MPDISPLAY. Keep these points in mind:

   - Each overview map file adequately represents its database area of interest but contains as little data as possible so it will be displayed quickly.

   - Maps of each state, the continental United States, the United States plus Canada, the eastern seaboard states, and a world map of coastal outlines may exist in the map database.

- You can use the EXTRACT utility to extract useful overview map data from the map database. For more information on EXTRACT, refer to the *IBS Utilities Guide*.

- After you have copied or extracted a DMS file, you can always use GEDIT to modify the overview map for your purposes.

- If you create the overview map from several sources, you can combine DMS files by using the APPENDDMS utility to append one DMS file to another.

- After you have drawn or assembled an overview map, you can view it quickly by using the PLOTDMS utility. After creating the overview maps, you must update the TOPOCTL.FIL, either by editing the file directly or by using the FIXMAPDB and creating only a new topography control file.

10. Test the new map database: In other words, determine whether it all works:

    a. Set the default to your top-level map database directory.

    b. Enter *newmap*DATABASE or SYS*newmap*DATABASE to execute SETUPDB.COM and assign the logical names to refer to your new map database.

    c. Enter **MPDISPLAY** to run MPDISPLAY.

    If the TOPOCTL.FIL works correctly, MPDISPLAY will display a menu of the database area(s) listed in TOPOCTL.FIL.

    If you listed only one database area in TOPOCTL.FIL, only one menu item will be displayed.

    d. Pick each database area to display the overview map of the area.

    e. Pick a region on each overview map for display, and pick topographies for display within the region.

    If the topography data was entered into the database properly, MPDISPLAY will correctly plot the chosen topographies in the chosen region of your new map database. If instead an error occurs, recheck to make sure that a limit file exists in each topography directory and the overview map DMS file(s) in the top-level directory is properly named.

If everything works correctly, congratulations! You have just created a map database. At any time, you can select the map database by entering *newmap*DATABASE. When you are finished using the database, enter *oldmap*DATABASE to reset the original map database for access.

4.2 Creating User-Defined Icons

An icon is a graphical figure that can be associated with a particular data point (location). The graphics data for an icon are stored in a data file, referred to as an icon file. An icon file is just a standard DMS graphics data file—the kind created by using Graphics Edit (GEDIT). This discussion describes a procedure for creating personal icons and using them in various graphics display programs. Such icons are referred to as user-defined icons.

4.2.1 User-Defined Icons

When the graphics programs (for example, GEDIT and Map Display) manipulate icons, they must refer to an icon control file. The icon control file contains the current list of available icons and the icon files that are needed to plot each icon (See Section 4.2.3). The graphics programs normally refer to a system default icon control file, which designates a standard set of icons such as those used by IBS. To utilize the user-defined icons, create a different icon control file that refers to the icons—a user icon control file. This file must be located with the user-defined icon files in a user icon directory that you create.

Finally, the graphics programs must access the user icon control file and user icon directory, rather than the system default icon control file and icon directory. It is easy to do because the graphics programs refer to the current icon control file and icon directory through logical names, rather than through specific file and directory names. The logical name MAP$ICON$CTL: normally refers to the system default icon control file, and the logical name MAP$ICONS: normally refers to a system icon directory. The following procedure describes how to assign these logical names so they temporarily refer to the user icon control file and icon directory.

This procedure for creating user-defined icons takes place outside of IBS, at the operating system level. Before proceeding, you must be able to edit files using EDT, the system text editor, and be familiar with VAX DCL and file handling commands.

4.2.2 Procedure for Creating User-Defined Icons

Six general steps are used to create user-defined icons:

1. Create a user icon directory to hold the user-defined icons and a user icon control file.

2. Draw the icons using the graphics editor, generating the DMS files for the user-defined icons.

3. Create the user icon control file that will designate the user-defined icons for the graphics programs.

4. Copy into the user icon directory the DMS files for any system icons that will also be required.

5. Assign the icon logicals so the graphics programs will refer to the user icon control file, rather than the system default icon control file.

6. Test to be sure the user-defined icons work correctly.

These steps are explained in more detail in the following procedure:

1. Create a User Icon Directory: You must create a directory to hold the user-defined icon files and the user icon control file. This directory holds all user icon-related files in one location where they can be referred to by the graphics programs. The following procedure assumes the directory you are creating will be a subdirectory of the directory in which you begin the procedure:

   a. Enter **SHOW DEFAULT**

      The system will display the complete specification of the directory where you are beginning this procedure. This step is to check that you are starting from the correct (desired) directory. Although you can create the icon subdirectory anywhere, for consistency create it in the SITE$MAPDIR: directory.

   b. Enter **CREATE/DIRECTORY [.ICON]**

      This command will create a subdirectory with the name ICON. Assuming your current directory is SITE$MAPDIR, and your user account is located on DISK1, this creates a subdirectory SITE$MAPDIR:[ICON], with the complete path name of DISK1:[*user*.SITES.*site*.INPUT.MAPS.ICON].

   c. Enter **SET DEFAULT [.ICON]**

      This sets the default directory to the user icon directory. Files that are created later will then be created in this directory. All further steps in this procedure are performed using the user icon directory as the default directory.

2. Draw the Icons: To draw an icon, you must use a special input file that is created by copying the BLANKICON file into your directory. BLANKICON is a DMS file that contains the graphical depiction of a square approximately the size of the other icons. When you draw your icon, you must use the copy of BLANKICON as the first input file. When you exit GEDIT, the icon data are added to the file, which becomes an icon file. For each icon, follow these steps:

   a. Enter **COPY BLANKICON *filename*.DMS**

(In place of the *filename*, substitute a new name that describes the icon you will draw. Retain the .DMS file name extension.) This step creates an empty icon file with that name in your directory.

b. Enter **GEDIT**. (For more information on how to use GEDIT, refer to the *IBS Utilities Guide*.)

c. When the system prompts you to ENTER DMS FILE NAME >, enter the name of the file you created in Step 2(a): ***filename*.DMS**.

   This file is the only one you will edit. When the system prompts for another input file, press <CTRL Z> to stop entering input file names. A small red-bordered square appears in the center of the display. This square is about the size of icons that others have used.

d. Use the ZOOM IN function to enlarge the square to fill (approximately) the display area.

e. Use any of the drawing and editing functions to create (draw) the icon.

   The center point of the square is the location point of the icon. If the icon is centered on this point, it will be centered on the icon seed point when the icon is used later. You may draw outside the boundaries of the square; it is not meant to be a size restrictor.

f. When you are finished drawing the icon, use the DELETE COMPOSITE STRING function to delete the square.

g. Pick RESTORE BASE PLOT to zoom out.

   This command displays how the icon size would look like on a *real* plot. Icons are always plotted at this relative size on the screen.

h. Pick EXIT.

   GEDIT will save your icon file under the name you used in Step 2(a).

i. Repeat Steps 2(a) through 2(h) to create additional icons.

j. Make sure that every icon you are going to use is in your user icons directory. If you are using icons from other directories, copy the DMS icon files to your icon directory.

3. Create the User Icon Control File: You must now create an icon control file. This file will refer to your icons rather than the system default icons. The easiest way to create this user-defined file is to copy and edit an existing icon control file. To make the icons available to everyone, the DBA should edit the system icon

control file, MAP$ICON$CTL, and proceed to Step 6. If these icons will not be available to all, follow these steps:

a. Enter **COPY MAP$ICON$CTL USERICON.CTL**

   This command creates a copy of the system default icon control file in your directory. It is especially helpful when you want to use some standard system icons, as well as your new icons. The user icon control file in your directory will be named USERICON.CTL. If you have previously created a different user-defined icon control file, you can copy that file instead of the system default file.

b. To begin editing the icon control file, enter **EDIT USERICON.CTL**. The icon control file contains one record for each icon in the file. For example, the following icon control file has three records (one record per line) for three icons:

7 0110 AIRPORT	MAP$ICONS:AIRPORT.DMS	F
7 0115 MILITARY AIRPORT	MAP$ICONS:MAIRPORT.DMS	F
8 0010 CHURCH	MAP$ICONS:CHURCH.DMS	F

   Each record contains (in order) a major attribute code, a minor attribute code, an icon descriptor, an icon file name, a facility/resource code, and resource ring position. The attribute codes are special numeric values associated with particular icons; attribute codes are explained in Appendix B of the *IBS Utilities Guide*. The icon descriptor is the name that appears on the icon menu lists. The icon file name is the name of the DMS file that contains the graphic data for the icon. The facility/resource code is an F for a facility, an R for a resource, and blank otherwise. If it is a resource, you can also specify the ring position for the icon around a facility.

   The purpose of editing an existing icon control file is to start with a file already set up in the proper format; that is, each item in each record is in the proper order and column position for use by the graphics programs (such as GEDIT or Map Display). (See Appendix B or Section 4.2.3.)

c. For each icon you have created, edit one record of the icon control file by changing the attribute codes, icon descriptor, icon file name, and facility/resource code.

   For example, suppose you have created an icon for a train station in an icon file named STATION.DMS. You might edit the first line of the icon control file listed previously to read:

6 0110 TRAIN STATION	MAP$ICONS:STATION.DMS	F

Notice the attribute codes have been changed to 6 and 0110. This coding corresponds to a railroad station in the attribute dictionary (see Appendix B of the *IBS Utilities Guide*, IBS Attribute Dictionary). Notice also that directory name MAP$ICONS: is retained as the front portion of the icon file name.

As you edit a record, be careful NOT to change the starting column positions of items in a record. Use spaces between items, NOT tabs.

d. In the user icon control file, make sure that a complete record exists for each user-defined icon.

If you want to use any of the system default icons, include them in your user icon control file.

e. When you have completed the changes to the control file, exit the editor by pressing <CNTL Z> and entering **EXIT** at the command line to save the changes.

4. Copy Any Necessary System Icons: If you wish to use some of the system default icons, you must copy the corresponding icon DMS files into your icon directory. For each system icon still listed in your user icon control file:

   - Enter **COPY MAP$ICONS:*icon-name*.DMS *.***

   (Replace *icon-name* with the actual name of one icon.)

   For example: **COPY MAP$ICONS:EOC.DMS *.*** would copy the EOC icon.

5. Assign Logicals to Point to the User Icon Control File: Now tell the system to refer to the user icon control file, rather than the system default icon control file. This procedure involves changing the values of the MAP$ICON$CTL: and MAP$ICONS: logicals to point to your icon subdirectory and icon control file, respectively. To do this, follow these steps:

a. Edit the command file, **EDIT SITE$DIR:SETUP.COM.**

At the end of the file, type in the following two lines (the $ sign must be in the first column):

$ ASSIGN [*icon-directory*]USERICON.CTL MAP$ICON$CTL

$ ASSIGN [*icon-directory*]MAP$ICONS

Replace the phrase *icon-directory* with the full path name of your user icon directory. The SITE$MAPDIR: logical can be used as follows:

$ ASSIGN/JOB SITE$MAPDIR:[ICON]USERICON.CTL MAP$ICON$CTL

$ ASSIGN/JOB SITE$MAPDIR:[ICON] MAP$ICONS

b. Exit the text editor.

A new version of SETUP.COM will be created, containing references to the user icon control file and directory.

c. Enter **SWAPSITE N** *site*, press return, and enter **SETENV**.

This executes the SETUP.COM command file. The first ASSIGN command associates your icon control file with the logical name MAP$ICON$CTL:. The second ASSIGN command associates the user-defined icon directory with the logical name MAP$ICONS:. If you have followed these procedures successfully, the graphics programs can now access the user-defined icons because they use the logical names MAP$ICON$CTL: and MAP$ICONS:.

6. Test the User-Defined Icons: Before running any of the programs, use the following tests to make sure all of the previous steps have been performed properly:

a. Enter **TYPE MAP$ICON$CTL**

This command should display the contents of the current icon control file, which should be your new user icon control file. If your user icon control file is displayed, that assignment is correct.

b. If your user-defined icon control file is displayed correctly, then

- Re-edit SETUP.COM to see if you have made an error.
- Repeat step 5.c to reassign the icon control file.
- Restart your testing at Step 6(a).

c. Enter **TESTICONS**

This command executes a special program that checks to see whether the logical name MAP$ICONS is correct and that all of the icon files listed in the icon control file exist. Each icon is tested, and if a problem is found, a status code, the attribute data, and the file name are printed. At the end, the program prints out the number of icons.

Because only errors are printed when any icon file is listed, either SETUP.COM is incorrect or the icon file does not exist (under that name) in your user icons directory. If you re-edit SETUP.COM, be sure to reassign the logical names by repeating step 5.c before retesting. If the tests run correctly, you should be able to use your user-defined icons in graphics display programs.

The user-defined icons will remain in effect until you sign off the system (logout) or you deassign the logicals MAP$ICON$CTL: and MAP$ICONS:. Changing sites will not effect the user icons. To deassign them, enter:

DEASSIGN/JOB MAP$ICON$CTL

DEASSIGN/JOB MAP$ICONS

4.2.3 Icon Control File

This information about the icon control file is not critical to routine operation of the IBS software. The information may be of interest to users who want to customize an icon control file for a particular application.

The icon control file provides information about each icon and must be available if icons are to be displayed, inserted, or deleted. This file is made up of sequential ASCII records and contains one record for each icon type to be included. Each record has the following format:

Columns	1-2	Major attribute code
Columns	4-7	Minor attribute code
Columns	9-33	Icon Descriptor (used for display only)
Columns	34-73	Name of the graphical description file (icon DMS file) where the data points are stored
Column	74	Facility/Resource type code.
Columns	75-78	Resource Ring Location.

The icon files that store the graphical description (data points) of an icon are in standard DMS format, and the data points are in delta screen inches. That is, the graphical description of the icon is in terms of screen distances offset from the icon seed point. Thus, the icon remains a constant size on the screen display, and location coordinates are associated only with the seed point, not the icon data.

Appendix A
IBS File Descriptions

This appendix provides a detailed description of many of the files present in the IBS system. Much of the detailed description of the interface files was taken, with permission, from the EMIS 2.04-IBS 2.04 Interface Control Document (ICD). The ICD was written by Innovative Energy Management, Inc., 7423 Picardy Ave., Suite E, Baton Rouge, LA 70808-4362.

Each file description in this appendix has the following format. Unless otherwise noted as optional, all fields of this field are present in each file description.

Filename: The name of the file.

Directory: The location of the file, typically a logical. Multiple entries indicate the file occurs in each directory location.

Purpose: An explanation of what the file is, and why it exists.

See Also: (Optional) Associated files.

Access: Actions that provide access (either read or write) to the data.

References: (Optional) Other related documents.

Include: (Optional) Name and location of include files that contain internal common blocks.

Notes: (Optional) Additional information about the file.

Structure: The structure of the file. The first descriptor is either ASCII or binary. The second descriptor is either sequential or direct access. The third descriptor is either fixed length or variable length. The fourth descriptor is the record length (for each record if fixed length, or the maximum length, if variable).

Record n: The format of record n. The first line is optional and contains the length of the record and a record description. If the record is formatted, the second line is a FORTRAN format statement that can be used to generate the record. Following the format statement is a list of the variables used in the format statement. The list of variables includes the variable name, internal type, and description.

Filename: *_timestamp.BCK

Directory: SITE$BCK:

Purpose: The BCK files contained in the SITE$BCK: directory are VAX/VMS Backup save sets of IBS data. The timestamp has the format YYMMDDHHMMSSHH.

Access:
1. IBS -> BACKUP/RESTORE SITE DATA
2. SITEBKP

Include: IEMIS$INC:PERMISSION.INC, IEMIS$INC:IBSETC.INC, SPOCC$INC:ARCHIVE.INC

Notes: The following table defines the save set file names and the files backed up in the save sets for each of the backup options.

Backup Option Save Set Name	Files Saved to the Save Set
Backup Resource Management data	
RM_timestamp.BCK	OFF$DIR:FACILITY*.DAT; RESOURCE.DAT; KNOWN_POINTS.DAT; *_LOOKUP.DAT; PERSONNEL.DAT; SPECIAL_POPULATION.DAT;
RM_DMS_timestamp.BCK	MAP$DATA$RM:*.*;
Backup D2 Case data	
D2_timestamp.BCK	SITEID2:C*.*; SITEOD2:C*.*; SITEID2:D2_MASTER*.*;
D2_DMS_timestamp.BCK	MAP$DATA$D2:*.*;
D2_NDX_timestamp.BCK	SITE$DIR:CASEINDEX.DAT (D2 indices only)
Backup IP Case data	
IP_timestamp.BCK	SITE$IP:C*.*;
IP_NDX_timestamp.BCK	SITE$DIR:CASEINDEX.DAT (IP indices only)
Backup EVAC Case data	
IDYNEV_timestamp.BCK	SITEIDYNEV:C*.*; SITEODYNEV:C*.*;
IDYNEV_NDX_timestamp.BCK	SITE$DIR:CASEINDEX.DAT (IDYNEV indices only)

Backup OSPM Case data	
OSPM_timestamp.BCK	SITEIOSPM:C*.*; SITEOOSPM:C*.*;
OSPM_NDX_timestamp.BCK	SITE$DIR:CASEINDEX.DAT (OSPM indices only)
Backup CHEMS Case data	
CHEMS_timestamp.BCK	SITEICHEMS:C*.*; SITEOCHEMS:C*.*;
CHEMS_NDX_timestamp.BCK	SITE$DIR:CASEINDEX.DAT (CHEMS indices only)
Backup MESORAD Case data	
MESORAD_timestamp.BCK	SITEIMESORAD:C*.*; SITEOMESORAD:C*.*;
MESORAD_NDX_timestamp.BCK	SITE$DIR:CASEINDEX.DAT (MESORAD indices only)
Backup Risk Area data	
RA_timestamp.BCK	OFF$DIR:KNOWN_POLYGONS.DAT;
RA_DMS_timestamp.BCK	MAP$DATA$RP:*.*; MAP$DATA$SR:*.*; MAP$DATA$DA:*.*;
Backup data Received	
FROM_timestamp.BCK	OFF$FROMDIR:*.*;
Backup data Sent	
SEND_timestamp.BCK	OFF$SENDDIR:*.*;
Backup All Site data	
SITE_timestamp.BCK	SITE$TOPDIR:[...]*.*; /EXC=[*...]*.BCK
OFFSITE_timestamp.BCK	OFF$MAPDIR:[...]*.*; /EXC=[*...]*.BCK

<ins>Structure</ins>: VAX VMS Backup format

Filename: *.ATT

Directory: MAP$DATA$xx:

Purpose: *.ATT files are DLG (Digital Line Graph) ancillary attribute files that are created together with a DLG file (see *.DLG). When DMS data must be transferred to the onpost system, the DMS data is first converted to DLG optional format (level 2) in a DLG file. The *.ATT file is the DLG ancillary attribute file which contains the IBS attribute codes for the DMS data. The ancillary attribute file also contains the DMS pen number and line type for the DMS data. The *.ATT files have the same file name as the corresponding DLG file but with an .ATT file extension. Because the IBS attribute codes are not transferrable to the DLG attribute coding system, IBS attribute codes are contained in an ancillary file.

See Also: *.DLG

Access: 1. IBS -> MESSAGE BOARD -> CREATE IBS TO ONPOST REPORT

Structure: ASCII, sequential, fixed length, 40 bytes/record

Record 1: 40 bytes, header
`FORMAT (A40)`

The file header contains the name (filename and extension) of the corresponding DLG file.

DLG_FILE	C*40	DLG filename and extension

Record 2: 40 bytes, attribute
`FORMAT (A1,I5,3I6,2I3,10X)`
Each line contains the DLG object type and id, DMS attribute code, color, and line type for each object in the DLG file. Each of the DLG data objects are identified by a unique id number that is linked to an attribute record.

REC_TYPE	C*1	DLG record type ('N' for node, 'L' for line)
NODE_ID	I*4	Unique DLG object id number
MAJOR	I*4	Major attribute value
MINOR	I*4	Minor attribute value
PARAM	I*4	Parameter attribute value
DMSPEN	I*4	DMS pen number (color)
DMSLINETYPE	I*4	DMS line type

Filename: ***.DIS**

Directory: SYS$LOGIN:, INFO$DIR:

Purpose: The DIS files are the VMS mail distribution lists. They contain a list of user names to whom mail may be sent. The files are processed by VAX/VMS mail.

See Also: MAILLIST.DAT

Access:
1. IBS -> MESSAGE BOARD -> MAIL
2. MLMGR
3. MAIL_LIST

Notes: Each user can have a private set of distribution files. The INFOMANAGER's distribution lists are used as the system distribution files. The mail list utility creates/modifies these files.

Structure: ASCII, sequential, variable record length

Record 1-n:

```
FORMAT (A80)
```

USER ID	C*80	Node and user name (i.e., KIRK::JOHNDOE).

Filename: *.DLG

Directory: MAP$DATA$xx:

Purpose: DMS files selected to be sent to the onpost system are actually converted to 1:100,000 scale DLG (Digital Line Graph) Optional format (level 2) files that are sent instead.

See Also: *.ATT

Access: 1. IBS -> MESSAGE BOARD -> CREATE IBS TO ONPOST RPT

References:
1. Digital Line Graphs from 1:100,000-Scale Maps Data Users Guide 2 U.S Geological Survey, 1985
2. Specification of DMS file format

Structure: ASCII, sequential, fixed length, 80 bytes/record. See DLG reference for DLG file format.

1. The DLG files created are level 2 format; therefore, data linkages are not present (node-line linkages, etc.). Basically, the DLG data should be interpreted as a set of X/Y points defining a series of node or line objects. Each of these objects may have DMS attribute codes associated with them.

2. The X,Y coordinate system in the DLG file is in UTM.

3. DMS points are converted to DLG nodes. DMS lines, polygons, and complex polygons are converted to DLG lines. No DLG area data is created.

4. DLG header fields that are applicable only for USGS maps (not applicable to DMS data) are left blank or are filled with zero values (for instance, map sectional indicator).

5. All DLG data belongs to one category. Data is not repeated or grouped according to data type classifications, such as transportation or hydrography.

6. All DMS attributes are contained in an ancillary attribute file (see *.ATT). No attribute codes are contained in the DLG file.

Filename: *.DMS

Directory: MAP$DATA$xx:

Purpose: The DMS file(s) contained in the MAP$DATA$xx: is the graphical representation of the topography, xx. For more information on DMS files and topographies, see Section 2 of this guide.

See Also: *.DMS, *.DMX, LIMITxx.BIN

Access: The filenames for the DMS file(s) are located in the LIMITxx.BIN file. Both the LIMITxx.BIN file and the DMS file are read during the initialization phase of most graphical IBS programs.

Include: MAP$DEVELOP$INC:DMSIO.INC

Structure: DMS format (See Appendix C)

Filename: ***.DMS**

Directory: MAP$DATA$MAIN:

Purpose: The DMS files contained in the MAP$DATA$MAIN: directory are low-detail region overview maps. These DMS files are not used by IBS but by the program MPDISPLAY for providing an overview for the available map regions. The map region filenames and descriptions are contained in the topographic control file MAP$DATA$MAIN:TOPOCTL.FIL.

See Also: TOPOCTL.FIL

Access: The low-detail map region DMS files are not accessed by IBS, but rather by the program MPDISPLAY.

Structure: DMS format (See Appendix C)

Filename: *.DMS

Directory: MAP$ICONS:

Purpose: The DMS files contained in the MAP$ICONS: directory are IBS icons. Each icon is a symbol that can be drawn on the topographic map display to represent the location and availability of features such as facilities, resources, services, and physical landmarks. Icons can be grouped into three categories: facilities, resources, and nonfacility/resource icons. Facilities represent the location of facilities, such as schools, churches, and food distribution centers. Resource icons are used to depict a type of resource, such as communication equipment, food, and medical equipment. Nonfacility/resource icons represent items that do not fall within the facility or resource categories.

See Also: ICON.CTL, BLANKICON.DMS

Access: The filenames for the icon files are located in the icon control file, ICON.CTL. Both the ICON.CTL file and the filenames in ICON.CTL are read during the initialization phase of most graphical IBS programs.

Include: MAP$DEVELOP$INC:MAPROOT.INC

Structure: DMS format (See Appendix C)

Filename: *.DMX

Directory: MAP$DATA$xx:, MAP$DATA$MAIN:, MAP$ICONS:

Purpose: An ASCII version of the *.DMS file the user can edit. It can be converted from the DMS format with the DMSXASCII utility, and back to DMS format with the ASCIIXDMS utility. The DMX files are not used by the IBS system.

See Also: *.DMS, *.TXT

Access:
1. DMSXASCII
2. ASCIIXDMS

Include: MAP$DEVELOP$INC:DMSIOVAR.INC

Notes: A DMX file contains one or more graphics objects. The data for each graphics object is contained in multiple lines. Because of the complexity of the record number expression, record numbers are identified from the beginning of the object, not the beginning of the file.

Structure: ASCII, sequential, variable length.

Record 1: Header record
FORMAT (A16)

TXT_FILE	C*16	TXT filename, if used

Record 2: Graphics object header record
FORMAT (*)

NATT(I)	I*1	Number of attributes
NPEN(I)	I*1	Pen color number
LINTYP(I)	I*1	Line type
PLTFLG(I)	I*1	Type of graphic object 0 - Line 1 - Point 2 - Simple polygon 3 - Complex polygon 4 - Point
NP(I)	I*2	Number of points
OBJ(I)	I*4	Object Number

Record 3 for NATT(I): Attributes
FORMAT (*)

MAJOR	I*4	Major attribute code
MINOR	I*4	Minor attribute code
PARAM	I*4	Attribute parameter

Record NATT(I)+3: Text attributes, if any

FORMAT (*)		
NLINE	I*2	Number of lines of text
(NBYT_LINE(J), J=1,5)	I*2	Number of bytes for each line
CHR_SIZE	R*4	Text size, in inches
CHR_ROTAT	R*4	Text rotation angle, in degrees
CHR_FONT	I*2	Font
TEXT_SCALE	R*8	Text scale

Record NATT(I)+4 for NLINE: Text data

FORMAT (A)		
INREC	C*80	Text line

Record NATT(1)+NLINE+4 - ONLY if a Complex Polygon: Polygon definition

FORMAT (*)		
POLY_D(I)	I*2	Number of sub-polygons that make up the complex polygon (max 13)

Record NATT(1)+NLINE+5 - ONLY if a Complex Polygon: Sub-polygon definition

FORMAT (*)

(POLY_D(1+K), K=1, POLY_D(I))

Starting position in the following list of point data for the Kth sub-polygon. If the value is negative, the sub-polygon is excluded (i.e., the hole in a donut).

Record NATT(1)+NLINE+4(+2 if a Complex Polygon) for NP(I): Point data

FORMAT (*)		
XD(J)	R*8	X data
YD(J)	R*8	Y data

Filename: *.TAA

Directory: MAP$DATA$xx:

Purpose: Raster data file - the *.TAA file stores the raster data for an area. Raster data consists of a value (i.e. elevation) for a cell in a Cartesian grid. The TAA file name, coverage area, cell size, and other information is specified in the associated *.TAC file. The most common occurrence of raster data is elevation which has been converted from a USGS DEM file.

See Also: *.TAC

Access: The filenames for the *.TAC files are located in the LIMITxx.BIN file. The LIMITxx.BIN and TAC files are read during the initialization phase of most graphical IBS programs when the topography is selected. The .TAA file is only used in parts of MPDISPLAY and GEDIT, though you can show a cell value in IBS Map Analysis.

Include: MAP$DEVELOP$INC:TACIOVAR.INC
MAP$DEVELOP$INC:TVCIOVAR.INC

Notes: The .TAA file stores cell data values in rows from minimum to maximum along the X (east-west) axis. The rows are ordered from minimum to maximum along the Y (north-south) axis within the plane. Therefore the record length is determined by the number of columns per row (that is, each row in the matrix is equal to one record in the file). The size of the individual cell data values are determined by CFILEA_DCR(5) of the TAC file.

Under VMS, the length of each record in a binary, direct access file must be a multiple of 4 bytes. Therefore, a TAA file with 5 columns of I*2 data (10 bytes) will actually have an extra 2 bytes of padding at the end of each record. VMS does this automatically and it is transparent to the IBS code. However, if the file were moved to another operating system, the record padding would need to be removed.

Structure: Binary, direct access, fixed length

Filename: *.TAC

Directory: MAP$DATA$xx:

Purpose: Raster data header file - the *.TAC file stores the raster data header for an area. Raster data consists of a value (i.e. elevation) for a cell in a Cartesian grid. The .TAC file is a direct-access file that contains information defining the range, type, format, scale, and attribute status of the data and the beginning record number and word count of each row of data. The actual raster data is stored in a TAA file pointed to by the TAC file. The most common occurrence of raster data is elevation which has been converted from a USGS DEM file.

See Also: *.TAA

Access: The filenames for the *.TAC files are located in the LIMITxx.BIN file. The LIMITxx.BIN and TAC files are read during the initialization phase of most graphical IBS programs when the topography is selected.

Include: MAP$DEVELOP$INC:TACIOVAR.INC
MAP$DEVELOP$INC:TVCIOVAR.INC

Notes: Run length encoding and 3 dimensional data are supported by the TAC file structure, but are not supported in IBS.

Structure: Binary, direct access, fixed length, 8192 bytes/record

Byte 0:	CFILEO_XREC	I*2	TAV/TAC file record of next available word
Byte 2:	CFILEO_XWRD	I*2	Next available word in record x_XREC
Byte 4:	CFILEA_ID	I*4	Unique file id for TAC file
Byte 8:	CFILEA_SCALE	I*4	Scale of map for TAC file (250000 for DEM files)
Byte 12:	CFILEA_SECURITY	I*4	Security classification (0 for DEM files)
Byte 16:	CFILEA_LIMIT(6)	R*8	X-Y-Z limits of TAC file (Bounding rectangle) (1,2 - Min/Max X; 3,4 - Y; 5,6 - Z)
Byte 64:	CFILEA_STATS(6)	R*8	TAC file statistics array 1 - # of non-zero cells 2 - Minimum scaler value. 3 - Maximum scaler value. 4 - Average scaler values. 5 - Standard Deviation of scaler values. 6 - Sum of all scaler values.
Byte 112:	CFILEA_COVER(5)	R*8	TAC file X-Y-{Z} steps, cell scale, and cell offset 1 - X step size 2 - Y step size 3 - Z step size 4 - Cell scale (typically 1.0)

			5 - Cell offset (typically 0.0)
Byte 152:	CFILEA_PRJ_DATA(15)	R*8	TAC file's projection data parameters (typically 0)
Byte 272:	CFILEA_MAP_PRJ	I*4	Map projection data is stored in (1-Geographic, 2-UTM, etc.) (Refer to USGS standard map projections)
Byte 276:	CFILEA_DCR(14)	I*4	General descriptive information including #rows, #columns, #planes, data type, data format, etc. 1 - # of columns 2 - # of rows 3 - # of planes 4 - Data type (typically 3 - continuous) 5 - Size of data type (1-I*1, 2-I*2, 4-I*4) 8 - Offset to cell categorical descriptive attributes 9 - # of cell categorical interpretive attributes 10 - (set to 65536 for DEM conversion) 11 - # of cell categorical descriptive attributes 13 - No run length encoding (set to 1) 14 - No run length encoding (set to 0)
Byte 332:	CFILEA_DUNITS	C*32	Units of cell data ('METERS' for DEM files)
Byte 364:	CFILEA_ZUNITS	C*32	Units of Z data
Byte 396:	CFILEA_MAP	C*32	Name of MAP data was digitized from
Byte 428:	CFILEA_TAD	C*32	Name of the associated TAD file (if categorical cell data) (typically 'NONE')
Byte 460:	CFILEA_TAA	C*32	Name of the associated TAA file (Optional)
Byte 492:	CFILEA_DESC	C*32	ASCII description of file (user discretion)
Byte 524:	CFILEA_VALID(2)	C*14	Dates over which data is considered valid

Byte 552:

CFILEA_CTG_INTR(CFILEA_DCR(9))

I*4 Cell categorical interpretive attributes.

Byte 552+CFILEA_DCR(8)*2:

CFILEA_CTG_DESC(CFILEA_DCR(11))

I*4 Cell categorical descriptive attributes.

Filename: *.TXT

Directory: MAP$DATA$xx:

Purpose: The TXT files contains text labels for the associated DMS file(s).

See Also: *.DMS

Access: The DMS file contains the file name of the TXT file. Both the DMS file and the TXT file are read during the initialization phase of most graphical IBS programs.

Structure: DMS text format (See Appendix C)

Filename: ***.TXT**

Directory: MAP$DATA$MAIN:

Purpose: The TXT files contained in the MAP$DATA$MAIN: directory are DMS text files associated with the DMS files that provide low-detail region overview maps. These DMS text files are not used by IBS, but by the program MPDISPLAY for providing an overview of the available map regions. The map region filenames and descriptions are contained in the topographic control file MAP$DATA$MAIN:TOPOCTL.FIL.

See Also: TOPOCTL.FIL, *.DMS

Access: The low-detail map region DMS and TXT files are not accessed by the IBS, but rather by the program MPDISPLAY.

Structure: DMS text format (See Appendix C)

Filename: **generic.ATT**

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:, OFF$SENDDIR:

Purpose: Sent from onpost to offpost or from offpost to onpost. This file contains attribute data for a generic.DLG file. One generic.ATT file will accompany each generic.DLG file.

See Also: generic.DLG, OFF_EVR.DAT, OFF_KPL.DAT, OFF_PA.DAT, OFF_PEVR.DAT, and OFF_PIMR.DAT

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND IBS TO ONPOST REPORT
2. IBS -> MESSAGE BOARD -> VIEW IBS TO ONPOST REPORT
3. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT
4. ONPOSTSIM
5. EVENT

Notes: When the EVENT program finds a file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

OFF$SENDDIR:generic.ATT/DLG files are created by IBS from DMS files. These files can be generated with the following reports:

Evacuation Route	contains a graphic representation of an evacuation roadway network. ATT/DLG filenames are referenced in the Evacuation Route report, OFF_EVR.DAT.
Generic Graphics	contains the graphics of any DMS files. ATT/DLG filenames are referenced in the GENGRAPH.DAT file.
Known Points	See OFF_KP.ATT and OFF_KP.DLG.
Known Polygon	contains known polygons, describing such things as PAZs, IRZs, etc, and is a graphical version of the OFF_KPL.DAT file. The name of the known polygon is attached to each via a label. ATT/DLG filenames are referenced in the known polygon report, OFF_KPL.DAT.
Protective Action	contains a graphic representation of areas for which protective action has been ordered. ATT/DLG filenames are referenced in the Protective Action file, OFF_PA.DAT.
Proposed Evacuation Route	contains a graphic representation of an evacuation roadway network. ATT/DLG filenames are referenced in and accompany the Proposed Evacuation Route report, OFF_PEVR.DAT.

Proposed Military Routes contains a graphic representation of proposed incoming military routes. ATT/DLG filenames are referenced in the Proposed Military Routes file, OFF_PIMR.DAT.

Structure: ATT file

Filename: **generic.DAT**

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:, OFF$SENDDIR:

Purpose: Sent from onpost to offpost or from offpost to onpost. This file contains text which is not passed in any other file.

Access:
1. IBS -> MESSAGE BOARD -> CREATE IBS TO ONPOST REPORT
2. IBS -> MESSAGE BOARD -> RESEND IBS TO ONPOST REPORT
3. IBS -> MESSAGE BOARD -> VIEW IBS TO ONPOST REPORT
4. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT
5. ONPOSTSIM
6. EVENT

Include: IEMIS$INC:HEADER.INC

Notes: When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, variable format, <= 80 bytes/record
This file contains up to WS_GEN_MAX (100) records of miscellaneous information.

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K (K=J+2;J=1,N;1<=N<=WS_GEN_MAX):
FORMAT (A80)

WS_GN_RECORD(J)	C*(*)	Report record

Filename: **generic.DLG**

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:, OFF$SENDDIR:

Purpose: Sent from onpost to offpost or from offpost to onpost. This file contains graphic data not passed in any other file. It will always be accompanied by a generic.ATT file.

See Also: generic.ATT, OFF_EVR.DAT, OFF_KPL.DAT, OFF_PA.DAT, OFF_PEVR.DAT, and OFF_PIMR.DAT

Access:
1. IBS -> MESSAGE BOARD -> CREATE IBS TO ONPOST REPORT
2. IBS -> MESSAGE BOARD -> RESEND IBS TO ONPOST REPORT
3. IBS -> MESSAGE BOARD -> VIEW IBS TO ONPOST REPORT
4. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT
5. ONPOSTSIM
6. EVENT

Notes: When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

See generic.ATT for a list and description of the different types of generic ATT/DLG files sent from offpost to onpost.

Structure: DLG file

Filename: **pppp.DAT**

Directory: SITE$POP:

Purpose: Provide population/evac model centroid cross reference table for a particular population data set, and evacuation traffic assignment case.

Access: 1. BUILDNET

Include: EESF$INC:BUILDNET.INC
EESF$INC:EVAC_BOUNDARY.INC

Structure: Binary, sequential, variable length

Record 1: 12 bytes
UNFORMATTED

CENTROIDS	I*4	Number of IDYNEV model centroids.
NM_CED	I*4	Number of population points.
NPOINTS	I*4	Number of evacuation boundary vertices.

Record 2,CENTROIDS+1: 8-1208 bytes
UNFORMATTED

CENTROID_LOC(I)	I*4	Index to link to which centroid belongs.
CENTROID_POP(I)	I*4	Population to be loaded at centroid.
CENTROID_CED_LIST(J,I)	I*4	List of CED numbers associated with a given centroid (max val of j is 300)

Record CENTROIDS+2, CENTROIDS+NM_CED+1: 24 bytes
UNFORMATTED

CED_CENTROID_PTR(I)	I*4	Index to associated centroid.
CED_LATI(I)	R*8	Latitude of population point.
CED_LONG(I)	R*8	Longitude of population point.
CED_POPS(I)	I*4	Population at population point.

Record CENTROIDS+NM_CED+2,CENTROIDS+NM_CED+NPOINTS+1: 16 bytes
UNFORMATTED

X_COORD(I)	R*8	Longitude of evac boundary point.
CED_LATI(I)	R*8	Latitude of evac boundary point.

Filename: xxxxEL.TAA

Directory: SITE$DIR:

Purpose: This file provides the OSPM model with longitude/latitude projected elevation data in meters, at 3 arc second resolution. The naming convention is to replace *xxxx* with the 4-character site code.

Access: 1. OSPM_(NO)GR

Include: MAP$DEVELOP$INC:TVCIO.INC
MAP$DEVELOP$INC:TACIO.INC

Structure: TAA format.

Filename: xxxxEL.TAC

Directory: SITE$DIR:

Purpose: This file provides the OSPM model with longitude/latitude projected elevation data in meters, at 3 arc second resolution. The naming convention is to replace *xxxx* with the 4-character site code.

See Also: xxxxEL.TAA

Access: 1. OSPM_(NO)GR

Include: MAP$DEVELOP$INC:TVCIO.INC
MAP$DEVELOP$INC:TACIO.INC

Structure: TAC format

Filename: xxxxELAT.TAA

Directory: SITE$DIR:

Purpose: This file provides the OSPM model with longitude/latitude projected sound attenuation data for a site. The naming convention is to replace *xxxx* with the 4-character site code.

See Also: xxxxELAT.TAC

Access: 1. OSPM_(NO)GR

Include: MAP$DEVELOP$INC:TVCIO.INC
MAP$DEVELOP$INC:TACIO.INC

Structure: TAA format. This TAA file should contain long/lat projected byte sound attenuation data at 3 arc second resolution. Values of cells are 0 or 1 where 0 indicates acoustically soft terrain and 1 indicates acoustically hard terrain.

Filename: xxxxELAT.TAC

Directory: SITE$DIR:

Purpose: This file provides the OSPM model with longitude/latitude projected sound attenuation data for a site. The naming convention is to replace *xxxx* with the 4-character site code.

See Also: xxxx.ELAT.TAA

Access: 1. OSPM_(NO)GR

Include: MAP$DEVELOP$INC:TVCIO.INC
MAP$DEVELOP$INC:TACIO.INC

Structure: TAC format. This TAC file should contain long/lat projected byte sound attenuation data at 3 arc second resolution. Values of cells are 0 or 1 where 0 indicates acoustically soft terrain and 1 indicates acoustically hard terrain.

Filename: **xxxxEL_UTM.TAA**

Directory: SITE$DIR:

Purpose: This file provides the OSPM model with UTM projected elevation data for a site. The naming convention is to replace *xxxx* with the 4-character site code.

See Also: xxxxEL_UTM.TAC

Access: 1. OSPM_(NO)GR

Include: MAP$DEVELOP$INC:TVCIO.INC
MAP$DEVELOP$INC:TACIO.INC

Structure: TAA format. This TAA file should contain UTM projected integer elevation data in meters, at 3 arc second resolution.

Filename: xxxxEL_UTM.TAC

Directory: SITE$DIR:

Purpose: This file provides the OSPM model with UTM projected elevation data for a site. The naming convention is to replace *xxxx* with the 4-character site code.

See Also: xxxxEL_UTM.TAA

Access: 1. OSPM_(NO)GR

Include: MAP$DEVELOP$INC:TVCIO.INC
MAP$DEVELOP$INC:TACIO.INC

Structure: TAC format

Filename: **xxxPROP.DAT (903 FILES)**

Directory: IEMIS$CHEMS:

Purpose: Define chemical properties of a particular chemical.

Access: 1. CHEMS_(NO)GR

Include: PRPCOM.FOR

Notes: *xxx* in the filename is a representation of the HACS code of the chemical for which properties are defined in the file. If the file for the specified chemical is not found, a dummy prop dat file (ERRPROP.DAT) is opened and properties are read from it. Together, all these files define a chemical property database. Each file, as described in the record descriptions that follow is comprised of one or more sets (up to a maximum of 2000) of header lines, optional comment line(s) and either textual property description line(s) or numeric property line(s).

Structure: ASCII sequential.

Record 1: 16 bytes, File header

```
FORMAT (4I4,T30,A)
```

IP	I*4	Property number or if -1, a flag indicating header information.
IPARS	I*4	Number of parameters for property IP.
ICOMS	I*4	Number of comment lines.
ISTR	I*4	Type of parameter: 0=numeric, all others=characters.
CDUM	C*(*)	Filename or property name

Record 2: 34 bytes, Version header

```
FORMAT (21X,'VERSION ',I2,'.',I2)
```

IV1	I*4	Version number " 2".
IV2	I*4	Version point number "11".

Record 3,ICOMS: 80 bytes, Comment line

```
FORMAT (A1)
```

DUMMY	C*1	Comment.

Record ICOMS+1,ICOMS+IPARS: 80 bytes, Property strings

```
FORMAT (A80)
```

PROPST(ID)	C*80	Property string.

Record ICOMS+1,ICOMS+INT((IPARS-1)/5)+1:

80 bytes, Property values

```
FORMAT (5G16.5)
```

A	R*4	Property value.
B	R*4	Property value.
C	R*4	Property value.
D	R*4	Property value.
E	R*4	Property value.

Filename: AGENCY_LOOKUP.DAT

Directory: OFF$DIR:

Purpose: This file contains a list of agency acronyms or abbreviations together with their full name.

Access:
1. IBS -> RESOURCE MANAGEMENT
2. IBS -> PERSONNEL MANAGEMENT
3. IBS -> STATUS BOARD -> IP
4. RMMGR

Include: IEMIS$INC:AG.INC

Notes: Routine M_AG_READ reads the file. The parameter AG_MAX limits the number of records in the file.

Structure: ASCII, direct access, 80 bytes/record, shared and locked. This file contains *n* records, each with an agency acronym or abbreviation followed by the full agency name.

Record n: 80 bytes
`FORMAT (A5,A64,11X)`

AG_CODE	C*5	Agency acronym or abbreviation
AG_NAME	C*64	Full agency name

Filename: **ALLSITE.DAT**

Directory: IEMIS$SYSF:

Purpose: List of all of the accessible sites on the system. It is created by ALLSITE or created and used by FIXMSTRDIR. It is used primarily for documentation purposes.

Access:
1. ALLSITE
2. FIXMSTRDIR

Structure: ASCII, sequential, fixed length, 40 bytes/record.

Record N: `FORMAT(A6, 1X, A15, 1X, A12, 1X, A4)`

NODE	N/A	Node the site is on
DISK	N/A	Disk the site is on
USER	N/A	User the site is under
SITE	N/A	Name of the site

Filename: **ATTRIBDIC.BIN**

Directory: MAP$ATTRIB:

Purpose: This file is the attribute dictionary that lists the attribute codes and descriptions. The attribute information is used to assign physical or other descriptive features to topographic data within the IBS system.

See Also: ATTRIBDIC.BIX

Access: This file is read during the initialization phase of most GIS programs in IBS. The ATTXASC utility converts from the binary to the ASCII format, which can be printed or edited.

Include: MAP$DEVELOP$INC:ATTDIC.INC

Structure: Binary, sequential, 77 bytes/record. Each record contains a unique attribute code with additional descriptive information.

Record n: 77 bytes

NEW_MAJOR	I*4	Attribute major code
NEW_MINOR	I*4	Attribute major code
PARM_ATTDIC	I*2	Attribute parameter code
DESC	C*60	Attribute description
PEN_ATTDIC	I*1	Default pen type
LINTYP_ATTDIC	I*1	Default line type
CLUTTER_ATTDIC	I*1	Decluttering code
CLUTTER_SF_ATTDIC	R*4	Decluttering scale factor

Filename: **ATTRIBDIC.BIX**

Directory: MAP$ATTRIB:

Purpose: This file is an ASCII version of the binary file, ATTRIBDIC.BIN, that lists the attribute codes and descriptions.

See Also: ATTRIBDIC.BIN

Access: This file is not used directly by the IBS. Rather, the ASCII version of the attribute dictionary is used for viewing, editing, and reporting attribute information. The utilities ATTXASC and ASCXATT convert to/from the ASCII file from/to the binary file.

Include: MAP$DEVELOP$INC:ATTDIC.INC

Structure: ASCII, sequential, 99 bytes/record. Each record contains a unique attribute code with additional descriptive information.

Record n: 99 bytes

```
FORMAT (I2,I5,I7,3I4,F12.6,1X,A)
```

NEW_MAJOR	I*4	Attribute major code
NEW_MINOR	I*4	Attribute minor code
PARM_ATTDIC	I*2	Attribute parameter code
PEN_ATTDIC	I*1	Default pen type
LINTYP_ATTDIC	I*1	Default line type
CLUTTER_ATTDIC	I*1	Decluttering code
CLUTTER_SF_ATTDIC	R*4	Decluttering scale factor
DESC	C*60	Attribute description

Filename: **BASE_POP.DAT**

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: Transferred from onpost to offpost. This file contains population and planned evacuation information for the depot. It will be transmitted whenever a significant change is made to the base evacuation plan.

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> BASE POPULATION
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. ONPOSTSIM
4. EVENT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes: When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file has 5 records of miscellaneous information, followed by up to WS_POP_MAX (20) records with the names of gates to be used for evacuation and the number of vehicles expected at that gate.

Record 1-2: File header and description

```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record 3,7:

```
FORMAT ('Essential daytime population',T35,':',1X,I4/
'Essential nighttime population',T35,':',1X,I4/
'Non-essential daytime population',T35,':',1X,I4/
'Non-essential nighttime population',T35,':',1X,I4/
'People per vehicle',T35,':',1X,F4.1)
```

WS_POP_DE	I*4	Essential daytime population
WS_POP_NE	I*4	Essential nighttime population

WS_POP_DN	I*4	Non-essential daytime population
WS_POP_NN	I*4	Non-essential nighttime population
WS_POP_PPV	R*4	People per vehicle

Record K1 (K1=J+7;J=1,N;1<=N<=WS_POP_MAX):

```
FORMAT (A40,1X,I4,35X)
```

WS_POP_GATE(J)	C*40	Name (known point) of gate to be used for evacuation
WS_POP_NVEH(J)	I*4	# of vehicles expected at gate

Filename: **C000*.***

Directory: IEMIS$SYSF:

Purpose: These case 0 files are templates that a user can use to create a new case for one of the models.

See Also: The files listed in the notes section.

Access: NEWSITE copies them to the appropriate model directories.

Notes:

Model	File
CHEMS	SITEICHEMS:CnnnCHEMS.GEN
CHEMS	SITEICHEMS:CRASHFLG.DAT
D2	SITEID2:CnnnD2INP.DAT
DYNEV	SITEIDYNEV:CnnnINP.DAT
DYNEV	SITEIDYNEV:CnnnEVACPARAM.DAT
MESORAD	SITEIMESORAD:CnnnCHK.DAT
MESORAD	SITEIMESORAD:CnnnOBS.DAT
MESORAD	SITEIMESORAD:CnnnPOL.DAT
MESORAD	SITEIMESORAD:CnnnRSP.DAT
MESORAD	SITEIMESORAD:CnnnSnm.DAT
MESORAD	SITEIMESORAD:CnnnSTA.DAT
MESORAD	SITEIMESORAD:CnnnTER.DAT
MESORAD	SITEIMESORAD:CnnnTOP.DAT
MESORAD	SITEIMESORAD:CnnnV.DAT
OSPM	SITEIOSPM:CnnnOSPM.DAT

Structure: See the data files listed in the notes section.

Filename: **CnnnACTION_NOTES.DAT**

Directory: SITE$IP:

Purpose: This file contains all the actions notes for all the tasks listed for IP number nnn. The file can contain up to 99 lines of action notes for each of the 99 tasks and 99 tasks for each of the possible emergency functions.

Access:
1. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES -> CHANGE/USE TASKS -> ACTION NOTES. Which depends on whether you are in planning mode (CHANGE) or in operational mode (USE).
2. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES -> REPORT

Include: IEMIS$INC:IP_XN.INC

Structure: Direct access, 102 bytes/record, with record locking at the task level during operational mode. In planning mode, no record locking occurs. The first record lists the number of records contained in the file.

Record 0: 36 bytes, header
`FORMAT (I6,1X,A23,1X,A5,66X)`

NUM_REC	I*4	Number of records in the file
CDTS	C*23	Current system date/time when file header last modified
IP_VERSION	C*5	Current IP version

Record 1-NUM_REC: 102 bytes, action note record
`FORMAT (A20,A6,A76)`

ID	C*20	Emergency function
TASK_IDENT	C*6	ID linking note to given task
XN_NOTES	C*76	Actual line of notes

Filename: **CnnnACTIONS.DAT**

Directory: SITE$IP:

Purpose: This file contains all the actions for all the tasks listed for IP number nnn. There can be up to 99 actions for each of the 99 tasks and 99 tasks for each of the possible emergency functions.

Access:
1. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES -> CHANGE/USE TASKS -> ACTIONS. Which depends on whether you are in planning mode (CHANGE) or in operational mode (USE).
2. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES -> REPORT

Include: IEMIS$INC:IP_AC.INC

Structure: Direct access, 230 bytes/record, fixed length, with record locking at the task level during operational mode. In planning mode, records are not locked. The first record lists the number of records contained in the file.

Record 0: 36 bytes, header
`FORMAT (I6,1X,A23,1X,A5,194X)`

NUM_REC	I*4	Number of records in the file
CDTS	C*23	Current system date/time when file header last modified
IP_VERSION	C*5	Current IP version

Record 1-NUM_REC: 230 bytes, Action
`FORMAT (A20,A6,A4,A40,A160)`

ID	C*20	Emergency function corresponding to action
TASK_IDENT	C*6	Unique task number ID for given task in given EF
AC_NO	C*4	Action record number for given task
AC_LOC	C*40	Location (known point) associated with action
AC_DESC	C*160	Short description of action

Filename: **CnnnAIRDPL.ADV**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of time integrated ground-level air concentrations with deposition, washout, decay, and ingrowth for each advection (time) step.

Access: MES_(NO)GR

Include: MESO$INC:MATRIX.INC
MESO$INC:CIGMAT.INC

Notes: This file does not appear to be displayable because no menu lists it.

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((EXPARR(K,32-J,N),K=1,31),J=1,31)		
	R*4	Concentration levels in Cartesian grid (Curie-Seconds per Cubic Meter) (N=3).
((PGEARR(I,J,N),I=1,36),J=1,3)		
	R*4	Concentration levels in polar grid (Curie-Seconds per Cubic Meter) (N=3)

Filename: **CnnnAIRDPL.CUM**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of cumulative time integrated ground-level air concentrations with deposition, washout, decay, and ingrowth for each advection (time) step - cumulative from Start to Current time step.

Access: MES_(NO)GR

Include: MESO$INC:MATRIX.INC
MESO$INC:CIGMAT.INC

Notes: This file does not appear to be displayable because no menu lists it.

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((EXPARR(K,32-J,N),K=1,31),J=1,31)
R*4 Concentration levels in Cartesian grid (Curie-Seconds per Cubic Meter) (N=4)

((PGEARR(I,J,N),I=1,36),J=1,3)
R*4 Concentration levels in polar grid (Curie-Seconds per Cubic Meter) (N=4).

Filename: **CnnnAIRNDP.ADV**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of time integrated ground-level air concentrations assuming no depletion and no radioactive decay for each advection (time) step.

Access:
1. MES_(NO)GR->MESORAD GRAPHIC OPTIONS->Curr Relative Plume Concentration
2. EESF_GR->Display Map-Related Graphics->Curr Relative Plume Concentration

Include: MESO$INC:MATRIX.INC
MESO$INC:CIGMAT.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((EXPARR(K,32-J,N),K=1,31),J=1,31)	R*4	Concentration levels in Cartesian grid (Curie-Seconds per Cubic Meter) (N=3).
((PGEARR(I,J,N),I=1,36),J=1,3)	R*4	Concentration levels in polar grid (Curie-Seconds per Cubic Meter) (N=3).

Filename: **CnnnAIRNDP.CUM**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of time integrated ground-level air concentrations assuming no depletion and no radioactive decay for each advection (time) step - cumulative from start to current time step.

Access:
1. MES_(NO)GR->MESORAD GRAPHIC OPTIONS->Cumul Relative Plume Concentration
2. EESF_GR->Display Map-Related Graphics->Cumul Relative Plume Concentration

Include: MESO$INC:MATRIX.INC
MESO$INC:CIGMAT.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((EXPARR(K,32-J,N),K=1,31),J=1,31)

	R*4	Concentration levels in Cartesian grid (Curie-Seconds per Cubic Meter) (N=3).

((PGEARR(I,J,N),I=1,36),J=1,3)

	R*4	Concentration levels in polar grid (Curie-Seconds per Cubic Meter) (N=3).

Filename: **CnnnBIN.U36**

Directory: SITEODYNEV:

Purpose: This file provides the cumulative sector statistics in both polar and Cartesian coordinates for the MESORAD models.

Access: 1. MES_(NO)GR

Include: EESF$INC:LEV.INC

Notes: Most common block items within IDYNEV are maintained in their own separate common blocks rather than in an INCLUDE file. IDYNEV also uses IMPLICIT statements rather than IMPLICIT NONE in defining variables. The 640 records written to this file are divided as follows:

Records 1-240 - polar coordinate grid
Records 241-640 - Cartesian coordinate grid.

Structure: Binary, sequential, 29 bytes/record

Record 1-640:

CLOCK	I*4	Clock time
ISP	I*4	Sector number
RPVMIL(ISP)	R*4	Cumulative vehicle miles
RPVMIN(ISP)	R*4	Cumulative vehicle minutes
RSPD	R*4	Cumulative Sector Speed
RPCNT(ISP)	R*4	Current sector roadway content
RPSRC(ISP)	R*4	Current sector content
KKILOS(JLOS)	C*1	Service code

Filename: **CnnnCHEMS.GEN**

Directory: SITEICHEMS:, IEMIS$SYSF:

Purpose: This file contains the definition of case *nnn* for the CHEMS models. The version in IEMIS$SYSF:, C000CHEMS.GEN, is a template.

Access: 1. CHEMS_(NO)GR

Include: CHEMS$INC:GEN.INC

Notes: GEN.INC includes about 30 other include files, most or all of which correspond to data entry forms. It also includes a number of variables not obtained by forms. The file itself consists of 235 records. The first record is a case description, followed by 234 variables with one variable per record and a label following the variable.

Structure: ASCII, Sequential.

Record 1: 80 bytes
FORMAT (*)

DESCRIPTION	C*80	Case description.

Note this record actually spans two records, if updated by CHEMS software due to the (*) format it is written in.

Record 2: Variable length
FORMAT (*)

The following records are related to accident conditions.

literal	C*1	' '.
CD1	I*4	Current date (day).
literal	C*13	' CD1-ACC_COND'.

Record 3: Variable length
FORMAT (*)

literal	C*1	' '.
CD2	I*4	Current date (month).
literal	C*4	' CD2'.

Record 4: Variable length
FORMAT (*)

literal	C*1	' '.

	CD3	I*4	Current date (year).
	literal	C*4	' CD3'.
Record 5:	13 bytes `FORMAT (' ',A8,' TM1')`		
	TM1	C*8	Current time (HH:MM:SS).
Record 6:	Variable length `FORMAT (*)`		
	literal	C*1	' '.
	DATE1	I*4	Accident date (day).
	literal	C*6	' DATE1'.
Record 7:	Variable length `FORMAT (*)`		
	literal	C*1	' '.
	DATE2	I*4	Accident date (month).
	literal	C*6	' DATE2'.
Record 8:	Variable length `FORMAT (*)`		
	literal	C*1	' '.
	DATE3	I*4	Accident date (year).
	literal	C*6	' DATE3'.
Record 9:	Variable length `FORMAT (*)`		
	literal	C*1	' '.
	TIME1	I*4	Accident time (hour).
	literal	C*6	' TIME1'.
Record 10:	Variable length `FORMAT (*)`		
	literal	C*1	' '.
	TIME2	I*4	Accident time (minute).
	literal	C*6	' TIME2'.
Record 11:	Variable length `FORMAT (*)`		
	literal	C*1	' '.

	SP_LAT	R*8	Spill latitude.
	literal	C*4	' LAT'.
Record 12:	Variable length FORMAT (*)		
	literal	C*1	' '.
	SP_LON	R*8	Spill longitude.
	literal	C*4	' LON'.
Record 13:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KLOC	I*4	Spill location code (1, 2, or 3) If on land, on water, or under water.
	literal	C*5	' KLOC'.
Record 14:	Variable length FORMAT (*)		
	literal	C*1	' '.
	TAIR	R*4	Air temperature (for units see next field).
	literal	C*5	' TAIR'
Record 15:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KA2	I*4	Unit code (1, 2, or 3 for Fahrenheit, Celsius or Kelvin).
	literal	C*4	' KA2'.
Record 16:	Variable length FORMAT (*)		
	literal	C*1	' '.
	RHU	R*4	Relative humidity of air.
	literal	C*4	' RHU'.
Record 17:	Variable length FORMAT (*)		
	literal	C*1	' '.
	UW	R*4	Wind speed (see following field for units).
	literal	C*3	' UW'.
Record 18:	Variable length		

FORMAT (*)

literal	C*1	' '.
KA1	I*4	Unit code (1, 2, 3, 4, or 5 for mph, knots, ft/s, m/s, km/hr).
literal	C*4	' KA1'.

Record 19: Variable length
FORMAT (*)

literal	C*1	' '.
WH	R*4	Height (in meters) of wind speed measurement.
literal	C*3	' WH'.

Record 20: Variable length
FORMAT (*)

literal	C*1	' '.
WD	R*4	Direction wind is blowing from (degrees).
literal	C*3	' WD'.

Record 21: Variable length
FORMAT (*)

literal	C*1	' '.
CLCO	R*4	Cloud cover? (This definition may not be correct. It is read and written, but not modified or included in any forms.)
literal	C*5	' CLCO'

Record 22: Variable length
FORMAT (*)

literal	C*1	' '.
TG	R*4	Ground temperature (range -50 to 100). (See next field for units.)
literal	C*3	' TG'.

Record 23: Variable length
FORMAT (*)

literal	C*1	' '.
KKGP	I*4	Unit code (1, 2, or 3 for Fahrenheit, Celsius or Kelvin).
literal	C*5	' KKGP'.

Record 24: Variable length

FORMAT (*)

literal	C*1	' '.
RL	I*4	Surface roughness code for dispersion region (range 1 to 6). 1= Smooth mud flats, ice (0.001 cm) 2= Calm sea/snow covered flat ground (0.01 cm) 3= Flat desert/natural snow surface (0.1 cm) 4= Cut grass/few trees (wintertime) (1.0 cm) 5= Airport runways/farmlands (10.0 cm) 6= Many Trees/Hedges/Few Buildings (30.0 cm)
literal	C*3	' RL'

Record 25: Variable length
FORMAT (*)

literal	C*1	' '.
RLT	I*4	Surface roughness code for met tower region (range 1 to 6 - see def for RL).
literal	C*4	' RLT'.

Record 26: Variable length
FORMAT (*)

literal	C*1	' '.
LGROUND	I*4	Ground cover conditions code (range 1 to 3). 1= Dry 2= Wet 3= Snow
literal	C*8	' LGROUND'.

Record 27: Variable length
FORMAT (*)

literal	C*1	' '.
IMODE	I*4	Mode of operation (range 1 to 2). 1= Emergency mode 2= Contingency mode
literal	C*6	' IMODE'.

Record 28: Variable length
FORMAT (*)

literal	C*1	' '.

	VBA	R*4	Total capacity of leaking or ruptured tank(s). See next field for units.
	literal	C*4	' VBA'.
Record 29:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KBA1	I*4	Units of volume (range 1 to 6). 1= gallons (U.S.) 2= barrels 3= ft^3 4= m^3 5= liters 6= gallons (U.K.)
	literal	C*5	' KBA1'.
Record 30:	Variable length FORMAT (*)		
	literal	C*1	' '.
	LBA	R*4	Length of tank (see next field for units).
	literal	C*4	' LBA'.
Record 31:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KBA2	I*4	Units of length (range 1 to 2). 1= feet 2= meters
	literal	C*5	' KBA2'.
Record 32:	Variable length FORMAT (*)		
	literal	C*1	' '.
	WBA	R*4	Width of tank (range 0.> - <100.) (See next field for units).
	literal	C*4	' WBA'
Record 33:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KBA3	I*4	Units of length (range 1 to 2).

			1= feet
			2= meters
	literal	C*5	' KBA3'.
Record 34:	Variable length FORMAT (*)		
	literal	C*1	' '.
	HDBA	R*4	Distance from hole to top of tank (range 0.> <100.) (See next field for units.)
	literal	C*5	' HDBA'.
Record 35:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KBA4	I*4	Units of length (range 1 to 2). 1= feet 2= meters
	literal	C*5	' KBA4'.
Record 36:	Variable length FORMAT (*)		
	literal	C*1	' '.
	WDBA	R*4	Distance from water line to top of tank (range 0.> <100.) (See next field for units.)
	literal	C*5	' ADBA'.
Record 37:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KBA5	I*4	Units of length (range 1 to 2). 1= feet 2= meters
	literal	C*5	' KBA5'.
Record 38:	Variable length FORMAT (*)		
	literal	C*1	' '.
	BASIZE	R*4	Diameter of hole (range 0.> <100.) (See next field for units).
	literal	C*7	' BASIZE'.
Record 39:	Variable length FORMAT (*)		

	literal	C*1	' '.
	KBADESCR	I*4	Units of length (range 1 to 2). 1= inches 2= centimeters
	literal	C*9	' KBADESCR'.
Record 40:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KBAEME	I*4	Code corresponding to type of leak (range 1 - 4). 1= Crack on wall (0.5"d) 2= Puncture hole (2"d) 3= Fill pipe rupture (4"d) 4= Entire tank contents
	literal	C*7	' KBAEME'.
Record 41:	Variable length FORMAT (*)		
	literal	C*1	' '.
	ZLBA	R*4	Distance from water line to top of tank (range 0.> <1000). (See next field for units.)
	literal	C*5	' ZLBA'.
Record 42:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KZLBA	I*4	Units of length range (1 - 2). 1= Feet 2= Meters
	literal	C*6	' KZLBA'.
Record 43:	Variable length FORMAT (*)		
	The following records are related to bullet tank cars.		
	literal	C*1	' '.
	VTB	R*4	Total capacity of the tank. (range >=0. <1000000.) (See next record for units.)
	literal	C*11	' VTB-BULLET'.
Record 44:	Variable length FORMAT (*)		

	literal	C*1	' '.
	KB1	I*4	Units of volume range (1 - 6). 1 = gallons (U.S.) 2 = barrels 3 = ft^3 4 = m^3 5 = liters 6 = gallons (U.K.)
	literal	C*4	' KB1'.
Record 45:	Variable length FORMAT (*)		
	literal	C*1	' '.
	TDB	R*4	Diameter of tank (range 0.> <100.) (See next field for units.)
	literal	C*4	' TDB'.
Record 46:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KB2	I*4	Units of length (range 1 to 2). 1= feet 2= meters
	literal	C*4	' KB2'.
Record 47:	Variable length FORMAT (*)		
	literal	C*1	' '.
	BUSIZE	R*4	Diameter of hole (range 0.> <100.) (See next field for units.)
	literal	C*7	' BUSIZE'.
Record 48:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KBDESCR	I*4	Units of length (range 1 to 2). 1= inches 2= centimeters
	literal	C*8	' KBDESCR'.
Record 49:	Variable length FORMAT (*)		

	literal	C*1	' '.
	BULOC	R*4	Distance from tank bottom to the hole (range 0.> <100.) (See next field for units.)
	literal	C*6	' BULOC'.

Record 50: Variable length
FORMAT (*)

literal	C*1	' '.
KBLOC	I*4	Units of length (range 1 to 4). 1= feet 2= meters 3= inches 4= centimeters
literal	C*6	' KBLOC'.

Record 51: 9 bytes
FORMAT (' ',A1,' ANDIKB')

ANDIKB	C*1	"Y" If dike is present; "N" if not.

Record 52: Variable length
FORMAT (*)

literal	C*1	' '.
BDIKE	R*4	Diameter of the dike (range 0.> <1000.) (See next field for units.)
literal	C*6	' BDIKE'.

Record 53: Variable length
FORMAT (*)

literal	C*1	' '.
JDIKE	I*4	Units of length (range 1 to 2). 1= feet 2= meters
literal	C*6	' JDIKE'.

Record 54: Variable length
FORMAT (*)

literal	C*1	' '.
KBDEME	I*4	Code corresponding to type of leak (range 1 - 4). 1= Crack on wall (0.5"d) 2= Puncture hole (2"d) 3= Fill pipe rupture (4"d)

		4= Entire tank contents
literal	C*7	' KBDEME'.

Record 55: Variable length
FORMAT (*)

literal	C*1	' '.
KBLOCEME	I*4	Code corresponding to approximate hole location (range 1 - 4). 1 = 75% Height 2 = 50% Height 3 = 25% Height 4 = 0% Height - bottom
literal	C*9	' KBLOCEME'.

Record 56: Variable length
FORMAT (*)

The following records are related to chemical information.

literal	C*1	' '.
CODE	I*4	Code corresponding to method of identifying the chemical released (range 1 - 5). 1 = CHRIS Chemical code 2 = Chemical Name (full and exact) 3 = CAS # (Chemical Abstract Number) 4 = UN / DOT # (UN or D.O.T. 4 digit number) 5 = STCC # (Standard Transportation Commodity Code)
literal	C*15	' CODE-CHEM_INFO'.

Record 57: 46 bytes
FORMAT (' ',A35,' CHEM_NAME')

CHEM_NAME	C*35	Chemical identifier consistent with CODE.

Record 58: 31 bytes
FORMAT (' ',A3,' CHRIS CODE-CHEMICAL COMMON')

CHRIS	C*3	Chemical CHRIS code.

Record 59: Variable length
FORMAT (*)

The following records are related to tank farms.

	literal	C*1	' '.
	VTF	R*4	Tank capacity (range 0. - 1000000.) (See next record for units.)
	literal	C*9	' VTF-FARM'.
Record 60:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KF1	I*4	Units of volume range (1 - 6). 1 = gallons (U.S.) 2 = barrels 3 = ft^3 4 = m^3 5 = liters 6 = gallons (U.K.)
	literal	C*4	' KF1'.
Record 61:	Variable length FORMAT (*)		
	literal	C*1	' '.
	TDF	R*4	Diameter of the tank (range 0.> <100.) (See next field for units.)
	literal	C*4	' TDF'.
Record 62:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KF2	I*4	Units of length (range 1 to 2). 1= feet 2= meters
	literal	C*4	' KF2'.
Record 63:	Variable length FORMAT (*)		
	literal	C*1	' '.
	FRSIZE	R*4	Diameter of the hole (range 0.> <100.) (See next field for units).
	literal	C*7	' FRSIZE'.
Record 64:	Variable length FORMAT (*)		
	literal	C*1	' '.

	KFDESCR	I*4	Units of length (range 1 to 2). 1= inches 2= centimeters
	literal	C*8	' KFDESCR'.
Record 65:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KFDEME	I*4	Code corresponding to type of leak (range 1 - 4). 1= Crack on wall (0.5"d) 2= Puncture hole (2"d) 3= Fill pipe rupture (4"d) 4= Entire tank contents
	literal	C*7	' KFDEME'.
Record 66:	Variable length FORMAT (*)		
	literal	C*1	' '.
	FRLOC	R*4	Distance from the tank bottom to the hole (range 0.> <100.) (See next field for units.)
	literal	C*6	' FRLOC'.
Record 67:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KFLOC	I*4	Units of length (range 1 to 4). 1= feet 2= meters 3= inches 4= centimeters
	literal	C*6	' KFLOC'.
Record 68:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KFLOCEME	I*4	Code corresponding to approximate hole location (range 1 - 4). 1 = 75% Height 2 = 50% Height 3 = 25% Height 4 = 0% Height- Bottom
	literal	C*9	' KFLOCEME'.

Record 69: 9 bytes
`FORMAT (' ',A1,' ANDIKF')`

ANDIKF	C*1	"Y" If dike is present; "N" if not.

Record 70: Variable length
`FORMAT (*)`

literal	C*1	' '.
DIDIKE	R*4	Diameter of the dike (range 0.> <500.) (See next field for units.)
literal	C*7	' DIDIKE'.

Record 71: Variable length
`FORMAT (*)`

literal	C*1	' '.
KDIKE	I*4	Units of length (range 1 to 2). 1= feet 2= meters
literal	C*4	' KDIKE'.

Record 72: Variable length
`FORMAT (*)`

The following records contain general information.

literal	C*1	' '.
CDATE1	I*4	Contact date (day).
literal	C*16	' CDATE1-GEN_INFO'.

Record 73: Variable length
`FORMAT (*)`

literal	C*1	' '.
CDATE2	I*4	Contact date (month).
literal	C*7	' CDATE2'.

Record 74: Variable length
`FORMAT (*)`

literal	C*1	' '.
CDATE3	I*4	Contact date (year).
literal	C*7	' CDATE3'.

Record 75: 16 bytes
`FORMAT (' ',A8,' CTIME1')`

	CTIME1	C*8	Contact time.
Record 76:	45 bytes `FORMAT (' ',A40,' OSC')`		
	OSC	C*40	Name of On-Scene Coordinator.
Record 77:	51 bytes `FORMAT (' ',A40,' CONT_PERS')`		
	CONT_PERS	C*40	Name of Contact Person.
Record 78:	9 bytes `FORMAT (' ',A3,' PHF1')`		
	PHF1	C*3	3-digit FTS exchange of telephone number of On-Scene Coordinator.
Record 79:	10 bytes `FORMAT (' ',A4,' PHF2')`		
	PHF2	C*4	4-digit FTS telephone number extension of On-Scene Coordinator.
Record 80:	9 bytes `FORMAT (' ',A3,' PHCC')`		
	PHCC	C*3	3-digit commercial line area code of On-Scene Coordinator.
Record 81:	9 bytes `FORMAT (' ',A3,' PHC1')`		
	PHC1	C*3	3-digit commercial line exchange number of On-Scene Coordinator.
Record 82:	10 bytes `FORMAT (' ',A4,' PHC2')`		
	PHC2	C*4	4-digit commercial telephone number extension of On-Scene Coordinator.
Record 83:	61 bytes `FORMAT (' ',A55,' SYN1)`		
	SYN1	C*55	First line of synopsis of event.

Record 84: 61 bytes
```
FORMAT (' ',A55,' SYN2')
```

SYN2	C*55	Second line of synopsis of event.

Record 85: 61 bytes
```
FORMAT (' ',A55,' SYN3')
```

SYN3	C*55	Third line of synopsis of event.

Record 86: 20 bytes
```
FORMAT (' ',A1,' ICON-chem_release')
```

(The following records contain chemical release information.)

ICON	C*1	Code for vessel to use in modeling the chemical release. (range "1" - "7") 1 = Barge 2 = Ship 3 = Rail tanker 4 = Road tanker 5 = Fixed storage tank 6 = Bullet tank 7 = Pipeline

Record 87: Variable length
```
FORMAT (*)
```

literal	C*1	' '.
ACC_LOCL	I*4	Code corresponding to spill environment site's conditions (range 1 - 4). 1= City 2= Suburbs 3= Farmland 4= Desert
literal	C*14	' ACC_LOCL-LAND'.

Record 88: 49 bytes
```
FORMAT (' ',A35,' CH_NAME-PATH')
```

CH_NAME	C*35	Name of chemical to be modeled.

Record 89: 34 bytes
```
FORMAT (' ',A28,' MSG1')
```

MSG1	C*28	Line 1 of model menu ("1 = FIRE RADIATION").

Record 90: 34 bytes
```
FORMAT (' ',A28,' MSG2')
```

MSG2	C*28	Line 2 of model menu ("2 = VAPOR DISPERSION").

Record 91: 34 bytes
```
FORMAT (' ',A28,' MSG3')
```

MSG3	C*28	Line 3 of model menu (" ").

Record 92: 34 bytes
```
FORMAT (' ',A28,' MSG4')
```

MSG4	C*28	Line 4 of model menu ("4 = EXPLOSION").

Record 93: 34 bytes
```
FORMAT (' ',A28,' MSG5')
```

MSG5	C*28	Line 5 of model menu (" ").

Record 94: Variable length
```
FORMAT (*)
```

literal	C*1	' '.
MODEL	I*4	Code corresponding to model selected (range 1 - 5) (See MSG previous records for meaning.)
literal	C*14	' MODEL'.

Record 95: Variable length
```
FORMAT (*)
```

The following records contain pipe spill information.

literal	C*1	' '.
PDIA	R*4	Pipe diameter (range 0.> <100.) (See next field for units.)
literal	C*10	' PDIA-PIPE'.

Record 96: Variable length
```
FORMAT (*)
```

literal	C*1	' '.
KP1	I*4	Units of length (range 1 to 4). 1= feet 2= meters 3= inches 4= centimeters

	literal	C*4	' KP1'.
Record 97:	Variable length FORMAT (*)		
	literal	C*1	' '.
	PP	R*4	Pipe pressure - absolute (range 0.> <1000000.) (See next field for units.)
	literal	C*3	' PP'.
Record 98:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KP2	I*4	Units of pressure (range 1 to 6). 1 = psig 2 = inches of water 3 = mm Hg 4 = atm 5 = bars 6 = Pascal
	literal	C*4	' KP2'.
Record 99:	Variable length FORMAT (*)		
	literal	C*1	' '.
	PT	R*4	Pipe temperature (range -150.> <600.) (See next field for units.)
	literal	C*3	' PT'.
Record 100:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KP3	I*4	Units of temperature (range 1 to 3). 1 = Fahrenheit 2 = Celsius 3 = Kelvin
	literal	C*4	' KP3'.
Record 101:	Variable length FORMAT (*)		
	literal	C*1	' '.
	FL	R*4	Flow rate of chemical through the pipe (>0. <100.) (See next field for units.)

	literal	C*3	' FL'.
Record 102:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KFL	I*4	Units of pressure (range 1 to 2). 1 = ft/s 2 = m/s
	literal	C*4	' KFL'.
Record 103:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KPST	I*4	Code for the state of the chemical in the pipe (1 or 2). 1 = Gas 2 = Liquid
	literal	C*5	' KPST'.
Record 104:	Variable length FORMAT (*)		
	literal	C*1	' '.
	PHSIZE	R*4	Diameter of the hole (range 0.> <100.) (See next field for units.)
	literal	C*7	' PHSIZE'.
Record 105:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KPHDESCR	I*4	Units of length (range 1 to 2). 1= inches 2= centimeters
	literal	C*9	' KPHDESCR'.
Record 106:	Variable length FORMAT (*)		

The following records contain rail car spill information.

literal	C*1	' '.
VTR	R*4	Total capacity of tank (range 0.> <100000.) (See next field for units.)
literal	C*9	' VTR-RAIL'.

Record 107: Variable length
FORMAT (*)

literal	C*1	' '.
KR1	I*4	Units of volume (range 1 to 6). 1 = gallons (U.S.) 2 = barrels 3 = ft^3 4 = m^3 5 = liters 6 = gallons (U.K.)
literal	C*4	' KR1'.

Record 108: Variable length
FORMAT (*)

literal	C*1	' '.
TDR	R*4	Tank diameter (range 0.> <100.) (See next field for units.)
literal	C*4	' TDR'.

Record 109: Variable length
FORMAT (*)

literal	C*1	' '.
KR2	I*4	Units of length (range 1 to 2). 1 = feet 2 = meters
literal	C*4	' KR2'.

Record 110: Variable length
FORMAT (*)

literal	C*1	' '.
RSIZE	R*4	Hole diameter (range 0.> <100.) (See next field for units.)
literal	C*6	' RSIZE'.

Record 111: Variable length
FORMAT (*)

literal	C*1	' '.
KRDESCR	I*4	Units of length (range 1 to 2). 1 = inches 2 = centimeters
literal	C*8	' KRDESCR'.

Record 112: Variable length
FORMAT (*)

literal	C*1	' '.
RLOCA	R*4	Distance from the tank bottom to the hole (range 0.> <100.) (See next field for units.)
literal	C*6	' RLOCA'.

Record 113: Variable length
FORMAT (*)

literal	C*1	' '.
KRLOC	I*4	Units of length (range 1 to 4). 1= feet 2= meters 3= inches 4= centimeters
literal	C*6	' KRLOC'.

Record 114: Variable length
FORMAT (*)

literal	C*1	' '.
KRDEME	I*4	Code corresponding to type of leak (range 1 - 4). 1= Crack on wall (0.5"d) 2= Puncture hole (2"d) 3= Fill pipe rupture (4"d) 4= Entire tank contents
literal	C*7	' KRDEME'.

Record 115: Variable length
FORMAT (*)

literal	C*1	' '.
KRLOCEME	I*4	Code corresponding to approximate hole location (range 1 - 4). 1 = 75% Height 2 = 50% Height 3 = 25% Height 4 = 0% Height- Bottom
literal	C*9	' KRLOCEME'.

Record 116: Variable length
FORMAT (*)

The following records contain road tanker spill information.

	literal	C*1	' '.
	VTRO	R*4	Total tank capacity range 0.> <100000.) (See next field for units.)
	literal	C*10	' VTRO-ROAD'.
<u>Record 117</u>:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KRO1	I*4	Units of volume (range 1 to 6). 1 = gallons (U.S.) 2 = barrels 3 = ft^3 4 = m^3 5 = liters 6 = gallons (U.K.)
	literal	C*5	' KRO1'.
<u>Record 118</u>:	Variable length FORMAT (*)		
	literal	C*1	' '.
	TDRO	R*4	Tank diameter (range 0.> <100.) (See next field for units.)
	literal	C*5	' TDRO'.
<u>Record 119</u>:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KRO2	I*4	Units of length (range 1 to 2). 1 = feet 2 = meters
	literal	C*4	' KRO2'.
<u>Record 120</u>:	Variable length FORMAT (*)		
	literal	C*1	' '.
	HSIZE	R*4	Hole diameter (range 0.> <100.) (See next field for units.)
	literal	C*6	' HSIZE'.
<u>Record 121</u>:	Variable length FORMAT (*)		
	literal	C*1	' '.

	KRODESCR	I*4	Units of length (range 1 to 2). 1 = inches 2 = centimeters
	literal	C*9	' KRODESCR'.
Record 122:	Variable length FORMAT (*)		
	literal	C*1	' '.
	HLOCA	R*4	Distance from the tank bottom to the hole (range 0.> <100.) (See next field for units.)
	literal	C*6	' HLOCA'.
Record 123:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KROLOC	I*4	Units of length (range 1 to 4). 1= feet 2= meters 3= inches 4= centimeters
	literal	C*7	' KROLOC'.
Record 124:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KRODEME	I*4	Code corresponding to type of leak (range 1 - 4). 1= Crack on wall (0.5"d) 2= Puncture hole (2"d) 3= Fill pipe rupture (4"d) 4= Entire tank contents
	literal	C*8	' KRODEME'.
Record 125:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KROLOCEME	I*4	Code corresponding to approximate hole location (range 1 - 4). 1 = 75% Height 2 = 50% Height 3 = 25% Height 4 = 0% Height- Bottom
	literal	C*10	' KROLOCEME'.

Record 126: Variable length
FORMAT (*)

The following records contain ship spill information.

literal	C*1	' '.
VSH	R*4	Total tank capacity range (0.> <1000000.) (See next field for units.)
literal	C*9	' VSH-SHIP'.

Record 127: Variable length
FORMAT (*)

literal	C*1	' '.
KSH1	I*4	Units of volume (range 1 to 6). 1 = gallons (U.S.) 2 = barrels 3 = ft^3 4 = m^3 5 = liters 6 = gallons (U.K.)
literal	C*5	' KSH1'.

Record 128: Variable length
FORMAT (*)

literal	C*1	' '.
LSH	R*4	Length of tank (see next field for units).
literal	C*4	' LSH'.

Record 129: Variable length
FORMAT (*)

literal	C*1	' '.
KSH2	I*4	Units of length (range 1 to 2). 1= feet 2= meters
literal	C*5	' KSH2'.

Record 130: Variable length
FORMAT (*)

literal	C*1	' '.
WSH	R*4	Tank width (range 0.> <100.) (See next field for units.)
literal	C*4	' WSH'

Record 131: Variable length
FORMAT (*)

literal	C*1	' '.
KSH3	I*4	Units of length (range 1 to 2). 1 = feet 2 = meters
literal	C*4	' KSH3'.

Record 132: Variable length
FORMAT (*)

literal	C*1	' '.
HDSH	R*4	Distance from hole to top of tank (range 0.> <1000.) (See next field for units.)
literal	C*5	' HDSH'.

Record 133: Variable length
FORMAT (*)

literal	C*1	' '.
KSH4	I*4	Units of length (range 1 to 2). 1= feet 2= meters
literal	C*5	' KSH4'.

Record 134: Variable length
FORMAT (*)

literal	C*1	' '.
WDSH	R*4	Distance from water line to top of tank (range 0.> <100.) (See next field for units.)
literal	C*5	' WDSH'.

Record 135: Variable length
FORMAT (*)

literal	C*1	' '.
KSH5	I*4	Units of length (range 1 to 2). 1= feet 2= meters
literal	C*5	' KSH5'

.Record 136: Variable length
FORMAT (*)

literal	C*1	' '.

	SHSIZE	R*4	Diameter of hole (range 0.> <100.) (See next field for units.)
	literal	C*7	' SHSIZE'.
Record 137:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KSHDESCR	I*4	Units of length (range 1 to 2). 1= inches 2= centimeters
	literal	C*9	' KSHDESCR'.
Record 138:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KSHEME	I*4	Code corresponding to type of leak (range 1 - 4). 1= Crack on wall (0.5"d) 2= Puncture hole (2"d) 3= Fill pipe rupture (4"d) 4= Entire tank contents
	literal	C*7	' KSHEME'.
Record 139:	Variable length FORMAT (*)		
	literal	C*1	' '.
	ZLSH	R*4	Distance from water line to top of tank (range 0.> <1000.) (See next field for units.)
	literal	C*5	' ZLSH'.
Record 140:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KZLSH	I*4	Units of length range (1 - 2). 1= feet 2= Meters
	literal	C*6	' KZLSH'.
Record 141:	Variable length FORMAT (*)		
	literal	C*1	' '.
	IROADRAL	I*4	Code for tanker type (1 - 2).

			1= Road tanker 2= Rail tanker
	literal	C*16	' IROADRAL-TANKER'.

Record 142: Variable length
FORMAT (*)

The following records contain water condition information.

literal	C*1	' '.
ACC_LOC	I*4	Code corresponding to the type of water body in which the release occurred (range 1 - 5). 1 = River/Urban 2 = River/Rural 3 = Coastal Zone 4 = Port Area 5 = Open Water
literal	C*14	' ACC_LOC-WATER'.

Record 143: Variable length
FORMAT (*)

literal	C*1	' '.
WTEMP	R*4	Water temperature (range -10.> <100.) (See next field for units.)
literal	C*6	' WTEMP'.

Record 144: Variable length
FORMAT (*)

literal	C*1	' '.
KW1	I*4	Units of temperature (range 1 to 3). 1 = Fahrenheit 2 = Celsius 3 = Kelvin
literal	C*4	' KW1'.

Record 145: Variable length
FORMAT (*)

literal	C*1	' '.
UCURR	R*4	Velocity of water current (range 0.> <10.) (See next field for units.)
literal	C*6	' UCURR'.

Record 146: Variable length
FORMAT (*)

	literal	C*1	' '.
	KW2	I*4	Units of velocity (range 1 to 5). 1 = mph 2 = knots 3 = ft/s 4 = m/s 5 = km/hr
	literal	C*4	' KW2'.
<u>Record 147</u>:	Variable length FORMAT (*)		
	literal	C*1	' '.
	WDEP	R*4	Depth of water at accident site (range 0.> <60.) (See next field for units.)
	literal	C*5	' WDEP'.
<u>Record 148</u>:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KW3	I*4	Units of length (range 1 to 2). 1 = ft 2 = meters
	literal	C*4	' KW3'.
<u>Record 149</u>:	Variable length FORMAT (*)		
	literal	C*1	' '.
	WDTH	R*4	Average width of body of water at accident site (> 0.0) (See next field for units.)
	literal	C*5	' WDTH'.
<u>Record 150</u>:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KW4	I*4	Units of length (range 1 to 2). 1 = ft 2 = meters
	literal	C*4	' KW4'.
<u>Record 151</u>:	Variable length FORMAT (*)		
	literal	C*1	' '.

	SP_LOC	I*4	Code corresponding to the approximate location of the release (range 1 - 3). 1 = Mid-river 2 = Near shore 3 = Shoreline
	literal	C*7	' SP_LOC'.
Record 152:	Variable length FORMAT (*)		
	literal	C*1	' '.
	MGR	R*4	Mass of explosive released (> 0.0) (kg).
	literal	C*11	' MGR-EXGMAS'.
Record 153:	Variable length FORMAT (*)		
	literal	C*1	' '.
	DBPC	R*4	Maximum diameter of fire (meters).
	literal	C*5	' DBPC'.
Record 154:	Variable length FORMAT (*) The following records contain information on tanks.		
	literal	C*1	' '.
	ISHAPE	I*4	A flag which indicates the type of tank. 0: Circular cross section & Axis horizontal (Rail & Road Tankers, Bullet Tanks) 1: Tank cross section is rectangular (Barge and ship tanks) 2: Circular cross section, Axis vertical (Large liquid storage tanks) 3: Pipeline
	literal	C*14	' ISHAPE-TNKDIM'.
Record 155:	Variable length FORMAT (*)		
	literal	C*1	' '.
	TNKD	R*4	Tank diameter in meters (given only for circular tanks).
	literal	C*5	' TNKD'.
Record 156:	Variable length FORMAT (*)		

	literal	C*1	' '.
	TNKW	R*4	Mean tank width - meters.
	literal	C*5	' TNKW'.

Record 157: Variable length
FORMAT (*)

literal	C*1	' '.
TNKH	R*4	Depth or height of inside of tank - meters.
literal	C*5	' TNKH'.

Record 158: Variable length
FORMAT (*)

literal	C*1	' '.
TNKL	R*4	Length of tank - meters.
literal	C*5	' TNKL'.

Record 159: Variable length
FORMAT (*)

literal	C*1	' '.
TNKVOL	R*4	Volume of tank - cubic meters.
literal	C*7	' TNKVOL'.

Record 160: Variable length
FORMAT (*)

literal	C*1	' '.
TNKTSI	R*4	Temperature of the tank (K).
literal	C*7	' TNKTSI'.

Record 161: Variable length
FORMAT (*)

literal	C*1	' '.
TNKPSI	R*4	Pressure inside tank (N/m**2).
literal	C*12	' TNKPSI-TNKC'.

Record 162: Variable length
FORMAT (*)

literal	C*1	' '.
DH	R*4	Equivalent diameter of the hole (meters).
literal	C*11	' DH-HOLEDIM'.

Record 163: Variable length
FORMAT (*)

literal	C*1	' '.
HH	R*4	Hole Height from tank bottom (meters).
literal	C*3	' HH'.

Record 164: Variable length
FORMAT (*)

literal	C*1	' '.
HLI	R*4	Initial depth of liquid of chemical in tank (meters).
literal	C*11	' HLI-CLEVEL'.

Record 165: Variable length
FORMAT (*)

literal	C*1	' '.
ZW	R*4	Height above ground at which wind is measured (meters).
literal	C*3	' ZW'.

Record 166: Variable length
FORMAT (*)

literal	C*1	' '.
IDIKE	I*4	Flag indicating whether tank has a dike around it. 0: No dike 1: Dike exists around tank.
literal	C*6	' IDIKE'.

Record 167: Variable length
FORMAT (*)

literal	C*1	' '.
DDIKE	R*4	Dike diameter (meters).
literal	C*6	' DDIKE'.

Record 168: Variable length
FORMAT (*)

literal	C*1	' '.
HDIKE	R*4	Dike height (meters).
literal	C*6	' HDIKE'.

Record 169: Variable length
FORMAT (*)

literal	C*1	' '.
PCH	R*4	Storage pressure of the vessel (N/m**2).
literal	C*10	' PCH-TNKCH'.

Record 170: Variable length
FORMAT (*)

literal	C*1	' '.
TCH	R*4	Storage temperature of the chemical (K).
literal	C*4	' TCH'.

Record 171: Variable length
FORMAT (*)

literal	C*1	' '.
ISAT	I*4	A flag that indicates the condition of chemical in the tank. 0: Chemical in SUBCOOLED state 1: Chemical in SATURATED state (equbm liq/vap) 2: Chemical in SUPERHEATED state (gaseous)
literal	C*5	' ISAT'.

Record 172: Variable length
FORMAT (*)

literal	C*1	' '.
IPHASE	I*4	A flag indicating the phase of chemical in the tank. 0: COMPRESSED GAS state 1: Liquid state 2: Liquified compressed gas.
literal	C*7	' IPHASE'.

Record 173: Variable length
FORMAT (*)

literal	C*1	' '.
ITNKP	I*4	A flag indicating whether the tank (chemical) pressure is provided or not. 0: Tank pressure not given by the user 1: Tank pressure is given by the user
literal	C*15	' ITNKP-ITNKCOND'.

Record 174: Variable length
FORMAT (*)

literal	C*1	' '.
ITNKT	I*4	A flag indicating whether the tank (chemical) temperature is provided or not. 0: Tank temperature not given by the user 1: Tank temperature is given by the user
literal	C*6	' ITNKT'.

Record 175: Variable length
FORMAT (*)

literal	C*1	' '
PATM	R*4	Atmospheric pressure, (N/m**2)
literal	C*13	' PATM-ATMCOND'

Record 176: Variable length
FORMAT (*)

	literal C*1 ' '.	
TA	R*4	Atmospheric temperature (K).
literal	C*3	' TA'.

Record 177: Variable length
FORMAT (*)

literal	C*1	' '.
RH	R*4	Relative humidity of the air (%).
literal	C*3	' RH'.

Record 178: Variable length
FORMAT (*)

The following record types contain information on fire model specific parameters.

literal	C*1	' '.
HTARGC	R*4	Height of target relative to base of the tank. (See next record for units.)
literal	C*7	' HTARGC'.

Record 179: Variable length
FORMAT (*)

literal	C*1	' '.
KTRGC	I*4	Code corresponding to units of length. 1: feet

			2: meters
	literal	C*6	' KTRGC'.
Record 180:	Variable length FORMAT (*)		
	literal	C*1	' '.
	OBJAN	I*4	Code corresponding to the orientation of the target. 1: Horizontal 2: Vertical 3: To receive maximum flux
	literal	C*6	' OBJAN'.
Record 181:	Variable length FORMAT (*)		
	literal	C*1	' '.
	FLX1	R*4	Fire radiation heat flux level 1 - (Range 0.> <1500.) (See next record for units.)
	literal	C*12	' FLX1-FLUXEM'.
Record 182:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KFL1	I*4	Code corresponding to the units of heat flux. 1: Kw/m**2 2: BTU/s.ft**2
	literal	C*5	' KFL1'.
Record 183:	Variable length FORMAT (*)		
	literal	C*1	' '.
	FLX2	R*4	Same as FLX1 for level 2 (Range 0.> <1500.) (See next record for units.)
	literal	C*5	' FLX2'.
Record 184:	Variable length FORMAT (*)		
	literal	C*1	' '
	KFL2	I*4	Code corresponding to the units of heat flux. 1: kW/m**2 2: BTU/s.ft**2
	literal	C*5	' KFL2'.

Record 185: Variable length
FORMAT (*)

literal	C*1	' '.
FLX3	R*4	Same as FLX1 for level 3 (Range 0.> <1500.) (See next record for units.)
literal	C*5	' FLX3'.

Record 186: Variable length
FORMAT (*)

literal	C*1	' '.
KFL3	I*4	Code corresponding to the units of heat flux. 1: kW/m**2 2: BTU/s.ft**2
literal	C*5	' KFL3'.

Record 187: Variable length
FORMAT (*)

literal	C*1	' '.
EXSEL	I*4	Code corresponding to the type of explosion. 1: Tank explosion 2: Gas Cloud Explosion
literal	C*12	' EXSEL-EXTYP'

Record 188: Variable length
FORMAT (*)

literal	C*1	' '.
BISEC	R*4	Initial guess at bisection value for iterative solution.
literal	C*11	' BISEC-TOLE'.

Record 189: Variable length
FORMAT (*)

The following record types contain output contour specifications.

literal	C*1	' '.
NO_CONT	I*4	Number of contours to produce (range 1 - 3).
literal	C*22	' NO_CONT-DISP_OUT_SPEC'.

Record 190: Variable length
FORMAT (*)

	literal	C*1	' '.
	NO_1	I*4	Number corresponding to contour of interest for contour 1. 1: TLV - Threshold Limit Value 2: TLVTWA - Threshold Limit Time Weighted 3: LFL - Lower Flammability Limit 4: Other - User Input Concentration
	literal	C*5	' NO_1'.
Record 191:	Variable length FORMAT (*)		
	literal	C*1	' '.
	VNO_1	R*4	Concentration value for contour 1. (See next record for units.)
	literal	C*6	' VNO_1'.
Record 192:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KVNO_1	I*4	Code for concentration units 1: ppm 2: kg/m**3 3: mg/m**3
	literal	C*7	' KVNO_1'.
Record 193:	Variable length FORMAT (*)		
	literal	C*1	' '.
	NO_2	I*4	Number corresponding to contour of interest for contour 2. (see NO_1, Record 190)
	literal	C*5	' NO_2'
Record 194:	Variable length FORMAT (*)		
	literal	C*1	' '.
	VNO_2	R*4	Concentration value for contour 2. (See next record for units.)
	literal	C*6	' VNO_2'.
Record 195:	Variable length FORMAT (*)		
	literal	C*1	' '.

	KVNO_2	I*4	Code for concentration units. (See KVNO_1)
	literal	C*7	' KVNO_2'.
Record 196:	Variable length FORMAT (*)		
	literal	C*1	' '.
	NO_3	I*4	Number corresponding to contour of interest for contour 3. (See NO_1, Record 190.)
	literal	C*5	' NO_3'.
Record 197:	Variable length FORMAT (*)		
	literal	C*1	' '.
	VNO_3	R*4	Concentration value for contour 3. (See next record for units.)
	literal	C*6	' VNO_3'.
Record 198:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KVNO_3	I*4	Code for concentration units (see KVNO_1).
	literal	C*7	' KVNO_3'.
Record 199:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KDISP_U	I*4	Code corresponding to output units. 1: English - FPS = Foot Pound Seconds 2: Metric - MKS = Meters Kilograms Seconds
	literal	C*8	' KDISP_U'.
Record 200:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KUNIT	I*4	Code corresponding to plotting units (in tabular reports). 1: feet 2: meters 3: kilometers 4: miles
	literal	C*16	' KUNIT - EXPUNIT'.

Record 201: Variable length
FORMAT (*)

The following records contain target location data.

literal	C*1	' '.
XLOC	R*4	Level of concern. (See next record for units.)
literal	C*19	' [XLOC] - WD_TARGET'.

Record 202: Variable length
FORMAT (*)

literal	C*1	' '.
KWLOC	I*4	Code corresponding to plotting units (in tabular reports). 1: kg/m**3 2: ppm 3: mg/l
literal	C*8	' [KWLOC]'.

Record 203: Variable length
FORMAT (*)

literal	C*1	' '.
XCODE	I*4	Code corresponding to whether the marking on the distance marker is provided or whether the absolute distance from spill location to target is entered. 1: Distance Marker 2: Actual Distance
literal	C*8	' [XCODE]'.

Record 204: Variable length
FORMAT (*)

literal	C*1	' '.
XDIST1	R*4	Distance from spill to target. (See next record for units.)
literal	C*9	' [XDIST1]'.

Record 205: Variable length
FORMAT (*)

literal	C*1	' '.
KDIST1	I*4	Code corresponding to unit of downstream distance. 1: kilometers

			2: miles
	literal	C*9	' [KDIST1]'.
Record 206:	Variable length FORMAT (*)		
	literal	C*1	' '.
	YCODE1	I*4	Code corresponding to cross stream spill location. 1: Left Bank 2: Towards Left Bank 3: Mid River 4: Towards Right Bank 5: Right Bank 6: Other
	literal	C*9	' [YCODE1]'.
Record 207:	Variable length FORMAT (*)		
	literal	C*1	' '.
	YDIST1	R*4	Cross stream location of target. (See next record for units.)
	literal	C*9	' [YDIST1]'.
Record 208:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KYD1	I*4	Code corresponding to unit of cross stream location. 1: feet 2: meters
	literal	C*7	' [KYD1]'.
Record 209:	Variable length FORMAT (*)		
	literal	C*1	' '.
	DCODE1	I*4	Code corresponding to depth at which spill occurred. 1: surface 2: mid depth 3: bottom 4: other
	literal	C*9	' [DCODE1]'.

Record 210: Variable length
FORMAT (*)

literal	C*1	' '.
DEPTH1	R*4	Depth of target. (See next record for units.)
literal	C*9	' [DEPTH1]'.

Record 211: Variable length
FORMAT (*)

literal	C*1	' '.
KDEPTH1	I*4	Code corresponding to unit of depth. 1: feet 2: meters
literal	C*10	' [KDEPTH1]'.

Record 212: Variable length
FORMAT (*)

The following records contain spill data.

literal	C*1	' '.
XMM0	R*4	Marking on nearest distance marker - 0.0 if unknown (see next record for units) range (0.0>)
literal	C*7	' [XMM0]'.

Record 213: Variable length
FORMAT (*)

literal	C*1	' '.
KXMM	I*4	Code corresponding to unit of distance. 1: kilometers 2: miles
literal	C*7	' [KXMM]'.

Record 214: Variable length
FORMAT (*)

literal	C*1	' '.
YCODE0	I*4	Code for cross stream location of spill. 1: Left Bank 2: Towards Left Bank 3: Mid River 4: Towards Right Bank 5: Right Bank 6: Other

	literal	C*9	' [YCODE0]'.
Record 215:	Variable length FORMAT (*)		
	literal	C*1	' '.
	YDIST0	R*4	Cross stream location of spill. (See next record for units.)
	literal	C*9	' [YDIST0]'.
Record 216:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KYD0	I*4	Code corresponding to unit of cross stream distance. 1: feet 2: meters
	literal	C*7	' [KYD0]'.
Record 217:	Variable length FORMAT (*)		
	literal	C*1	' '.
	DCODE0	I*4	Code corresponding to depth at which spill occurred. 1: surface 2: mid depth 3: bottom 4: other
	literal	C*9	' [DCODE0]'.
Record 218:	Variable length FORMAT (*)		
	literal	C*1	' '.
	DEPTH0	R*4	Depth of spill. (See next record for units.)
	literal	C*9	' [DEPTH0]'.
Record 219:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KDEPTH0	I*4	Code corresponding to unit of depth. 1: feet 2: meters
	literal	C*10	' [KDEPTH0]'.

Record 220:	Variable length FORMAT (*)		
	literal	C*1	' '.
	MRF	I*4	Code corresponding to the nature of the river (Manning Roughness Factor). 1: Clean, straight bank, full stage (0.030) 2: Winding, some pools and shoals (0.040) 3: Same as 2, but with stony sections (0.050) 4: Sluggish reaches, deep pools weeds (0.070) 5: Same as 4, but more (0.095) 6: Same as 4, but extreme (0.125)
	literal	C*6	' [MRF]'.
Record 221:	Variable length FORMAT (*)		

The following records contain information pertaining to property calculations of a chemical.

	literal	C*1	' '.
	LCH1	R*4	Temperature at which properties are to be calculated (range (-459 >= < 310.) (See next record for units.)
	literal	C*19	' lch1 - p13_calchem'.
Record 222:	Variable length FORMAT (*)		
	literal	C*1	' '.
	KCH1	I*4	Code corresponding to unit of temperature. 1: Fahrenheit 2: Celsius 3: Kelvin 4: Rankine
	literal	C*5	' kch1'.
Record 223:	Variable length FORMAT (*)		
	literal	C*1	' '.
	NCH1	I*4	Code corresponding to unit of result display. 1: Systeme Internationale (SI) 2: Foot-Pound-Second (FPS) 3: Centimeter-Gram-Second (CGS)
	literal	C*5	' nch1'.

Record 224: Variable length
FORMAT (*)

literal	C*1	' '.
TYPE	I*4	Code corresponding to type of release. 0: Instantaneous 1: Continuous
literal	C*20	' type-reltype common'.

Record 225: Variable length
FORMAT (*)

literal	C*1	' '.
IDOSE	I*4	This variable is set to 0 by default input case and is not modified by any chems software.
literal	C*26	' idose - vapor_disp common'.

Record 226: Variable length
FORMAT (*)

literal	C*1	' '.
AVGTIM	R*4	This variable is set to 15. by default input case and is not modified by any chems software.
literal	C*26	' AVGTIM'

Record 227: Variable length
FORMAT (*)

literal	C*1	' '.
XDOSE	R*4	This variable is set to 0. by default input case and is not modified by any chems software.
literal	C*6	' xdose'.

Record 228: Variable length
FORMAT (*)

literal	C*1	' '.
IWINDIR	I*4	This variable is set to 0 by default input case and is not modified by any chems software.
literal	C*8	' iwindir'.

Record 229: 18 bytes
FORMAT (' ',A1,' atmclc-atm_info')

The following records contain atmospheric information.

ATMCLC	C*1	"Y" If overriding atmospheric and calculating stability parameter and wind friction velocity based on this. "N" if not.

Record 230: Variable length
`FORMAT (*)`

literal	C*1	' '.
STAB	R*4	Stability parameter best describing the atmospheric conditions (range 0.5<= x <= 5.5).
literal	C*5	' stab'.

Record 231: 6 bytes
`FORMAT (' ',A1,' avg')`

AVG	C*1	"Y" If calculating stability parameter based on statistical data. "N" if not.

Record 232: Variable length
`FORMAT (*)`

literal	C*1	' '.
STH	R*4	Sigma theta (degrees) (range 0. <= x < 360.)
literal	C*4	' sth'.

Record 233: Variable length
`FORMAT (*)`

literal	C*1	' '.
TWA	R*4	Wind averaging time (minutes) (range 0. < x < 60.)
literal	C*4	' twa'.

Record 234: Variable length
`FORMAT (*)`

literal	C*1	' '.
CTR	R*4	Cloud transmittance (range 0. <= x <= 1.)
literal	C*4	' ctr'.

Record 235: Variable length
`FORMAT (*)`

literal	C*1	' '.
BETA	R*4	POOLXY beta
literal	C*17	' beta-beta common'

Filename:	**CnnnCHK.DAT**
Directory:	SITEIMESORAD:, IEMIS$SYSF:
Purpose:	This file contains checkpoints (locations where dose levels will be monitored) for the MESORAD model. The version in IEMIS$SYSF:, C000CHK.DAT, is a template.
Access:	1. MES_(NO)GR->Update Case Data->Change Checkpoints
Include:	EESF$INC:MESOI_FORM_REC.INC
Structure:	ASCII, Sequential.
Record 1:	80 bytes FORMAT (A)

HEADER	C*80	File header.

Record K (K = 2, N*2, 2): 23 bytes

```
FORMAT (1X,F6.2,1X,F6.2,1X,A8)
```

CKX	R*4	Checkpoint X location from grid center in kilometers.
CKY	R*4	Checkpoint Y location from grid center in kilometers.
CKNAME	C*8	Checkpoint name.

Record L (L = 3, N*2 +1, 2): 66 bytes

```
FORMAT (3(2X,2F10.5))
```

CKCOV1	R*4	Low level concentration checkpoint.
CKCOV2	R*4	High level concentration checkpoint.
CKTHV1	R*4	Low level concentration checkpoint.
CKTHV2	R*4	High level concentration checkpoint.
CKWBV1	R*4	Low level concentration checkpoint.
CKWBV2	R*4	High level concentration checkpoint.

Filename: **CnnnCOD.RPT**

Directory: SITEODYNEV:

Purpose: This file provides tabular reports and diagnostics to users of the IDYNEV evacuation model and input to IBS evacuation graphic functions for case nnn.

Access:
1. IBS -> EVACUATION
2. IDYNEV_(NO)GR

Include:
EESF$INC:GLOBAL.INC
EESF$INC:DATPRM.INC
EESF$INC:INTDAT.INC
EESF$INC:LEV.INC
EESF$INC:TADATA.INC

Notes: The IDYNEV model produces this file. IDYNEV has considerable flexibility built into it in order to make it useful for many applications. IBS is one of these applications. The number and, to some extent, the type of output tables included in this report are under control of the user (see record types 04 and 05 in CnnnINP.DAT). The source code installed for IBS is the reference used to determine what output could be written to CnnnCOD.RPT. In practice, some of the output described in this document may never be included in an IBS-generated file.

The report can contain up to 26 report types of tables and other output ordered into four groups: a banner page, input processing, trip distribution/traffic assignment, and traffic simulation. The report types match, to the extent possible, the descriptions in the updated *User's Guide for the IDYNEV System*. Some of the potential outputs of the model were not included in that document, but are included here. In addition, nearly 200 coded diagnostic messages and some self-explanatory diagnostics may be included in the CnnnCOD.RPT.

Structure: ASCII, sequential, 134 bytes/record

DYNEV Reports Generated in the CnnnCOD.RPT file			When Produced			
Report Code	Description	Position In File	Always	TA Model	Simulation	Optional
**)	Time Period Banner	**			X	
0)	Banner	1	X			
1)	Run Identification Report, Part 1	2	X			
2)	Card File List Report	3	X			
3)	Run Identification Report, Part 2, plus Run Control Data	4	X			
4)	Traffic Assignment Parameters Table	11		X		
5)	Trip Productions Table, Attractions and Path Links	12		X		
6)	Traffic Assignment Results	13*		X		
7)	Characteristics of I-DYNEV Links	5	X			
8)	Turn Operations Data and Blockage Factors	6	X			
9)	Node Coordinate Table	7	X			
10)	Traffic Control Table	8	X			
11)	Interpretation of Signal Codes	9	X			
12)	Traffic control table - Signs and Fixed Time Signals	10	X			
13)	Entry Link Volumes	11***			X	X
14)	Source/Sink Volumes	12***			X	

DYNEV Reports Generated in the CnnnCOD.RPT file (cont.)			When Produced			
Report Code	Description	Position In File	Always	TA Model	Simulation	Optional
15)	Initialization Statistics	13			X	
16)	Link and Subnetwork Statistics	14****			X	
17)	Polar Sector Statistics	15****			X	X
18)	Cartesian Sector Statistics	16****			X	X
19)	Person Measures of Effectiveness	17****			X	
20)	Link Data for Time Periods Other than 1 Table	??			X	X
21)	Evacuation Table Reflecting Data on the Type 49 Record					X
22)	Load Factor Table Reflecting Data on the Type 52 Record					X
23)	Intermediate Output Report of Measures of Effectiveness				X	X
24)	Link Capacity Factors for Time Interval					X
25)	Bus Statistics (Cumulative and Final)				X	X
26)	Run Terminated Message				X	

* 6 may repeat, subject to convergence algorithm.
** May repeat in SIMULATION output, subject to time periods modeled.
*** Produced for each time period during initialization.
**** Produced for each cumulative time period during simulation (see record type 05 field NTI_OUT_CUM).
?? Only occurs for time periods 2 or later which include type 11 records in input.

REPORT DESCRIPTIONS

The 26 report types are described on the following pages. Each report type will begin a new page with the report type number and name in the first two lines on the page.

The record numbers within report types are coded with a sequence number relative to the start of its respective report type. For example, the 5th record in the Polar Sector Statistics section is coded RECORD 5.

In some cases, more than one table is provided within a single report type; for example, report type 5 of Polar Sector Statistics includes a trip table, internal centroid table, trip attraction table, and path network links table. When more than one table is included in a single report type, record descriptions are preceded by a short description.

Some reports are defined in more than one way. This happens when different formats are used, depending on the type of simulation being performed. For example, TRAF traffic assignment runs use different input and produce different output tables than TRAD traffic assignment runs.

Some record types are only applicable to cases that are outside the area of interest to IBS, such as information on fuel consumption and emission rates. Such information is not generally included in this document. Similarly, IBS does not provide for stacking of multiple cases within a single simulation run. No attempt is made to show the effect on report structure of multiple cases within a single report.

Report type **
Time-Period Banner

Record 1-5:

```
FORMAT (1H1,131(1H*),//50X,11HTIME PERIOD,I4,
8H - LEVEL,I3,5H DATA,//1H ,131(1H*))
OR
FORMAT (1H1,131(1H*),//51X,11HTIME PERIOD,I4,
15H - FREEWAY DATA,//1H ,131(1H*))
OR
FORMAT (1H1,131(1H*),//49X,11HTIME PERIOD,I4,
14H - NETSIM DATA,//1H ,131(1H*))
OR
FORMAT (1H1,131(1H*),//42X,'TIME PERIOD',I4,
' - IDYNEV DATA FOR ','SUBNETWORK',I4,//' ',131('*'))
```

TPCNT	I*4	Time period
SUBNUM	I*4	Subnetwork number
INUM	I*4	Network level

Report type 00
Banner Report

Record 1-26:

```
FORMAT (1H1, //////////////////////////)
```

Record 27-38:

```
FORMAT (27X,A76)
```

IL	C*76	Line of banner. 12 lines are used to spell IDYNEV by placing each letter so that 12-line high letters are formed.

Report type 01
Run Identification Report, Part 1

Record 1-7: If not an evacuation case, then the following:

```
FORMAT (1H1,53X,21HTRAF SIMULATION MODEL,//
59X,13HDEVELOPED FOR,//
49X,34HU. S. DEPARTMENT OF TRANSPORTATION,/
51X,30HFEDERAL HIGHWAY ADMINISTRATION,/
54X,24HTRAFFIC SYSTEMS DIVISION)
```

If an evacuation case, then the following:

```
FORMAT (1H1,54X,23HIDYNEV EVACUATION MODEL, //,
60X, 12HDEVELOPED BY, //
55X, 23HK L D  ASSOCIATES, INC., /,
60X, 12H300 BROADWAY, /,
52X, 29HHUNTINGTON STATION, NY  11746)
```

Record 8-12:

```
FORMAT (1H0,131(1H*),//59X,14HSTART OF CASE,I3,//1H, 131(1H*))
```

CASE	I*4	Case number

Report type 02
Card File List Report

Record 1-2:

```
FORMAT (1H1,33X,14HCARD FILE LIST,/)
```

Record 3-(2+n):

```
FORMAT (1H,A80)
```

CCBUF C*80 Case input records (up to 20000 records)

Report type 03
Run Identification Report, Part 2; Plus Run Control Data

Record 1-7: If not an evacuation case, then the following:

```
FORMAT (1H1,55X,21HTRAF SIMULATION MODEL,//
59X,13HDEVELOPED FOR,//
49X,34HU. S. DEPARTMENT OF TRANSPORTATION,/
51X,30HFEDERAL HIGHWAY ADMINISTRATION,/
54X,24HTRAFFIC SYSTEMS DIVISION)
```

If an evacuation case, then the following:

```
FORMAT (1H1,54X,23HIDYNEV EVACUATION MODEL,//,
60X,12HDEVELOPED BY,//
55X,23HK L D  ASSOCIATES, INC.,/,
60X,12H300 BROADWAY,/,
52X,29HHUNTINGTON STATION, NY  11746)
```

Record 8-12:

```
FORMAT (1H0,131(1H*),//59X,14HSTART OF CASE ,I3,//1H,131(1H*))
```

CASE	I*4	Case number

Record 13-(12+n):

```
FORMAT (1H0,26X,A72)
```

CCBUF(1:72)	C*80	Case input record of type 00. A maximum of 50 such records may exist.

Record 13+n-16+n:

```
FORMAT (1H0,///,51X,6HDATE =,A4,2(1H/,A4))
```

CCBUF(37:40)	C*80	Month from case input record of type 01
CCBUF(41:44)	C*80	Day from case input record of type 01
CCBUF(45:48)	C*80	Year from case input record of type 01

Record 17+n:

```
FORMAT (1H0,50X,7HUSER = ,A36)
```

CCBUF(1:36)	C*80	User name from case input record of type 01

Record 18+n:

```
FORMAT (1H0,48X,9HAGENCY = ,A24)
```

CCBUF(49:72)	C*80	Agency name from case input record of type 01

Record (19+n)-(24+n):

```
FORMAT (1H1,55X,16HRUN CONTROL DATA,///,
22X,5HVALUE,25X,26HRUN PARAMETERS AND OPTIONS,/)
```

Record 25+n:

```
FORMAT (1H0,21X,I5,5X,25HRUN IDENTIFICATION NUMBER)
```

NRUN	I*4	Run identification number.

Record 26+n:

```
FORMAT (1H0,I26,5X,33HNEXT CASE CODE = (0,1) IF ANOTHER,
29H CASE (DOES NOT, DOES) FOLLOW)
```

NEXTRN	I*4	Next case code (0,1), If another case (does not, does) follow.

Record (27+n)-(29+n):

```
FORMAT (1H0,I26,5X,26HRUN TYPE CODE = (1, 2) IF ,
31H(SIMULATION, TRAF. ASSIGNMENT) , 14HTO BE EXECUTED,/
48X,38H(-1, -2) IF ONLY DIAGNOSTICS ARE BEING,
27H PERFORMED ON (SIMULATION, ,/
48X,22HTRAF. ASSIGNMENT) DATA)
```

TYPERN	I*4	Run type code (-2,-1,1,2). See format for explanation.

Record 30+n:

```
FORMAT (1H0,I26,5X,35HCLOCK TIME AT START OF SIMULATION, ,
17HHOURS AND MINUTES)
```

BEGTME	I*4	Simulation start time (HHMM) no punctuation.

Record 31+n:

```
FORMAT (1H0,I26,5X,32HTRIP ASSIGNMENT AND DISTRIBUTION,
51H CODE = (0,1) IF INTEGRATION (IS NOT, IS) REQUESTED)
```

XBUF(17)	I*4	See format statement.

Record 32+n:

```
FORMAT (1H0,I26,5X,18HRANDOM NUMBER SEED)
```

XSEED	I*4	See format statement.

Record (33+n)-(34+n+j):

```
FORMAT (1H0,19(/,I27,5X,33HDURATION (SEC) OF TIME PERIOD NO.,I3))
```

(LNTMPR(I), I, I = 1, J)	I*4	Time period lengths for all time periods

Record 35+n+j:

```
FORMAT (1H0,I26,5X,34HLENGTH OF A TIME INTERVAL, SECONDS)
```

LENINT I*4 See format statement.

Record 36+n+j:

```
FORMAT (1H0,I26,5X,29HMAXIMUM INITIALIZATION TIME, ,
24HNUMBER OF TIME INTERVALS)
```

TMINIT I*4 See format statement.

Record 37+n+j:

```
FORMAT (1H0,I26,5X,25HNUMBER OF TIME INTERVALS ,
35HBETWEEN SUCCESSIVE STANDARD OUTPUTS)
```

PUTSTD I*4 See format statement.

Record 38+n+j:

```
FORMAT (1H0,I26,5X,25HNUMBER OF TIME INTERVALS ,
39HBETWEEN SUCCESSIVE INTERMEDIATE OUTPUTS,
23H FOR MACROSCOPIC MODELS)
```

PUTINT I*4 See format statement.

Report type 4
Traffic Assignment Parameters Table

Record 1-5:

```
FORMAT (1H1,48X,27HAVERAGE VEHICLE OCCUPANCIES,/,
47X,34H(HUNDREDTHS-OF-A-PERSON / VEHICLE),//,
45X,34HAUTOS   CAR-POOLS   TRUCKS   BUSES, /,
39X,4I10)
```

VEHOCC(1)	I*4	See format statement.

Record 6-18:

```
FORMAT (1H1,50X,29HTRAFFIC ASSIGNMENT PARAMETERS, ///,
41X,18HEPSILON = 10 ** (-, I3, 1H), A1,//,
41X,28HMAXIMUM NO. OF ITERATIONS = ,I2,A1,//,
41X,42HCOEFFICIENT -A- IN BPR IMPEDANCE EQ. X 100,
3H = ,I2,A1,//,
41X,41HEXPONENT -B- IN BPR IMPEDANCE EQ. X 10 = ,
I3,A1,///,53X,25H* INDICATES DEFAULT VALUE)
```

EPSLON	I*4	See format statement.
KKDEF(1:1)	C*1	See format statement.
MAXITA	I*4	See format statement.
KKDEF(2:2)	C*1	See format statement.
MBPRA	I*4	See format statement.
KKDEF(3:3)	C*1	See format statement.
MBPRB	I*4	See format statement.
KKDEF(4:4)	C*1	See format statement.

Record type 5
Trip Productions Table

If this is a TRAD traffic assignment run, then

Record 1-7:

```
FORMAT (1H1,60X,10HTRIP TABLE,//,22X,
43HFOR EACH ORIGIN NODE TABLE PROVIDES LISTING,
43H OF VOLUME AND ASSOCIATED DESTINATION NODES, ///,
15X,'ORIGIN',5X,'ORIGIN', /,
15X,' NODE ',2X,'VOLUME (VPH)', 25X,'DESTINATION NODES')
```

Record 8-(7+n*x):

```
FORMAT (/,16X,I4,7X,I4,9X,14I6,(/,40X,14I6))
```

KK	I*4	Origin node (centroid) number
SCRCHA(1)	I*4	Volume (Vehicles per hour from source node KK)
(SCRCHB(I),I=1,IDN)	I*4	Destination nodes

This record type is repeated for each origin node and will occupy as many physical records as are necessary, depending on the value of IDN.

If this is a TRAF traffic assignment run, then

Record 1-3:

```
FORMAT (1H1,60X,10HTRIP TABLE, //,22X,
43HFOR EACH ORIGIN NODE TABLE PROVIDES LISTING,
44H OF PAIRS OF DATA :  DESTINATION NODE/VOLUME)
```

Record 4-7:

```
FORMAT (///,56X,14HORIGIN NODE = ,I4)
```

KK I*4 Origin node (centroid) number

Record 8-(7+n*x):

```
FORMAT ((//,7(1X,=I4,1H/,I4,5X)))
```

SCRCHB(I)	I*4	Destination node.
SCRCHA(I)	I*4	Volume (Vehicles per hour from source node KK).

This record type is repeated for each origin node and will occupy as many physical records as are necessary, depending on the value of I ranging from 1 to IDN for destination, volume pairs.

Trip Productions Table - (Internal Centroids)

Record 1-4:

```
FORMAT (1H1,56X,18HINTERNAL CENTROIDS,//,
54X,8HCENTROID,7X,11HACCESS LINK,/)
```

Record 5-(4+n):

```
FORMAT (1H,54X,I4,9X,1H(,I4,1H,,I4,1H))
```

CNTRYD(J)	I*4	Internal centroid number
GUPNOD(IL)	I*4	Upstream node of access link
GDWNOD(IL)	I*4	Downstream node of access link

This record type is repeated for each internal centroid.

Trip Productions Table - (Trip Attractions)

Record 1-5: Header, repeated at the top of each page of trip attractions

```
FORMAT (1H1,60X,16HTRIP ATTRACTIONS,//,
43X,11HDESTINATION,27X,10HATTRACTION,/,
46X,4HNODE,34X,5H(VPH),/
,43X,11H-----------,29X,6H------)
```

Record 6-(5+n):

```
FORMAT (1H0,45X,I4,34X,I4)
```

KNODE	I*4	Destination node number
KVOL	I*4	Attraction of KNODE in vehicles per hour

This record type is repeated for each destination node. A page break is provided every 26 records.

Trip Productions Table - (Path Network Links)

Record 1-4:

```
FORMAT (1H1,57X,18HPATH NETWORK LINKS,/,20X,9HGEOMETRIC,/,
22X,4HLINK,14X,4HSINK,11X,5HRIGHT,11X,4HTHRU,11X,
4HLEFT,8X,8HDIAGONAL,/)
```

Record 5-(4+n):

```
FORMAT (19X,1H(,I4,1H,,I4,1H),22X,4(1H(,I4,1H,,I4,1H),4X))
```

IBUF(1)	I*4	Upstream node of entry link
IBUF(2)	I*4	Downstream node of entry link
(KBUF(J), J = 1, 8)	I*4	4 upstream and downstream node pairs of right, through, left, and diagonal receiving links

This record type is repeated for each entry link.

Record (5+n)-(4+n+m):

```
FORMAT (19X,1H(,I4,1H,,I4,1H),7X,5(1H(,I4,1H,,I4,1H),4X))
```

IBUF(1)	I*4	Upstream node of entry link
IBUF(2)	I*4	Downstream node of entry link
(KBUF(J), J = 1, 10)	I*4	5 upstream and downstream node pairs of sink, right, through, left, and diagonal receiving links

This record type is repeated for each internal link.

Trip Productions Table - (Totals)

Record 1-6:

```
FORMAT (22H0ITERATION NUMBER... =, 9X,I7/
22H0TOTAL VEHICLE HOURS =,E16.10/
22H0TOTAL VEHICLE MILES =,E16.10/
22H0AVERAGE SPEED       =,F16.2//)
```

ITER	I*4	Iteration number
VEHRS	R*4	Total vehicle hours
VEMLS	R*4	Total vehicle miles.
VESPD	R*4	Average vehicle speed

Report type 6
Traffic Assignment Results

Record 1-3:

```
FORMAT (1H1,53X,26HTRAFFIC ASSIGNMENT RESULTS, //)
```
or
```
FORMAT (1H1,49X,34HTRAFFIC ASSIGNMENT RESULTS (CONT.), //)
```

Only appears as header of continuation pages, if any are needed, for this table.

Record 4-6:

```
FORMAT (1H ,24X,'INTERNAL    RIGHT TURN      THRU        '
'LEFT TURN     DIAGONAL     SOURCE    SINK  DISCHARGE',/,
14X,51HLINK       CENTROID    VOL.  PCT.    VOL.  PCT.     ,
52HVOL.  PCT.     VOL.  PCT.      FLOW      FLOW     VOLUME,/,
26X,4(11X,3HVPH),7X,3(6X,3HVPH))
```

Record 7-(6+n):

```
FORMAT (10X,1H(,I4,1H,,I4,1H),6X,I4,4X,I5,3X,I3,I8,
I6,I8,I6,I8,I6,1X,3I9)
```

IBUF(1)	I*4	Upstream node of source link or link associated with internal centroid IBUF(5)
IBUF(2)	I*4	Downstream node of source link or link associated with internal centroid IBUF(5)
IBUF(5)	I*4	Internal centroid
ITURN(1)	I*4	Volume (vph) taking right turn
ITNPCT(1)	I*4	Volume (%) taking right turn
ITURN(2)	I*4	Volume (vph) taking thru movement
ITNPCT(2)	I*4	Volume (%) taking thru movement
ITURN(3)	I*4	Volume (vph) taking left turn
ITNPCT(3)	I*4	Volume (%) taking left turn
ITURN(4)	I*4	Volume (vph) taking diagonal turn
ITNPCT(4)	I*4	Volume (%) taking diagonal turn
JSORCE	I*4	Source flow rate (vph)
JSINK	I*4	Sink flow rate (vph)
JVOL	I*4	Discharge volume (vph)

This record type is repeated for each link.

Traffic Assignment Results - (Assigned destination volumes, capacity, & trips)

Record 1-6:

```
FORMAT (1H1,53X,28HASSIGNED DESTINATION VOLUMES,//,
83X,8HASSIGNED,/,
43X,11HDESTINATION,10X,8HCAPACITY,13X,5HTRIPS,/,
46X,4HNODE,16X,5H(VPH),14X,5H(VPH),/,
43X,11H-----------,10X,8H--------,11X,8H--------)
```

Record 7-(6+n):

```
FORMAT (1H0,45X,I4,14X,I6,13X,I6)
```

KNODE	I*4	Destination node number.
PATHLK(I+5)	I*4	Capacity of destination exit link (vph).
PATHLK(I+7)	I*4	Trips assigned to destination (vph).

Traffic Assignment Results - (Entry link volumes)

(Only included when entry links are specified.)

Record 1-7:

```
FORMAT (1H1,55X,18HTRAFFIC ASSIGNMENT,//,
52X,25HIDYNEV ENTRY LINK VOLUMES,///,
42X,4HLINK,14X,36HFLOW RATE      TRUCKS      CAR POOLS,/
60X,36H(VEH/HOUR)    (PERCENT)    (PERCENT),/)
```

Record 8-(7+n):

```
FORMAT (39X,1H(,I4,1H,,I4,1H),6X,I11,2I13)
```

IBUF(1)	I*4	Upstream node number of subject link.
IBUF(2)	I*4	Downstream node number of subject link.
DELFLW(JL)	I*4	Flow rate of subject link (vph).
LENGTH(JL)	I*4	Percent trucks on subject link.
J	I*4	Percent car pool vehicles on subject link.

This record type is repeated for each link.

Traffic Assignment Results - (Convergence attained)

Record 1:

```
FORMAT (21H CONVERGENCE ATTAINED)
```

Record 2-5:

```
FORMAT (1H1,44X,'     PATH LINKS VOLUMES AND CAPACITIES     ',//,
34X,'LINK',13X,'VOLUME',13X,'CAPACITY',15X,'V/C',/,
34X,'----',13X,'------',13X,'--------',15X,'---')
```

Record 6 - (5+n):

```
FORMAT (1H ,30X, '(', I4, ',', I4, ')', 7X, I7, 13X, I7, 15X,
F5.2)
```

PATHLK(K)	I*4	Upstream node number of subject link.
PATHLK(K+1)	I*4	Downstream node number of subject link.
PATHLK(K+6)	I*4	Volume of subject link (vph).
PATHLK(K+4)	I*4	Capacity of subject link (vph).
RATIO	I*4	Volume/capacity.

This record type is repeated for each internal link.

Record (6+n)-(7+n):

```
FORMAT (1H1,10X,'A TOTAL OF',I5,' PATH LINKS OUT OF',I6,
' HAVE ASSIGNED VOLUMES THAT EXCEED CAPACITY',
' BY MORE THAN 10 PERCENT.',/,11X,
'OF THESE,',I5,' EXCEED CAPACITY BY MORE THAN 50 PERCENT.')
```

ITEN	I*4	See format statement.
ICLAS1	I*4	See format statement.
IFIFTY	I*4	See format statement.

This record type is a diagnostic that is only produced when needed.

Record (8+n)-(9+n):

```
FORMAT (1H0,10X,'CONSIDER REDUCING THE TRIPS PER-HOUR ',
'GENERATED AT ORIGINS AND ATTRACTED AT ',
'DESTINATIONS (I.E. INCREASE THE',/,11X,
'ESTIMATE OF EVACUATION TIME) TO REDUCE THE ',
'NUMBER OF LINKS WITH V/C .GT. UNITY.')
```

This record type is a diagnostic that is only produced when needed.

Report type 7
Characteristics of IDYNEV Links

Record 1-3:

```
FORMAT (1H1,59X,12HIDYNEV LINKS,//)
```
or
```
FORMAT (1H1,59X,20HIDYNEV LINKS (CONT.),//)
```

Appears as header of continuation pages, if any are needed for this table.

Record 4-6:

```
FORMAT (1H0,24X,10HPKT LENGTH,7X,3HPKT,49X,
17HLOST  Q DIS  FREE,/17X,
29HLENGTH     FEET   FULL  LANES,8X,9HLANE CHAN,
6X,41HDESTINATION NODES  OPP.  TIME  HDWY.  SPD,
12H   RTOR  PED,/,
10X, 43HLINK   MI*100    L    R  LANES  L R  GRD   ,
47H1 2 3 4 5 6   LEFT THRU RGHT  DIAG NODE  SEC   ,
23H SEC   MPH   CODE  CODE)
```

This record appears in the 4th - 6th lines of each page of Characteristics of IDYNEV Links table.

Record 7-n:

```
FORMAT (7H      (,I4,1H,,I4,1H),I5,I6,3I5,I2,I4,3X,
6I2,2X,4I5,I6,F5.1,A1,F6.1,A1,I5,A1,I5,I6)
```

(XBUF(I), I = 1, 9)	I*4	Upstream, downstream nodes of link, length of link in centimiles, left and right pocket length in feet, no. of full lanes, no. of left and right pocket lanes, percent grade.
(XBUF(J), J = 11, 16)	I*4	Channelization codes for lanes 1-6.
(XBUF(K), K = 18, 22)	I*4	Downstream node number of link receiving left, through, right, and diagonal turning from subject link. Upstream node number of opposing link.
RLOST	R*4	Startup lost time in seconds.
KKSPS(1:1)	C*1	"*" if default value for RLOST.
RHDWY	R*4	Mean queue discharge headway (seconds).
KKSPS(2:2)	C*1	"*" if default value for RHDWY
XBUF(25)	I*4	Free flow speed (mph).
KKSPS(3:3)	C*1	"*" if default value for XBUF(25).
XBUF(26)	I*4	Right turn on red code.
XBUF(27)	I*4	Pedestrian volume code.

This record type is repeated for each link, 48 records per page.

Record n+1:

```
FORMAT (1H1)
```

Conditional page break if 37 or more lines have been used in current page.

Record (n+1)-(n+4):

```
FORMAT (///45X,41H* INDICATES DEFAULT VALUES WERE SPECIFIED)
```

Record (n+5)-(n+19):

```
FORMAT (//// 27X,19HLANE CHANNELIZATION,22X,4HRTOR,
25X,10HPEDESTRIAN,/,34X,5HCODES,28X,5HCODES,28X,5HCODES,//
26X,15H0  UNRESTRICTED,18X,17H0  RTOR PERMITTED,
17X,17H0  NO PEDESTRIANS,/
26X,18H1  LEFT TURNS ONLY,
15X,18H1  RTOR PROHIBITED,16X,8H1  LIGHT,/
26X,13H2  BUSES ONLY,54X,11H2  MODERATE,/
26X, 9H3  CLOSED,58X,8H3  HEAVY,/
26X,19H4  RIGHT TURNS ONLY,/
26X,14H5  CAR - POOLS, /
26X,22H6  CAR - POOLS + BUSES)
```

Report type 8
Turn Operations Data and Blockage Factors

Record 1-2:

```
FORMAT (1H1,25X,25HTURN MOVEMENT PERCENTAGES,18X,
22HTURN MOVEMENT POSSIBLE,14X,'CAPACITY REDUCTION',/
6X,4HLINK,9X,2(4HLEFT,5X,7HTHROUGH,5X,5HRIGHT,5X,
8HDIAGONAL,5X),2X,'(PERCENT)')
```

Record n+2:

```
FORMAT (3X,1H(,I4,1H,,I4,1H),5X,4(I3,8X),4(A4,7X),5X,I3)
```

IUP	I*4	Upstream node of link.
IDN	I*4	Downstream node of link.
ITRNS	I*4	Array of turn movement percentages for left, through, right, and diagonal movements.
KKTRNS	C*4	Array of turn movement flags for left, through, right, and diagonal movements. ("YES": movement possible, "NO": movement not possible).
IBLK	I*4	Blockage factor of link (%).

This record type is repeated for each link.

Report type 9
Node Coordinate Table

Record 1-6:

```
FORMAT (1H1,54X,21HNODE COORDINATE TABLE,///,
2X,5(4HNODE,6X,1HX,6X,1HY,6X),//)
```

Record 7-n:

```
FORMAT (2X,5(1H(,I4,1H),2I7,4X))
```

(IBUF(I), I = 1, IB)	I*4	1 to 5 sets of node number, x-coordinate of node in centimiles from model origin, y-coordinate of node in centimiles from model origin.

This record type is repeated as many times as needed to report coordinates of all nodes:

n = 6 + (IB/5) (+ 1 if IB/5 has remainder).

Record (n+1)-(n+4):

```
FORMAT (///,43X,37HALL COORDINATES EXPRESSED IN UNITS OF, 1 10H
MILES*100)
```

Report type 10
Traffic Control Table, Subtype 1

Record 1:

```
FORMAT (1H1,34X,36HSPECIFIED FIXED-TIME SIGNAL CONTROL,,
24H AND SIGN CONTROL,CODES)
```

The following 4 record subtypes of report type 10 (Traffic Control Table) are used conditionally. Either record subtype 2 or 3 is used, depending on whether signal intervals are specified. Record subtypes 4 and 5 will be used for each node, but record subtype 4 will not contain any data, if ILMAX = 0, and record subtype 5 will contain only the first three variables, if ILMAX = 0. In the following descriptions, n is the sum of the number of records written to one or another of the subtypes. Each node is processed in sequence, so the file produced contains all appropriate subtypes for the first node, then all for the second, etc. Record subtypes within a set for a node are ordered by subtype number.

Traffic Control Table, Subtype 2

Record (n+1)-(n+2):

```
FORMAT (1H0,60X,4HNODE,I5,/,29X,6HOFFSET,I4,4H SEC,43X,
12HCYCLE LENGTH,I4,4H SEC)
```

IGLOBE	I*4	Internal node number
NFST(KN)	I*4	Offset to network wide reference time (seconds)
ICYCLE	I*4	Length of complete signal cycle (seconds)

Traffic Control Table, Subtype 3

Record n+1:

```
FORMAT (1H0,49X,4HNODE,I5,22H IS UNDER SIGN CONTROL)
```

IGLOBE	I*4	Internal node number

Traffic Control Table, Subtype 4

Record (n+1)-(n+2):

```
FORMAT (1H0,16X,8HINTERVAL,4X,8HDURATION,4X,2H +,15(2H- ),
10HAPPROACHES,15(2H -),2H +,/17X,6HNUMBER,
18H   (SEC)  (PCT)   ,5(1H(,I4,1H,,I4,1H),4X))
```

IWORK(IA)	I*4	Upstream node number of IAth approach link
IGLOBE	I*4	Downstream node number links

These variables are repeated for IA from 1 to ILMAX (the number of non-zero signal interval values).

Traffic Control Table, Subtype 5

Record (n+1)-(n+j):

```
FORMAT (I21,I9,I7,I10,4I15)
```

INTV	I*4	Interval number
IDURT	I*4	Duration of interval (seconds)
IPERCT	I*4	Duration of interval (%)
(IWORK(IA), IA = 1,ILMAX)	I*4	Signal interval control code(s)

This record subtype is repeated for each non-zero signal interval duration (j).

Report type 11
Interpretation of Signal Codes

Record 1-22:

```
FORMAT (1H1,51X,30HINTERPRETATION OF SIGNAL CODES,/
//,43X,25H0          YIELD OR AMBER,
//,43X,16H1          GREEN,
//,43X,14H2          RED,
//,43X,37H3          RED WITH GREEN RIGHT ARROW,
//,43X,36H4          RED WITH GREEN LEFT ARROW,
//,43X,15H5          STOP,
//,43X,40H6          RED WITH GREEN DIAGONAL ARROW,
//,43X,36H7          NO TURNS-GREEN THRU ARROW,
//,43X,40H8          RED WITH LEFT AND RIGHT GREEN,6H ARROW,
//,43X,38H9          NO LEFT TURN-GREEN THRU AND,6H RIGHT)
```

Report type 12
Traffic Control Table; Signs and Fixed Time Signals

Record 1-8: Header 1

```
FORMAT ('1',37X,'TRAFFIC CONTROL TABLE' ,
' - SIGNS AND FIXED TIME SIGNALS',//,
5X,'CONTROL CODES  GO   = PROTECTED',/,
20X,'NOGO = NOT PERMITTED',/,
20X,'PERM = PERMITTED NOT PROTECTED',/,
20X,'PROT = PROTECTED',/,
20X,'STOP = STOP SIGN',/,
20X,'YLD  = YIELD SIGN')
```

Record number n is used in the following descriptions, because the actual position of a given record type in the file varies, even though their relative position is ordered by subtype number, should they occur.

Record n-(n+3): Node Identifier
if signal intervals are specified.

```
FORMAT (///,4X,4HNODE,I5,6X,18HFIXED TIME CONTROL,7X,8HOFFSET =,
I4,8H SECONDS,6X,14HCYCLE LENGTH =,I4,8H SECONDS)
```

JN	I*4	Node number
NFST(IN)	I*4	Offset to network wide reference time (seconds)
ICYCLE	I*4	Length of complete signal cycle (seconds)

or
if signal intervals are not specified.

```
FORMAT (///,4X,4HNODE,I5,6X,12HSIGN CONTROL)
```

JN	I*4	Node number

For each Node Identifier record,
Subtype 1 will be used for nodes having no approaches,
or
Report subtypes 2, 3, and 4 will be used for nodes having approaches.

Traffic control table; Signs and Fixed Time Signals; Subtype 1

Record n+4:

```
FORMAT (6X,26HNO APPROACHES TO THIS NODE)
```

Traffic control table; Signs and Fixed Time Signals; Subtype 2

Record (n+4)-(n+6):

```
FORMAT (/,15H PHASE DURATION,4X,43(1H-),
16H   APPROACHES   ,44(1H-),/,13X,5(10X,A1,I4,1H,,I4,1H)))
```

KKPARN	C*1	"("
IUP(IL)	I*4	Upstream node number of approach link
JN	I*4	Node number at signal

Each field is repeated for each signal controlled approach.

Traffic control table; Signs and Fixed Time Signals; Subtype 3

Record 12-n:

```
FORMAT (19X,5(21HLEFT THRU RITE DIAG  ))
```

Traffic control table; Signs and Fixed Time Signals; Subtype 4

Record 12-n:

```
FORMAT (I5,I9,5X,5(4(A4,1X),1X))
```

IPH	I*4	Signal phase
IDURT(IPH)	I*4	Phase duration (seconds)
((KKCODE(IM,IL),IM=1,4),IL=1,ILMAX)	I*4	One of: ' GO ', 'NOGO', 'PERM', 'PROT', 'STOP', ' YLD', ' ' where IM ranges through 4 possible turn movements and IL ranges through all approaches to node during current phase of signal.

Record subtype 4 occurs once for each signal phase at node.

Report type 13
Entry Link Volumes

Record 1-5:

```
FORMAT (1H1,55X,18HENTRY LINK VOLUMES,//,42X,4HLINK,14X,
36HFLOW RATE      TRUCKS      CAR POOLS,/,60X,
36H(VEH/HOUR)    (PERCENT)    (PERCENT)/)
```

This record type appears as the first five records of each page required to complete the entry link volumes table.

Record 6-n:

```
FORMAT (39X,1H(,I4,1H,,I4,1H),6X,I11,2I13)
```

(XBUF(I), I = I1, I2)	I*4	(1) Source node and (2) associated internal node, (3) Flow rate in vph and % of flow that are (4) trucks and (5) car pool vehicles, respectively.

This record type is repeated for each source node.

Report type 14
Source/Sink Volumes, Subtype 1

Record 1-4:

```
FORMAT (1H1,54X,22HSOURCE/SINK FLOW RATES,//,40X,
15HCENTROID NUMBER,4X,4HLINK,5X,
25HSOURCE/SINK RATE (VEH/HR)/)
```

This record subtype appears as the first four records of each page required to complete the entry link volumes table.

Record subtypes 2 and 3 appear conditionally:

- Subtype 2 will appear for each non-zero centroid number.
- Subtype 3 will appear for any source/sink with unnumbered centroid.

Source/Sink Volumes, Subtype 2

Record 5-n:

```
FORMAT (45X,I4,7X,1H(,I4,1H,,I4,1H),11X,I4)
```

(XBUF(I),I=I1,I2)	I*4	(1) Centroid number and associated link (2) upstream and (3) downstream node, (4) flow rate in vph, respectively.

This record type is repeated for each source/sink centroid.

Source/Sink Volumes, Subtype 3

Record 5-n:

```
FORMAT (56X,1H(,I4,1H,,I4,1H),11X,I4)
```

(XBUF(I),I=I3,I2)	I*4	Link (1) upstream and (2) downstream node, (3) flow rate in vph, respectively.

Report type 15
Initialization Statistics, Subtype 1

Record 1-3:

```
FORMAT (1H1,52X,25HINITIALIZATION STATISTICS,/,18X,
13HTIME INTERVAL,9X,10HSUBNETWORK,8X,
14HPRIOR CONTENT ,21H     CURRENT CONTENT ,
21H   PERCENT DIFFERENCE,/,21X,6HNUMBER,15X,
6HNUMBER,12X,10H(VEHICLES),10X,10H(VEHICLES))
```

This record subtype appears as the first three records of the Initialization Statistics table.

- Record subtype 2 appears for each time interval covered by fill time field in input record type 02.
- Record subtype 3 will occur one time, if equilibrium was attained.
  or
  Record subtype 4 will occur one time, if equilibrium was not attained.

Initialization Statistics, Subtype 2

Record 4-(n+3):

```
FORMAT (I25,17X,I5,I19,2I20)
```

ITICNT	I*4	Time interval number
INUM	I*4	Subnetwork number
TVEHS	I*4	Prior content (vehicles)
IVEHS	I*4	Current content (vehicles)
IDIFER	I*4	Percent difference

Initialization Statistics, Subtype 3

Record n+4:

```
FORMAT (1H,43X,44HALL EXISTING SUBNETWORKS REACHED EQUILIBRIUM)
```

Initialization Statistics, Subtype 4

Record n+4:

```
FORMAT (1H31X,42HINITIALIZATION TIME EXHAUSTED, SIMULATION ,
24HWILL BE PERFORMED ANYWAY)
```

Report type 16
Link and Subnetwork Statistics, Subtype 1

Report type 16 consists of 6 subtypes. Subtypes 1, 2, and 3 are provided for all pages. Subtype 4 provides information on 2 links, 10 variables for each link. The record type occurs for all internal links. Record subtype 5 is provided as a page break. A maximum of 56 records per page are written (10 of subtypes 1, 2, and 3 + up to 46 of subtype 4). Record type 5 is used if insufficient room remains on the last page of subtype 4 records to include a subtype 6 record. Record subtype 6 occurs once as the last record written to report type 16.

Record 1:

```
FORMAT (1H1,25X,32HCUMULATIVE IDYNEV SUBNETWORK NO.,I4,1X,
16HSTATISTICS SINCE,24H BEGINNING OF SIMULATION)
```

ISNUM	I*4	Subnetwork number

Link and Subnetwork Statistics, Subtype 2

Record 2:

```
FORMAT (1H0,22X,15HPRESENT TIME IS,3I3,1H,,
28H   ELAPSED SIMULATED TIME IS,I3,
7H HOURS,,I3,9H MINUTES,,I3,8H SECONDS)
```

IHRS	I*4	Current simulation time hours
IMINS	I*4	minutes
ISECS	I*4	seconds
IEHRS	I*4	Elapsed simulation time hours
IEMNS	I*4	minutes
ISECS	I*4	seconds

Link and Subnetwork Statistics, Subtype 3

Record 3-10:

```
FORMAT (1H0,58X,15HLINK STATISTICS,///,
2(39X,'AVG',5X,'AVG',16X),/,
2(18X,'VEH-    VEH-',9X,'T-TIME  SPEED  CNTNT',7X),/,
2('  NO.',3X,'LINK',6X,'MILES   DISCH  M/T   SEC/V',
'   MPH    VEH   LOS',3X)//)
```

Link and Subnetwork Statistics, Subtype 4

Record 11-n:

```
FORMAT (2(1X,I4,'(',I4,',',I4,')',F8.1,I7,F5.2,F9.1,
F6.1,F8.0,A3,4X))
```

I1	I*4	Link number

I2	I*4	Upstream node of link I1
I3	I*4	Downstream node of link I1
R1	R*4	Cumulative vehicle miles on link I1
I4	I*4	Cumulative vehicles discharged from link I1
R2	R*4	Movement/travel time
R3	R*4	Average travel time (seconds/vehicle)
R4	R*4	Average speed MPH
R5	R*4	Content (vehicles)
KKIC1	C*1	Level of service
IL	I*4	Link number
IUP	I*4	Upstream node of link IL
IDWN	I*4	Downstream node of link IL
RVMILS	R*4	Cumulative vehicle miles on link IL
ITRIPS	I*4	Cumulative vehicles discharged from link IL
RMT	R*4	Movement/travel time
RATT	R*4	Average travel time (seconds/vehicle)
RASPD	R*4	Average speed MPH
RDEN	R*4	Content (vehicles)
KKILOS(JLOS)	C*1	Level of service

Link and Subnetwork Statistics, Subtype 5

Record 1+57*page:

```
FORMAT (1H1)
```

Page break only.

Link and Subnetwork Statistics, Subtype 6

Record n-n+12:

```
FORMAT (///,51X,30HIDYNEV   SUBNETWORK STATISTICS,///,
21X,16HVEHICLE-MILES = ,F15.2,
3X,18HVEHICLE-MINUTES = ,F15.2,
5X,22HVEHICLE-TRIPS(EST.) = ,I8,//,
20X,27HPCT OF VEHS THAT STOPPED = ,F7.3,
3X,25HMOVING/TOTAL TRIP TIME = ,F5.3,
3X,18HAVG. SPEED(MPH) = ,F5.2,//,
15X,21HAVG. QUEUE CONTENT = ,F9.1,5H VEH.,
4X,16HAVG DELAY/VEH = ,F9.2,5H SEC.,
5X,14HTOTAL DELAY = ,F15.1,5H MIN.,//,
25X,17HDELAY/VEH-MILE = ,F5.2,11H MIN/V-MILE,
8X,23HTRAVEL TIME/VEH-MILE = ,F5.2,11H MIN/V-MILE)
```

RSVMLS	R*4	Total vehicle miles
RSTTT	R*4	Total vehicle travel time (minutes)
IVTRP2	I*4	Total vehicle trips (estimated)
RPSTPS	R*4	Percent of vehicles that stopped

RMSTS	R*4	Moving/total trip time
RSPID	R*4	Average speed (MPH)
RAVGQ	R*4	Average queue content (vehicles)
RDLVEH	R*4	Average delay/vehicle
RSDLY	R*4	Total delay (minutes)
RSDVML	R*4	Delay/vehicle/mile
RSTVML	R*4	Travel time/vehicle-mil (minutes/vehicle-mile)

Report type 17
Polar Sector Statistics

Report type 17 consists of 4 subtypes:

- Subtypes 1, 2, and 3 are provided for all pages.
- Subtype 4 provides information on two sectors, seven variables for each sector.

Record subtype 4 occurs for all 240 polar coordinate sectors. A maximum of 56 records per page are written (10 of subtypes 1, 2, and 3 + up to 46 of subtype 4).

Polar Sector Statistics, Subtype 1

Record 1:

```
FORMAT (1H1,30X,'CUMULATIVE IDYNEV POLAR SECTOR STATISTICS ',
'SINCE BEGINNING OF SIMULATION')
```

Polar Sector Statistics, Subtype 2

Record 2:

```
FORMAT (1H0,22X,15HPRESENT TIME IS,3I3,1H,,
28H   ELAPSED SIMULATED TIME IS,I3,
7H HOURS,,I3,9H MINUTES,,I3,8H SECONDS)
```

IHRS	I*4	Current simulation time hours
IMINS	I*4	minutes
ISECS	I*4	seconds
IEHRS	I*4	Elapsed simulation time hours
IEMNS	I*4	minutes
ISECS	I*4	seconds

Polar Sector Statistics, Subtype 3

Record 3-10:

```
FORMAT (1H0,57X,17HSECTOR STATISTICS,///,
2(37X,4HROAD,6X,3HVEH,7X,5HLEVEL,3X),/,
2(10X,4HVEH-,5X,4HVEH-,4X,5HSPEED,4X,7HCONTENT,2X,
9HYET TO BE,5X,2HOF,4X),/,
2(1X, 6HSECTOR,3X,5HMILES,3X,7HMINUTES,2X,5H(MPH),
5X,5H(VEH),3X,7HEMITTED,4X,7HSERVICE,2X), //)
```

Polar Sector Statistics, Subtype 4

Record 11-n:

```
FORMAT (2(I6,1X,F9.1,1X,F9.1,1X,F5.1,1X,F10.1,1X,F10.1,
5X,A1,5X))
```

I1	I*4	Sector number
R1	R*4	Cumulative vehicle miles on link in sector I1
R2	R*4	Vehicle minutes in sector I1
R3	R*4	Average speed MPH
R4	R*4	Content (vehicles)
R5	R*4	Vehicles yet to be emitted
KKI2	C*1	Level of service (one of A, B, C, D, E, F)
ISP	I*4	Sector number
RPVMIL(ISP)	R*4	Cumulative vehicle miles on link in sector ISP
RPVMIN(ISP)	R*4	Vehicle minutes in sector ISP
RSPD	R*4	Average speed MPH
RPCNT(ISP)	R*4	Content (vehicles)
RPSRC(ISP)	R*4	Vehicles yet to be emitted
KKILOS(JLOS)	C*1	Level of service (one of A, B, C, D, E, F)

Report type 18
Cartesian Sector Statistics, Subtype 1

Report type 18 consists of 4 subtypes:

- Subtypes 1, 2, and 3 are provided for all pages.
- Subtype 4 provides information on two sectors and seven variables for each sector.

Record type 4 occurs for all 400 Cartesian coordinate sectors. A maximum of 56 records per page are written (10 of subtypes 1, 2, and 3 + up to 46 of subtype 4).

Cartesian Sector Statistics, Subtype 1

Record 1:

```
FORMAT (1H1,28X,45HCUMULATIVE IDYNEV CARTESIAN SECTOR STATISTICS,
30H SINCE BEGINNING OF SIMULATION)
```

Cartesian Sector Statistics, Subtype 2

Record 2:

```
FORMAT (1H0,22X,15HPRESENT TIME IS,3I3,1H,,
28H   ELAPSED SIMULATED TIME IS,I3,
7H HOURS,,I3,9H MINUTES,,I3,8H SECONDS)
```

IHRS	I*4	Current simulation time hours
IMINS	I*4	minutes
ISECS	I*4	seconds
IEHRS	I*4	Elapsed simulation time hours
IEMNS	I*4	minutes
ISECS	I*4	seconds

Cartesian Sector Statistics, Subtype 3

Record 3-10:

```
FORMAT (1H0,57X,17HSECTOR STATISTICS,///,
2(37X,4HROAD,6X,3HVEH,7X,5HLEVEL,3X),/,
2(10X,4HVEH-,5X,4HVEH-,4X,5HSPEED,4X,7HCONTENT,2X,
9HYET TO BE,5X,2HOF,4X),/,
2(1X,6HSECTOR,3X,5HMILES,3X,7HMINUTES,2X,5H(MPH),
5X,5H(VEH),3X,7HEMITTED,4X,7HSERVICE,2X),//)
```

Cartesian Sector Statistics, Subtype 4

Record 11-n:

```
FORMAT (2(I6,1X,F9.1,1X,F9.1,1X,F5.1,1X,F10.1,1X,F10.1,5X,A1,5X))
```

I1	I*4	Sector number
R1	R*4	Cumulative vehicle miles on links in sector I1
R2	R*4	Vehicle minutes in sector I1
R3	R*4	Average speed MPH
R4	R*4	Content (vehicles)
R5	R*4	Vehicles yet to be emitted
KKI2	C*1	Level of service (one of A, B, C, D, E, F)
ISC	I*4	Sector number
RCVMIL(ISC)	R*4	Cumulative vehicle miles on links in sector ISCP
RCVMIN(ISC)	R*4	Vehicle minutes in sector ISC
RSPD	R*4	Average speed MPH
RCCNT(ISC)	R*4	Content (vehicles)
RCSRC(ISC)	R*4	Vehicles yet to be emitted
KKILOS(JLOS)	C*1	Level of service (one of A, B, C, D, E, F)

Report type 19
Person Measures of Effectiveness, Subtype 1

Report type 19 consists of 2 subtypes:

- Subtype 1 is provided for all pages.
- Subtype 2 provides information on three links and five variables for each sector.

Record subtype 2 occurs for all internal links. A maximum of 56 records per page are written (6 of subtype 1 + up to 50 of subtype 2).

Record 1-6: HEADER

```
FORMAT (1H1,35X,21HIDYNEV SUBNETWORK NO.,I4,1X,
1   32HPERSON MEASURES OF EFFECTIVENESS,///,
2   1X,3(14X,2(6HPERSON,3X),9HTRVL-TIME,1X),/,
3   1X,3(3X,4HLINK,8X,5HMILES,4X,5HTRIPS,3X,9HPRSON-MIN,1X),/)
```

ISNUM	I*4	Subnetwork number

Person Measures of Effectiveness, Subtype 2
Record 7-n:

```
FORMAT (1X,3(1H(,I4,1H,,I4,1H),F9.1,F9.1,F12.1,1X))
```

IBUF(II,1)	I*4	Upstream node of link
IBUF(II,2)	I*4	Downstream node of link
RBUF(II,J),J=1,3)	I*4	Person miles, person trips, and travel time in person minutes.

Three sets of link information per record (II=1,3).

Report type 20
Link Data for Time Periods Other Than 1 Table

Record 1-7:

```
FORMAT (1H1,56X, 12HIDYNEV LINKS,///62X, 4HLANE,/,
1   57X,28HCHANNELIZATION    PEDESTRIAN,/,
2   46X,36HLINK         1 2 3 4 5 6        CODE,/)
```

Record 7-n: 81 bytes

```
FORMAT (1H,42X,1H(,I4,1H,,I4,1H),4X,6I2,9X,I2)
```

XBUF(1)	I*4	Upstream node of link
XBUF(2)	I*4	Downstream node of link
XBUF(11)	I*4	Channelization code for lane 1, the right-most (curb) full lane. 0 or blank: unrestricted 1: Left turn only 2: Buses only 3: Closed 4: Right-turn only
XBUF(12)	I*4	Channelization code for lane 2, adjacent to lane 1, if any.
XBUF(13)	I*4	Channelization code for lane 3, if any.
XBUF(14)	I*4	Channelization code for lane 4, if any.
XBUF(15)	I*4	Channelization code for lane 5, if any.
XBUF(16)	I*4	Channelization code for lane 6, if any.
XBUF(26)	I*4	Pedestrian code. 0 or blank: No pedestrian traffic 1: Light (100-250 peds/h) 2: Moderate (250-500 peds/h) 3: Heavy (above 500 peds/h)

This record type is repeated for each link having a record type 11 input to the time period (2 or later).

Report type 21
Evacuation Table Reflecting Data on the Type 49 Record

Record 1-9:

```
FORMAT (1H1,////,57X,15HEVACUATION DATA,
///,28X,33HMANDATORY EVACUATION FACTOR (PCT),8X,
33HVOLUNTARY EVACUATION FACTOR (PCT),/,43X,I3,40X,I3)
```

FMAND	I*4	Mandatory evacuation factor (%)
FVOLN	I*4	Voluntary evacuation factor (%)

Record 10-22:

```
FORMAT (///,38X,32HDESIGNATED AREAS TO BE EVACUATED,
21H (E) OR SHELTERED (S),//,58X,13HRINGS (MILES),/,
50X,30H  0 -  2      2 -  5      5 - 15,/,
50X,32(1H-),//,4X,A1,10X,A1,10X,A1,///,61X,7HSECTORS )
```

(KB(I),I=1,3)	C*1	"E" / " S" if associated ring Evacuated / Sheltered

Record 23-26:

```
FORMAT (1H0,25X,16I5,/,28X,80(1H-),/,28X,16(2X,A1,2X),/)
```

(I,I=1,16)		
(KB(I+3),I=1,16)	C*1	"E" / "S" if associated ring Evacuated / Sheltered

Record 27-30:

```
FORMAT (1H0,25X,16I5,/,28X,80(1H-),/,28X,16(2X,A1,2X),/)
```

(I,I=17,32),		
(KB(I+3),I=17,32)	C*1	"E" / "S" if associated ring Evacuated / Sheltered

Record 31-34:

```
FORMAT (1H0,25X,16I5,/,28X,80(1H-),/,28X,16(2X,A1,2X),/)
```

(I,I=33,48),		
(KB(I+3),I=33,48)	C*1	"E" / "S" if associated ring Evacuated / Sheltered

Report type 22
Load Factory Table Reflecting Data on Type 52 Record

Record 1-4:

```
FORMAT (1H1,48X,27HAVERAGE VEHICLE OCCUPANCIES,/,
47X,34H(HUNDREDTHS-OF-A-PERSON / VEHICLE),//,
45X,34HAUTOS   CAR-POOLS   TRUCKS   BUSES,/,39X, 4I10)
```

VEHOCC(1)	I*4	Average occupancy of autos (hundredths-of-a-person)
VEHOCC(2)	I*4	Average occupancy of car-pools (hundredths-of-a-person)
VEHOCC(3)	I*4	Average occupancy of trucks (hundredths-of-a-person)
VEHOCC(4)	I*4	Average occupancy of buses (hundredths-of-a-person)

Report type 23
Intermediate Output Report of Measures of Effectiveness

Each page of the Report type 23 begins with a 7-line header (subtype 1, below), followed by up to 49 of subtype 2 records and a 3-line trailer record (subtype 3).

Record 1-7: HEADER, Subtype 1

```
FORMAT (1H1,35X,35HIDYNEV INTERMEDIATE LINK STATISTICS,
9H (AT TIME, 3I4, 1H),/,54X,11HTIME PERIOD,7H NUMBER,I3,///,
28X,17HAVE NO. OF       ;30HVEHICLES TURN MOVEMENT,10X,
6HNUMBER,8X,11HSIGNAL CODE,/,16X,
4HLINK,6X,32HVEHS.ON LINK DISCHARGED    ,
45HLEFT THRU RIGHT DIAGNL STOPS LEFT THRU,13H RIGHT DIAGNL,/)
```

IHRS	I*4	Simulation time - hours
IMINS	I*4	Simulation time - minute
ISECS	I*4	Simulation time - second
TPCNT	I*4	Time period

Record 8-n: Report body, Subtype 2 - 114 bytes

```
FORMAT (1H,12X,1H(,I4,1H,,I4,1H),I11,I15,6X,
2I5,2I6,I12,2X,2I5,2I6)
```

IUP	I*4	Upstream node of subject link
IDWN	I*4	Downstream node of subject link
IDEN	I*4	Average no. of vehicles on subject link
ICUMVH	I*4	Vehicles discharged from subject link
ILEFT	I*4	Number of vehicles making left turn from subject link
ITHRU	I*4	Number of vehicles making thru move from subject link
IRITE	I*4	Number of vehicles making right turn from subject link
IDIAG	I*4	Number of vehicles making diagonal turn from subject link
ISTOP	I*4	Number of vehicles stopping on subject link
ISIGL	I*4	Signal code for left turn on subject link
ISIGT	I*4	Signal code for thru movement on subject link
ISIGR	I*4	Signal code for right turn on subject link
ISIGD	I*4	Signal code for diagonal turn on subject link

Signal codes are:
0:Go
1:No-Go
2:Conditional-Go
3:Stop
4:Yield

Record n-(n+2): Trailer, Subtype 3
FORMAT (//1H,35X,30HSIGNAL CODES - 0=GO, 1=NO-GO,
28H 2=COND. GO, 3=STOP, 4=YIELD)

Report type 24
Link Capacity Factors for Time Interval (Debugging Report)

This report is not produced in IBS.

Record 1-3:

```
FORMAT (1H0,35X,34HCAPACITY FACTORS FOR TIME INTERVAL,I4,
//,1X,10(4HLINK,1X,4HFACT,3X))
```

ITT	I*4	Time interval

Record 4-n:

```
FORMAT ((1X,10(1H(,I3,2H),I3,3X)))
```

(JDUM(K),IDUM(K),K=1,ID)	I*4	Link numbers and their associated capacity factors

Report type 25
Bus Statistics (Cumulative and Final)

This report is not produced in IBS.

The cumulative report uses subtypes 1 and 2 while the final report uses subtypes 3-7: Subtype 1 produces a 6-line header at the top of each page of Cumulative Bus Statistics, followed by up to 51 detail records (subtype 2).

Subtypes 3, 4, and 5 (a total of 9 records) appear at the top of each page of Final Bus Statistics, followed by up to 46 detail records (Subtype 6). The final page of this report will contain a 3 record trailer (Subtype 7).

Bus Statistics (Cumulative), Subtype 1

Record 1-6: Cumulative Report Header

```
FORMAT (1H1,55X,21HIDYNEV BUS STATISTICS,///,
21X,33HLINK BUS TRIPS PERSON ,32HTRIPS MOVING TIME DELAY ,
26HTIME M/T NO. STOPS,/,   68X,19HMIN MIN ,/)
```

This 6-line header will appear at the top of each page of the cumulative Bus Statistics, followed by up to 51 lines of data.

Bus Statistics (Cumulative), Subtype 2

Record 7-n: Cumulative Report Data

```
FORMAT (18X,1H(,I4,1H,,I4,1H),6X,F6.1,8X,F8.1,
9X,F6.1,10X,F6.1,6X,F5.2,6X,F5.1)
```

IUP	I*4	Upstream node number of subject link
IDWN	I*4	Downstream node number of subject link
RCUMBS	R*4	Bus trips on subject link
RPTRPS	R*4	Person trips on subject link
RBMOVT	R*4	Moving time of buses on subject link
RBSDLY	R*4	Delay time of buses on subject link
RBSMT	R*4	Moving/Delay time of buses on subject link
RBSTPS	R*4	Number of bus stops on subject link

Bus Statistics (Final), Subtype 3

Record (n+6)-(n+7): Final Report Header 1

```
FORMAT (1H1,30X,39HCUMULATIVE NETWORK-WIDE BUS STATISTIC,
30HSINCE BEGINNING OF SIMULATION.)
```

Bus Statistics (Final), Subtype 4

Record 1-2: Final Report Header 2

```
FORMAT (1H0,22X,15HPRESENT TIME IS, 3I3, 1H,,
28H  ELAPSED SIMULATED TIME IS, I3,
7H HOURS,, I3, 9H MINUTES,, I3,
8H SECONDS)
```

IHRS	I*4	Current simulation time hours
IMINS	I*4	minutes
ISECS	I*4	seconds
IEHRS	I*4	Elapsed simulated time hours
IEMNS	I*4	minutes
ISECS	I*4	seconds

Bus Statistics (Final), Subtype 5

Record 3-9: Final Report Header 3

```
FORMAT (1H0,57X,16HROUTE STATISTICS,///,
7X,5HROUTE,15X,9HBUS TRIPS,15X,5HTOTAL,
17X,4HMEAN,14X,12HPERSON TRIPS,14X,6HPERSON,/,
48X,11HTRAVEL TIME,11X,11HTRAVEL TIME,
33X,11HTRAVEL TIME,/,
48X,10H(BUS-MIN.),13X,9H(SEC/BUS),35X,9H(MINUTES),/)
```

Bus Statistics (Final), Subtype 6

Record 10-n: Final Report Data

```
FORMAT (1H,I9,I25,2F21.1,I22,F24.1)
```

IBR	I*4	Bus route
BSTRPS(IBR)	I*4	Bus trips on route IBR
RXBSTR	I*4	Total travel time on route IBR (bus-minutes)
RMTVLT	I*4	Mean travel time on route IBR (bus-seconds)
IPSTRP	I*4	Person trips on route IBR
RPSTRT	I*4	Person travel time on route IBR (minutes)

Bus Statistics (Final), Subtype 7

Record n-(n+2): Final Report Trailer

```
FORMAT (//,15X,37HTHESE ESTIMATES ASSUME AN AVERAGE BUS,
1X,34HOCCUPANCY OF 25 PASSENGERS PER BUS,
1X,23HTHROUGH OUT THE NETWORK)
```

Report type 26
Run Terminated Message

Record 1.6:

```
FORMAT (46H0*** CUMULATIVE VEHICLE TRIPS HAVE NOT CHANGED,
49H FOR MORE THAN ONE TIME INTERVAL - RUN TERMINATED)
```

Filename: **CnnnCOD.U37**

Directory: SITEODYNEV:

Purpose: This file provides turn movement percentages to the simulation model as calculated by the traffic assignment model.

Access:
1. IBS -> EMERGENCY ACTIVITIES -> EVACUATION
2. IDYNEV_(NO)GR
3. EESF_(NO)GR

Notes: Most common block items within IDYNEV are maintained in their own separate common blocks rather than in an INCLUDE file. IDYNEV also uses IMPLICIT statements rather than IMPLICIT NONE in defining variables.

Structure: ASCII, sequential, fixed length 80-byte/ record -- FORTRAN carriage control. 1 or 2 turn movement sets can be entered in a single record. The file contains as many records as are necessary to describe turn movements for all links for time period number 1.

Record 1-n: 80 bytes, Type 21
`FORMAT (6I4, 8X, I4, 4X, 6I4, 8X, I4, 4H  21)`

IH1	I*4	Upstream node number
IH2	I*4	Downstream node number
IH3	I*4	Left turn volume
IH4	I*4	Right turn volume
IH5	I*4	Through movement volume
IH6	I*4	Diagonal turn volume
IH7	I*4	Flag indicating if data set described
IBUF(1)	I*4	Upstream node number
IBUF(2)	I*4	Downstream node number
ITNPCT(3)	I*4	Left turn volume
ITNPCT(2)	I*4	Right turn volume
ITNPCT(1)	I*4	Through movement volume
ITNPCT(4)	I*4	Diagonal turn volume
ITYPE	I*4	Flag indicating if data set described

Filename: **CnnnCONTOUR.DMS**

Directory: SITEOMESORAD:

Purpose: This file contains a graphic representation of dose/concentration isopleths, created using the graphic display module from MESORAD output.

Access: 1. MES_(NO)GR

Structure: DMS file.

Filename: **CnnnCONTOUR.DMS**

Directory: SITEOOSPM:

Purpose: This file contains sound pressure contours generated by OSPM for case nnn.

Access: 1. OSPM_(NO)GR

Structure: DMS format

Filename: **CnnnD2.TXT**

Directory: MAP$DATA$D2:

Purpose: This file contains the text that is associated with the CnnnPLUME.DMS file. Both files are generated by the D2 model. LIMITD2.BIN points to the current case.

See Also: MAP$DATA$D2:CnnnPLUME.DMS, MAP$DATA$D2:LIMITD2.BIN

Access: This file is created at the same time as its corresponding CnnnPLUME.DMS file.

Structure: TXT Format

Filename: **CnnnD2INP.DAT**

Directory: SITEID2:, IEMIS$SYSF:

Purpose: This file is the input file to the D2 model. It contains all the D2 model input parameters and their reference names. It also contains two lines of notes and settings for which type of units are preferred by the user. The version in IEMIS$SYSF:, C000D2INP.DAT, is a template.

See Also: D2INPnnn.DAT

Access: 1. IBS -> EMERGENCY ACTIVITIES -> HAZARD ANALYSIS -> EXECUTE D2.

Include: IEMIS$INC:D2_CONTROL.INC

Structure: ASCII, sequential, variable format, <= 80 bytes/record. Comment lines begin with #, but blank lines are also allowed. Six sections are contained in the file and data records within each section can occur in any order except that array sizes must appear before the array elements (NDI must appear before DI(n)). Not all possible keys must be used — just those the user wants. For example, the offpost file may not contain any of the tower selection section.

The data sections are as follows:
- Onpost header record (required)
- Miscellaneous data
- Tower selection data
- D2 input data
- Met change data
- End-of-release record (This record is required if information for another release is to follow. If no release follows, this record should not be included.)

The D2 input and met change sections can be repeated for multiple releases after the end-of-release record. However, only those variables that change between releases will be included in the succeeding section(s).

The format of all data records except the onpost header record is: `FORMAT (A8,1X,A1,1X,*)`. The contents are: an 8-character key, a source flag, and then the actual data, which is free-format and left-justified.

In all but the onpost header section, the variable names are actually the keys used to identify the record, not the actual names of the FORTRAN variables. In the D2 input and met change sections, the keys are generally the 3-character D2 mnemonics.

All records will have a value for the source flag, to tell where the data originated:
- N - Null, information not supplied
- D - D2PC default

C - CSEPP default (that is, VDP)
M - Met tower
U - User-supplied information

Header Record:

FORMAT (I4,1X,A12,A60,3X)

WS_D2_MAGIC	I*4	Magic # (9362)
WS_D2_ID	C*12	File ID
WS_D2_DESC	C*60	File description

Miscellaneous Section:

EVENT_DT	C*28	Date/time of event (Ex: Thu Apr 15 09:30:08 MDT 1993)
EVENT_LA	R*8	Source latitude (decimal degrees)
EVENT_LO	R*8	Source longitude (decimal degrees)
IGLOO	C*12	Igloo ID (blank if none)
REM1	C*64	Free-format remark 1
REM2	C*64	Free-format remark 2
RUN_CODE	C*4	Run type (ignored offpost)
RUN_DT	C*28	Date/time model executed (Ex: Mon Jan 24 09:30:08 MDT 1994)

Tower Selection Section:

For the following Tower (TWR) and Cluster Records (CLS):

A source flag of M means the data is available from a met tower.
A source flag of N means the data is not available from a met tower.
A tower ID of 00099999 denotes user-supplied data.

TWR_STAB	C*8	Met tower ID - Stability
CLS_STAB	C*8	Met cluster ID - Stability
TWR_WSPD	C*8	Met tower ID - Wind speed
CLS_WSPD	C*8	Met cluster ID - Wind speed
TWR_WDIR	C*8	Met Tower ID - Wind Direction
CLS_WDIR	C*8	Met Cluster ID - Wind Direction
TWR_TEMP	C*8	Met Tower ID - Temperature
CLS_TEMP	C*8	Met Cluster ID - Temperature
TWR_ATMP	C*8	Met Tower ID - Atmospheric Pressure
CLS_ATMP	C*8	Met Cluster ID - Atmospheric Pressure
TWR_CLDH	C*8	Met Tower ID - Cloud Height
CLS_CLDH	C*8	Met Cluster ID - Cloud Height
TWR_HMLR	C*8	Met Tower ID - Height of Mixing Layer
CLS_HMLR	C*8	Met Cluster ID - Height of Mixing Layer

D2 Input Section:

NOV	I*4	Novice Level
LOC	C*3	Location
SEA	C*3	Season
MUN	C*3	Munition
AGN	C*2	Agent
REL	C*3	Method of Release
TMP	R*4	Temperature
VDP	I*4	Vapor Depletion (0=Off | 1=On) (D2PC default=0)
OPO	I*4	Output Control (CSEPP default=3)
NMU	R*4	Number of Munitions (D2PC default=1)
PMM	R*4	Atmospheric Pressure (mm hg) (D2PC default=tbl)
BRT	R*4	Breathing Rate (liters/min) (D2PC default=25)
2MC	I*4	Two-minute Corrections Control (D2PC default=2)
NDI	I*4	Number of Dosages of Interest (max 10) (D2PC default=0)

For n = 1 to NDI, repeat DI and DI_D pair

DI(n)	R*4	Dosage of Interest n
DI_D(n)	C*64	Description of Dosage n
IMA	I*4	Method of Assessment Control when requesting concentration plot. Note: If IMA > 0, then NCI (Number of Concentrations of Interest) must also be greater than 0. (D2PC default=0) (CSEPP default=1)
NCI	I*4	Number of Concentrations of Interest (see IMA)

For n = 1 to NCI, repeat CI and CI_D pair

CI(n)	R*4	Concentration of Interest n
CI_D(n)	C*64	Description of Concentration n
NQI	I*4	Number of Time Intervals (max 6) (D2PC default=0)

For n = 1 to NQI, repeat Q and QT pair

Q(n)	R*4	Quantity for Release n (min)
QT(n)	R*4	Cumulative Time of Release from 1 (min)
TEV	R*4	Time of Evaporation (min)

SUR	C*3	Surface Type
ARE	R*4	Area of Puddle
LEN	R*4	Length of Puddle, Downwind
OPC	I*4	Output Control Code
HST	R*4	Height of the Stack (m)
DST	R*4	Diameter of Stack (m)
TST	R*4	Temperature of Stack (deg C)
VST	R*4	Velocity of Effluent (m/sec)
RDE	R*4	Relative Density of Effluent
HRL	R*4	Heat Released (cal)
CRD	R*4	Cloud Radius (m)
IYR	I*4	Year (yyyy)
MON	C*3	Month (and SUN) for Pasquill Stability
IDD	I*4	Number of the Day
HRS	I*4	Hours - Local Std Military Time (24hr)
CCT	I*4	Cloud Cover 0-10 (1=10%)
CHT	R*4	Cloud Height (ft)
ALF	R*4	Slope of Sigma-Y Versus X Curve
SYR	R*4	Reference Sigma Y at 100 m
BTA	R*4	Slope of Sigma-Z Versus X Curve
SZR	R*4	Reference Sigma Z at 100 m
WOO	C*2	Woods Type
FMW	R*4	Molecular Weight (cm**3/gm mole)
FMV	R*4	Molecular Volume (gm/mole)
DN25	R*4	Agent Density (vol to weight conv)
VAP	R*4	Vapor Pressure (mm hg)
BPT	R*4	Boiling Point I*4 (deg K)
ANA	R*4	Antoine Constant A
ANB	R*4	Antoine Constant B
ANC	R*4	Antoine Constant C
FRZ	R*4	Freezing Point I*4 (deg C)
TIM	R*4	Time After Functioning (min)
QQQ	R*4	Airborne Source
SLA	R*4	Station Latitude (decimal deg) default for site and used or Pasquill stability calculations (D2PC default=tbl)
SLO	R*4	Station Longitude (decimal deg) default for site and used for Pasquill stability calculations (D2PC default=tbl)
SUN	R*4	Sun Elevation Angle
FRO	R*4	Slope of the Frost Wind Profile
ZZO	R*4	Roughness Length
DLX	R*4	Change in X (1st cycle) (m)
HTS	R*4	Height of Source (m)
MNR	I*4	Minimum Response Level (D2PC default=0)
REF	R*4	Reflection Coefficient

SEV	R*4	Settling Velocity Cloud Centroid (m/sec)
SKF	R*4	Skin Factor for Subject Clothing
SMH	R*4	Sampling Height
SXS	R*4	Source Sigma X
SYS	R*4	Source Sigma Y
SZS	R*4	Source Sigma Z

Met Change Section:

STB	C*1	Stability Class
WND	R*4	Transport Wind Speed (m/sec)
W_D	R*4	Wind direction (degrees East of North)
HML	R*4	Height of the Mixing Layer (m) (D2PC default=tbl)
TMC	R*4	Time to Met Change (min). If MCOUNT = 0, then TMC = (1.E36 on EMIS, 1.E6 on IBS), else TMC(MCOUNT) = (1.E36 on EMIS, 1.E6 on IBS).
MCOUNT	I*4	Number of Met Change Records (maximum 100)

For n =1 to MCOUNT, repeat the STB, WND, W_D, HML, and TMC records

STB(n)	C*1	Met Change Stability Class (A-F)
WND(n)	R*4	Met Change Wind Speed (m/sec)
W_D (n)	R*4	Wind direction (degrees East of North)
HML(n)	R*4	Met Change Height of Mixing Layer (m)
TMC(n)	R*4	Time to Next Met Change (min). The last met record must set the Time to Met Change a number that D2PC will interpret as infinity to ensure proper completion of plume calculations. (1.E36 on EMIS, 1.E6 on IBS).

End-of-Release Record:

EOR	none	Marks the end of a release if another release is to follow. If no further releases are specified, this record will not exist.

Filename: **CnnnD2LOG.DAT**

Directory: SITEOD2:

Purpose: This file is a summary created by the D2 model. It is not used within IBS, although it may be printed or displayed.

See Also: D2LOGnnn.DAT

Access:
1. D2_(NO)GR
2. IBS -> ANALYZE TRACK -> PRINT D2 REPORT
3. IBS -> EMERGENCY ACTIVITIES -> HAZARD ANALYSIS -> PRINT D2 REPORT

Structure: ASCII, sequential, variable record length.

Record K1 (K1=1,N;1<=N<=WS_D2LOG_MAX):

```
FORMAT (A80)
```

WS_D2LOG_DESC(J)	C*80	Variable contents

Filename: **CnnnDBA.DAT**

Directory: SITEOOSPM:

Purpose: This file contains sound pressure values (DBA) generated by OSPM for case nnn for each cell in the OSPM model grid.

Access: 1. OSPM_(NO)GR

Include: OSPM$INC:OSPM.INC

Note: A model grid is defined in case input as IROWS x JCOLS, which covers a radius of interest (specified in meters) about a center of interest (specified as a lon,lat). For example, if the site specifications form contained a value of 20 for the variable grid points in each direction, a 20 x 20 model grid would be produced. Thus, 400 cells would be written, one cell per record, creating 400 records.

MODEL GRID

	(1,1)	(1,2)	(1,3)	. . .		<- (1,JCOL)
	(2,1)	(2,2)	(2,3)	. . .		
	(3,1)	(3,2)	(3,3)	. . .		
	. . .	. . .	. . .		. . .	
(IROW,1) ->						<- (IROW,JCOL)

Structure: Binary. One record for each cell in model grid. The order is column major; Rows 1 - IROW for column 1: rows 1 - IROW for column JCOL:

in{(1,1),(2,1)...(IROW,1),(1,2),(2,2)...(IROW,JCOL)}

Record 1,(JCOL*IROW): 4 bytes

DBA	R*4	DBA value for grid cell

Filename: **CnnnDBC.DAT**

Directory: SITEOOSPM:

Purpose: This file contains sound pressure values (DBC) generated by OSPM for case nnn for each cell in the OSPM model grid.

See Also: CnnnDBA.DAT

Access: 1. OSPM_(NO)GR

Include: OSPM$INC:OSPM.INC

Structure: Binary. See CnnnDBA.DAT for further explanation.

Record 1,(JCOL*IROW): 4 bytes

DBC	R*4	DBC value for grid cell

Filename: **CnnnDEPOST.ADV**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of surface concentrations resulting from depletion process - deposition and washout for each advection (time) step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Curr Relative Deposition
2. EESF_GR->Display Map-Related Graphics->Curr Relative Deposition

Include: MESO$INC:MATRIX.INC
MESO$INC:CIGMAT.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data end.s
II8	I*2	Last record of sector data
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((EXPARR(K,32-J,N),K=1,31),J=1,31)
R*4 Concentration levels in Cartesian grid (Curies per square meter) (N=5).

((PGEARR(I,J,N),I=1,36),J=1,3)
R*4 Concentration levels in polar grid (Curie per square meter) (N=5).

Filename: **CnnnDEPOST.CUM**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of surface concentrations resulting from depletion process - deposition and washout for each advection (time) step - cumulative Start to current time step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Curr Relative Deposition
2. EESF_GR->Display Map-Related Graphics->Curr Relative Deposition

Include: MESO$INC:MATRIX.INC
MESO$INC:CIGMAT.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((EXPARR(K,32-J,N),K=1,31),J=1,31)	R*4	Concentration levels in Cartesian grid (Curies per square meter) (N=5).
((PGEARR(I,J,N),I=1,36),J=1,3)	R*4	Concentration levels in polar grid (Curie per square meter) (N=5).

Filename: **CnnnDOSAGE.DAT**

Directory: SITEOD2:

Purpose: This file contains the data points used to create the D2 track.

See Also: DOSAGnnn.DAT

Access: This file is created by the D2 program and used within IBS to create the outer extent of the D2 track. The plume routine creates the map based on this file and the CnnnSLICE.DAT file.

Notes: This file supports up to 10 dosage or concentration levels. The summary records found at the beginning of the file are used to create D2 track attributes. All longitudes and latitudes are given in decimal degrees.

Structure: ASCII, sequential, variable format, <= 80 bytes/record. There are four control records, up to WS_D2_DC_MAX (10) level description records, followed by up to WS_DOS_MAX (50) records of dosage data, one separator record of zeros, and one record with the final distances for each track.

Record 1:

```
FORMAT ('N:', 1X, I4, 2X, 'WIND DIR:', 1X, I4, 2X, 'LON:', 1X,
F12.7, 2X, 'LAT:', 1X, F12.7, 19X)
```

WS_DOS_N	I*4	# dosage or concentration levels
WS_DOS_WDIR	I*4	Wind direction (degrees East of North)
WS_DOS_LON	R*8	Longitude
WS_DOS_LAT	R*8	Latitude

Record 2:

```
FORMAT (F9.2, 'MUN:', A3, 2X, 'AGN:', A2, 2X, 'REL:', A3, 2X,
'WND:', F4.1, '(M/S)', 2X, 'TMP=', F5.1, '(C)', 2X, A3, '-', A3,
1X, 'STB:', A1, 3X)
```

WS_DOS_NMU	R*4	Number of munitions
WS_DOS_MUN	C*3	Munition
WS_DOS_AGN	C*2	Agent
WS_DOS_REL	C*3	Release type
WS_DOS_WND	R*4	Wind speed (m/s)
WS_DOS_TMP	R*4	Temperature (C)
WS_DOS_LOC	C*3	Location
WS_DOS_SEA	C*3	Season
WS_DOS_STB	C*1	Stability

Record 3:

```
FORMAT ('Q(MG):', E9.3, 1X, 'TS(MIN):', E8.2, 1X, 'HTS(M):', E8.2,
1X, 'HML(M):', E8.2, 1X, 'IMA:', I1, 10X)
```

WS_DOS_QUAN	R*8	Total quantity (mg)
WS_DOS_RTIM	R*4	Time of release (min)
WS_DOS_HTS	R*4	Height of source (m)
WS_DOS_HML	R*4	Height of mixing layer (m)
WS_DOS_IMA	I*4	Method of assessment 0=dosage 1=concentration (mg/m^3) 2=concentration (ppm) 3=fumigation concentration.

Record 4 (J=1,WS_DOS_N;):

```
FORMAT (1P10E8.2)
```

WS_DOS_DOS(J)	R*4	Dosages or concentrations of interest

Record K1 (K1=J+4;J=1,WS_DOS_N):

```
FORMAT (A80)
```

WS_DOS_DESC(J)	C*80	Dosage or concentration level description. Note: If there are three levels and the description levels are blank, it will be assumed to be 1% lethality, no deaths, and no effects.

Record K2 (K2=J+4+WS_DOS_N;J=1,N;I=1,WS_DOS_N;1<=N<=WS_DOS_MAX):

```
FORMAT (F10.0, 10F7.0)
```

WS_DOS_DIST(J)	R*8	Distance (m)
WS_DOS_HALF(I,J)	R*8	Dosage/Concentration half-width (m)

Record K3 (K3=5+WS_DOS_N+N;): Separator record -- read dosages until distance is less than previous distance (this record) and then read the maximum distances (next record).

```
FORMAT (1X, 79 ('0'))
```

Record K4 (K4=6+WS_DOS_N+N; J=1, WS_DOS_N;):

```
FORMAT (10F8.0)
```

WS_DOS_MAXD(J)	R*8	Max distance (m) for Jth level

Filename: **CnnnEVAC.DMS**

Directory: SITEIDYNEV:

Purpose: This file contains a graphical representation of evacuation boundary or shelter point data created and maintained using the First-Cut approximation software (BUILDNET) available in REA_(NO)GR. This file is used by the BUILDNET software to create a evacuation network.

Access:
1. IBSSH -> MODELS -> GENERATE EVACUATION NETWORK
2. REA_(NO)GR -> GENERATE A FIRST-CUT EVACUATION CASE
3. BUILDNET

Structure: DMS Format

<u>Filename</u>: **CnnnEVACPARAM.DAT**

<u>Directory</u>: SITEIDYNEV:, IEMIS$SYSF:

<u>Purpose</u>: This file contains parameters for roadway selection and definition in first-cut approximation. The version in IEMIS$SYSF:, C000EVACPARM.DAT, is a template.

<u>Access</u>:
1. IBSSH -> MODELS -> GENERATE EVACUATION NETWORK
2. REA_(NO)GR -> GENERATE A FIRST-CUT EVACUATION CASE
3. BUILDNET

<u>Include</u>: EESF$INC:PARAMETER.INC
EESF$INC:TRIP_GENERATION.INC

<u>Structure</u>: ASCII, sequential, 16 bytes/record. This file contains as many as 100 records identifying road parameters.

<u>Record n</u>: 16 bytes
`FORMAT (I4,1X,A1,1X,I3,1X,I2,2X,I1)`

MINOR_ATT(I)	I*2	Minor attribute of graphic object (major is assumed to be 5)
CLASS_INCLUSION(I)	C*1	"Y" or "N" if road with this minor attribute should or should not be included in evacuation network
FREE_FLOW_SPEED(I)	I*2	Default free flow speed of traffic (mph) of links having this minor attribute
LANE_COUNT(I)	I*2	Default number of lanes of links having this minor attribute
ONE_WAY(I)	I*2	Default number of ways (1 or 2) for links having this minor attribute

Filename: **CnnnEXP.DMS**

Directory: MAP$DATA$EX:

Purpose: This file is the graphical representation of the Explosion contours calculated by the CHEMS model for the nnn case scenario. LIMITEX.BIN points to the current case.

See Also: MAP$DATA$EX:LIMITEX.BIN

Access: 1. CHEMS_(NO)GR

Structure: DMS Format

Filename: **CnnnEXTGRD.ADV**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of external dose to whole body from exposure to radionuclides deposited on the ground for each advection (time) step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Curr Dose - Deposition
2. EESF_GR->Display Map-Related Graphics->Curr Dose - Deposition

Include: MESO$INC:DOSE.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((DOSARR(K,32-J,N-6),K=1,31),J=1,31)
R*4 Dose levels in Cartesian grid (REM per hour) (N=15).

((PGDARR(I,J,N-6),I=1,36),J=1,3)
R*4 Dose levels in polar grid (REM per hour) (N=15).

Filename: **CnnnEXTGRD.CUM**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of external dose to whole body from exposure to radionuclides deposited on the ground for each advection (time) step - cumulative start to current time step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Cumul Dose - Deposition
2. EESF_GR->Display Map-Related Graphics->Cumul Dose - Deposition

Include: MESO$INC:DOSE.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((DOSARR(K,32-J,N-6),K=1,31),J=1,31)
R*4 Dose levels in Cartesian grid (REM) (N=16).

((PGDARR(I,J,N-6),I=1,36),J=1,3)
R*4 Dose levels in polar grid (REM) (N=16).

Filename: **CnnnEXTSIC.ADV**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of external dose to whole body from exposure to radionuclides in semi-infinite cloud for each advection (time) step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Curr Dose - Overhead Plume
2. EESF_GR->Display Map-Related Graphics->Curr Dose - Overhead Plume

Include: MESO$INC:DOSE.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((DOSARR(K,32-J,N-6),K=1,31),J=1,31)
R*4 Dose levels in Cartesian grid (REM) (N=13).
((PGDARR(I,J,N-6),I=1,36),J=1,3)
R*4 Dose levels in polar grid (REM) (N=13).

Filename: **CnnnEXTSIC.CUM**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of external dose to whole body from exposure to radionuclides in semi-infinite cloud for each advection (time) step - cumulative start to current time step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Cumul Dose - Overhead Plume
2. EESF_GR->Display Map-Related Graphics->Cumul Dose - Overhead Plume

Include: MESO$INC:DOSE.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((DOSARR(K,32-J,N-6),K=1,31),J=1,31)
R*4 Dose levels in Cartesian grid (REM) (N=14).

((PGDARR(I,J,N-6),I=1,36),J=1,3)
R*4 Dose levels in polar grid (REM) (N=14).

Filename: **CnnnFI.DMS**

Directory: MAP$DATA$FI:

Purpose: This file is the graphical representation of the Fire contours calculated by the CHEMS model for the nnn case scenario. LIMITFI.BIN points to the current case.

See Also: MAP$DATA$FI:LIMITFI.BIN

Access: 1. CHEMS_(NO)GR

Structure: DMS Format

Filename: **CnnnFLD.RPT**

Directory: SITEOMESORAD:

Purpose: Provides a report of MESORAD output covering Iodine air concentrations, Gamma and Gamma + Beta simulated readings for use by field teams during REPSS exercises.

Access:
1. MES_(NO)GR -> PRINT/DISPLAY REPORTS -> Field Team Summary Release Report
2. EESF_(NO)GR -> Print/Display Reports -> Field Team Summary Release Report

Include: MESO$INC:RSP_COM.INC
IEMIS$INC:JOB_ENV.INC

Notes: Although considered a MESORAD output, this file is actually produced by the graphic result display module of either MESORAD or EESF, based on MESORAD output files. One must first have executed MESORAD and generated output for Iodine air concentrations, Gamma, and Gamma + Beta readings, then select some graphic output display of model results in MES_GR or EESF_GR and this report will be generated based on the MESORAD output and the display parameters chosen.

Structure: ASCII, Sequential. Each page of this report begins with header records 1-19, followed by up to 31 detail records. A detail record is produced for each isopleth, for the reading (Iodine, Gamma, or Gamma + Beta) having the largest number of isopleths. A set of these detail records are followed by a blank record. Sets of such detail records are provided for each time step in MESORAD output.

Record 1,6: Variable length, header

```
FORMAT (1H1,///24X,'FIELD TEAM REPORT'//' REPORT DATE/TIME:  ',A)
```

DATE_STRING	C*23	Date and time report created.

Record 7: 46 bytes, header

```
FORMAT (' SIMULATION START DATE/TIME:   ',
I2,'/',I2,'/',I2,I4,':',I2.2)
```

RSP_STYR	I*4	Start year of MESORAD simulation.
RSP_STMO	I*4	Start month of MESORAD simulation.
RSP_STDY	I*4	Start day of MESORAD simulation.
RSP_STHR	I*4	Start hour of MESORAD simulation.
RSP_STMN	I*4	Start minute of MESORAD simulation.

Record 8: 46 bytes, header

```
FORMAT (' SIMULATION END DATE/TIME:     ',
I2,'/',I2,'/',I2,I4,':',I2.2)
```

E_YR	I*4	End year of MESORAD simulation
E_MO	I*4	End month of MESORAD simulation
E_DY	I*4	End day of MESORAD simulation
E_HR	I*4	End hour of MESORAD simulation
E_MN	I*4	End minute of MESORAD simulation

Record 9: 53 bytes, header

```
FORMAT (' SITE NAME:  ',A)
```

SITE_NAME	C*40	Site name (full).

Record 10: 34 bytes, header

```
FORMAT (' EVAC CASE #: ',I3,'  MET CASE #: ',I3)
```

EVAC_CASE_P	I*4	Evacuation case number.
MET_CASE_P	I*4	MESORAD case number.

Record 11,12: 121 bytes, header

```
FORMAT (' MESORAD RUN TITLE:  ',A/)
```

BIGTITL	C*100	Concatenated mesorad run title.

Record 13-19: Header

```
FORMAT (24X,'Field readings and associated isopleth numbers.'/
20X,54('-')/
'   Time period',10X,'Iodine',15X,'Gamma',10X,'Gamma + Beta'/
' ----------------   --------------     --------------    ---',
'------------'/
'   #  From   To     (mc/cc)  Iso. #      (mR/hr)  Iso. #',
'    (mR/hr)  Iso. #'/
'  --- ----- -----   -------    -            -------    -',
'        ---------  -'/)
```

Record K (K = 20,N): 71 bytes, header (maximum value of N is 50)

```
FORMAT (' ',A)
or
FORMAT (' ',' ')
```

OUT_LINE	C*70	Concatenated field values equivalenced as follows: byte - field type - meaning
1: B_1	C*1	see format of header records
2: T_STEP	C*3	"
5: B_2	C*1	"
6: F_HR	C*2	"

8:	COL_1	C*1	"
9:	F_MIN	C*2	"
11:	B_3	C*1	"
12:	T_HR	C*2	"
14:	COL_2	C*1	"
15:	T_MIN	C*2	"
17:	B_4	C*3	"
20:	AMT_I	C*7	"
27:	B_5	C*3	"
30:	IS_I	C*2	"
32:	B_6	C*8	"
40:	AMT_G	C*7	"
47:	B_7	C*3	"
50:	IS_G	C*2	"
52:	B_8	C*7	"
59:	AMT_GB	C*9	"
68:	B_9	C*1	"
69:	IS_GB	C*2	"

<u>Record N+1</u>: 75 bytes, footer

```
FORMAT (' ***Note: Time period and reading type dependent',     $'
Field Team Graphic Reports')
```

<u>Record N+2</u>: 75 bytes, footer

```
FORMAT (10X,'should be used to find geographic extent of',     $'
referenced isopleths.')
```

Filename: **CnnnGBRADV.BIN**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian matrix of Gamma + Beta dose rate for each advection (time) step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Curr Gamma + Beta
2. EESF_GR->Display Map-Related Graphics->Curr Gamma + Beta

Include: MESO$INC:FIELDT.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(23)	R*4	Minimum matrix data value for file 23.
RMAX(23)	R*4	Maximum matrix data value for file 23.

Record R (R = 2, NT): 3844 bytes (one record for each time step)
UNFORMATTED

((GAMMA_BETA(K,32-J),K=1,31),J=1,31)
R*4 Readings levels in Cartesian grid (mR/hr).

Filename: **CnnnGDRADV.BIN**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian matrix of Gamma dose rate for each advection (time) step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Curr Gamma
2. EESF_GR->Display Map-Related Graphics->Curr Gamma

Include: MESO$INC:FIELDT.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(22)	R*4	Minimum matrix data value for file 22.
RMAX(22)	R*4	Maximum matrix data value for file 22.

Record R (R = 2, NT): 3844 bytes (one record for each time step)
UNFORMATTED

((GAMMA(K,32-J),K=1,31),J=1,31)
R*4 Readings in Cartesian grid (mR/hr).

Filename: **CnnnING.RPT**

Directory: SITEOMESORAD:

Purpose: Provides a report of MESORAD output covering ingestion pathway doses.

Access:
1. MES_(NO)GR->PRINT/DISPLAY REPORTS->Ingestion Report
2. EESF_(NO)GR->Print/Display Reports->Ingestion Report

Include:
MESO$INC:RSP_COM.INC
MESO$INC:ING.INC
IEMIS$INC:JOB_ENV.INC
EESF$INC:EESFSTAT.INC

Structure: ASCII, Sequential.

Records 1-3: Variable length, header
```
FORMAT (1H1,24X,'INGESTION PATHWAY REPORT'//
' REPORT DATE/TIME:   ',A)
```

DATE_STRING	C*23	Date and time report created.

Record 4: 46 bytes, header
```
FORMAT (' SIMULATION START DATE/TIME:   ',
I2,'/',I2,'/',I2,I4,':',I2.2)
```

RSP_STYR	I*4	Start year of MESORAD simulation.
RSP_STMO	I*4	Start month of MESORAD simulation.
RSP_STDY	I*4	Start day of MESORAD simulation.
RSP_STHR	I*4	Start hour of MESORAD simulation.
RSP_STMN	I*4	Start minute of MESORAD simulation.

Record 5: 46 bytes, header
```
FORMAT (' SIMULATION END DATE/TIME:
',I2,'/',I2,'/',I2,I4,':',I2.2)
```

E_YR	I*4	End year of MESORAD simulation.
E_MO	I*4	End month of MESORAD simulation.
E_DY	I*4	End day of MESORAD simulation.
E_HR	I*4	End hour of MESORAD simulation.
E_MN	I*4	End minute of MESORAD simulation.

Record 6: 54 bytes, header
```
FORMAT (' SITE NAME:   ',A)
```

SITE_NAME	C*40	Site name (full).

Record 7: 34 bytes, header

```
FORMAT (' EVAC CASE #: ',I3,'  MET CASE #: ',I3)
```

EVAC_CASE	I*2	Evacuation case number.
MET_CASE	I*2	MESORAD case number.

Record 8,9: 121 bytes, header

```
FORMAT (' MESORAD RUN TITLE:  ',A/)
```

BIGTITL	C*100	Concatenated mesorad run title.

Record 10-15: Variable length, header

```
FORMAT (19X,'MILES FROM SOURCE RELEASE POINT',/,12X,55(1H-),/,
10X,10('---------- '),' which PAG',/,
' Pathway  ',10('Comit/sect '),'  exceeded',/,
11X,10('rem',8X),'Prev. Emer.',/,
' -------- ',10('---------  '),' ----  ----')
```

Record K (K = 16,17,19,20,22,23): 131 bytes, detail (I ranges from 1 to 6)

```
FORMAT (' ',A,10(F6.1,1X,A,1X),2X,F4.1,2X,F4.1)
```

PATH(I)	C*8	Ingestion pathway label.
VALUE(L,I)	R*4	Dose commitment (REM) for each distance
SECT_TABLE(SECT_VALUE(L,I)),L=1,10)	C*3	Direction code ('N ',NNW', etc.) at which dose greatest
PREV(I)	R*4	Greatest distance (miles) at which preventative action guideline exceeded for pathway I.
EMER(I)	R*4	Greatest distance (miles) at which emergency action guideline exceeded for pathway I.

Record L (L = 18,21,24): 1 bytes

```
FORMAT (' ')
```

Filename: **CnnnINHLNG.ADV**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of 50-year dose commitment to the lung from inhaled radionuclides for each advection (time) step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Curr Dose - Lung
2. EESF_GR->Display Map-Related Graphics->Curr Dose - Lung

Include: MESO$INC:DOSE.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((DOSARR(K,32-J,N-6),K=1,31),J=1,31)
R*4 Dose levels in Cartesian grid (REM) (N=7).
((PGDARR(I,J,N-6),I=1,36),J=1,3)
R*4 Dose levels in polar grid (REM) (N=7).

Filename: **CnnnINHLNG.CUM**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of 50-year dose commitment to the lung from inhaled radionuclides for each advection (time) step - cumulative Start to current time step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Cumul Dose - Lung
2. EESF_GR->Display Map-Related Graphics->Cumul Dose - Lung

Include: MESO$INC:DOSE.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((DOSARR(K,32-J,N-6),K=1,31),J=1,31)
R*4 Dose levels in Cartesian grid (REM) (N=8).
((PGDARR(I,J,N-6),I=1,36),J=1,3)
R*4 Dose levels in polar grid (REM) (N=8).

Filename: **CnnnINHTHY.ADV**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of dose commitment to thyroid from inhaled radionuclides for each advection (time) step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Curr Dose - Thyroid
2. EESF_GR->Display Map-Related Graphics->Curr Dose - Thyroid

Include: MESO$INC:DOSE.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((DOSARR(K,32-J,N-6),K=1,31),J=1,31)
R*4 Dose levels in Cartesian grid (REM) (N=9).

((PGDARR(I,J,N-6),I=1,36),J=1,3)
R*4 Dose levels in polar grid (REM) (N=9).

Filename: **CnnnINHTHY.CUM**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of dose commitment to thyroid from inhaled radionuclides for each advection (time) step - Cumulative Start to current time step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Cumul Dose - Thyroid
2. EESF_GR->Display Map-Related Graphics->Cumul Dose - Thyroid

Include: MESO$INC:DOSE.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((DOSARR(K,32-J,N-6),K=1,31),J=1,31)
R*4 Dose levels in Cartesian grid (REM) (N=10).
((PGDARR(I,J,N-6),I=1,36),J=1,3)
R*4 Dose levels in polar grid (REM) (N=10).

Filename: **CnnnINHWBD.ADV**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of inhalation dose to the whole body for each advection (time) step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Curr Dose - Inhalation/Whole Body
2. EESF_GR->Display Map-Related Graphics->Curr Dose - Inhalation/Whole Body

Include: MESO$INC:DOSE.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((DOSARR(K,32-J,N-6),K=1,31),J=1,31)
R*4 Dose levels in Cartesian grid (REM) (N=11).

((PGDARR(I,J,N-6),I=1,36),J=1,3)
R*4 Dose levels in polar grid (REM) (N=11).

Filename: **CnnnINHWBD.CUM**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of inhalation dose to the whole body for each advection (time) step - Cumulative from start to current time step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Cumul Dose - Inhalation/Whole Body
2. EESF_GR->Display Map-Related Graphics->Cumul Dose - Inhalation/Whole Body

Include: MESO$INC:DOSE.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((DOSARR(K,32-J,N-6),K=1,31),J=1,31)
R*4 Dose levels in Cartesian grid (REM) (N=12).
((PGDARR(I,J,N-6),I=1,36),J=1,3)
R*4 Dose levels in polar grid (REM) (N=12).

Filename: **CnnnINP.DAT**

Directory: SITEIDYNEV:, IEMIS$SYSF:

Purpose: This file describes the IDYNEV input case nnn. The version in IEMIS$SYSF:, C000INP.DAT, is a template.

Access:
1. IBS -> EMERGENCY ACTIVITIES -> EVACUATION
2. IDYNEV_(NO)GR

Include:
EESF$INC:FORMDS.INC
EESF$INC:FORMS.INC
EESF$INC:LSTCRD.INC
EESF$INC:XNE.INC
EESF$INC:XLN.INC
EESF$INC:XEC.INC
IEMIS$FRM:FD_00_REQUEST.INC
IEMIS$FRM:FD_01_REQUEST.INC
IEMIS$FRM:FD_02_REQUEST.INC
IEMIS$FRM:FD_03_REQUEST.INC
IEMIS$FRM:FD_04_REQUEST.INC
IEMIS$FRM:FD_06_REQUEST.INC
IEMIS$FRM:FD_11_REQUEST.INC
IEMIS$FRM:FD_21_REQUEST.INC
IEMIS$FRM:FD_35_REQUEST.INC
IEMIS$FRM:FD_49_REQUEST.INC
IEMIS$FRM:FD_50_REQUEST.INC
IEMIS$FRM:FD_51_REQUEST.INC
IEMIS$FRM:FD_52_REQUEST.INC
IEMIS$FRM:FD_175_REQUEST.INC
IEMIS$FRM:FD_176_REQUEST.INC
IEMIS$FRM:FD_177_REQUEST.INC
IEMIS$FRM:FD_178_REQUEST.INC
IEMIS$FRM:FD_276_REQUEST.INC

Notes: IDYNEV contains an internal database structure that consists of display images, CFORM, and instructions for their conversion to numeric values, XCODES. Numeric values are obtained using subroutine GETREC to retrieve a record and NUMCNV to extract numeric values from the image. The variable names presented in the following record descriptions are taken from the .TIN file-generated .INC blocks for each of the record types (designated FD_xxx_REQUEST.INC) as the data are not generally stored as variables except in relation to their use in forms. The first six INCLUDE file references are for the PREDYN software which manages this data file. The remaining 18 INCLUDE file references are for .TIN file-generated common blocks. Record types 04 and 05 share form image 04. Record types 35 and 36 share form image 35. Record types 170 and 210 are delimiters which have no corresponding form image.

Anomalies: Form FD_04_REQUEST identifies a field N_SLICES, which at one time allowed the user to control the resolution of simulation.

Structure: ASCII, sequential. The structure is complicated by 1) grouping of different record types into time period specifications, 2) the ability to have different kinds of cases (simulation and TRAFFIC/TRAD), and optional record types.

The following table, adapted from the updated *User's Guide for the IDYNEV System Including the TRAD Model*,[a] describes the requirements for inclusion of the various record types in this file; the legend for the table follows:

TYPE	Description	Simulation	TRAFFIC/TRAD
00	Title	R	R
01	Identification	R	R
02	Run Control	R	R
03	Time Period Classification	R	R
04	Time-Step Control	R	R
05	Output Options	R	R
06	Spatial Origin	O	O
11	Link Characteristics	R,O	R
21	Urban Street Turning Movements	R,O	O
35	Sign or Pre-timed Signal Control Timing	R	R
36	Sign or Pre-timed Signal Control Codes	R	R
49	Evacuation Data	O,O	NA
50	Entry Link Volumes	O,O	NA
51	Source/Sink Volumes	O,O	NA
52	Load Factors	O,O	NA
170	Delimiter Record	R,O	R
175	Traffic Assignment Parameters	NA	O
176	Origin and Destination Data	NA	R
177	Internal Centroids	NA	O
178	Destination Capacities	NA	R
210	Time Period Delimiter	R,R	R

Legend: R - Required; O - Optional; NA - Not Applicable (omit record type). When a single entry is shown, the record can be input for the first time period only. When two entries are shown, the first applies to the first time period and the second applies to all others.

(a) Prepared by KLD Associates, Inc., Huntington Station, New York 11746

Input records for subsequent time periods serve to define changes from the previous time period. It is not necessary to redefine elements of a network in subsequent time periods if unchanged from the previous time period.

The record descriptions which follow are provided in terms of record types, not record numbers, as the complexity of record number documentation would obscure rather than clarify the record definitions.

Record type 00: 80 bytes, title card

```
FORMAT (A72,6X,'00')
```

IDENTIFICATION(I)	C*72	Identification information. A minimum of 1 and a maximum of 50 records.

Record type 01: 80 bytes, identification card

```
FORMAT (A36,T39,I2,T43,I2,T47,I2,T49,A24,I4,T79,'01')
```

USER_NAME	C*36	Name of analyst performing simulation
DATE_MONTH	I*2	Month analysis performed
DATE_DAY	I*2	Day analysis performed
DATE_YEAR	I*2	Year analysis performed
NAME_OF_AGENCY	C*24	Agency or other group performing analysis
RUN_ID_NUMBER	I*2	Largely unused in IBS due to case numbers

Record type 02: 80 bytes, run control card

```
FORMAT (T7,I2,T17,I4,T25,I4,T53,I2,I2,T68,A1,T79,'02')
```

RUN_TYPE	I*2	1: traffic assignment case; 2: simulation
FILL_TIME	I*2	Minutes to fill network to equilibrium
S_FACTOR	I*2	Percentage of optimal capacity links will carry in congested conditions
SIM_START_HR	I*2	Hour of day (simulated time) simulation is to begin
SIM_START_MIN	I*2	Minute of hour (simulated time) simulation is to begin
TRAD_SW	C*1	"1": use TRAD, blank or "0" do no use TRAD

Record type 03: 80 bytes, time periods specification card

```
FORMAT (19I4,T79,'03')
```

TP_01_SEC	I*2	Duration of simulation time period (TP) No. 1 in minutes * 10
TP_02_SEC	I*2	Duration of simulation TP No. 2 in minutes * 10
TP_03_SEC	I*2	Duration of simulation TP No. 3 in minutes * 10

TP_04_SEC	I*2	Duration of simulation TP No. 4 in minutes * 10
TP_05_SEC	I*2	Duration of simulation TP No. 5 in minutes * 10
TP_06_SEC	I*2	Duration of simulation TP No. 6 in minutes * 10
TP_07_SEC	I*2	Duration of simulation TP No. 7 in minutes * 10
TP_08_SEC	I*2	Duration of simulation TP No. 8 in minutes * 10
TP_09_SEC	I*2	Duration of simulation TP No. 9 in minutes * 10
TP_10_SEC	I*2	Duration of simulation TP No. 10 in minutes * 10
TP_11_SEC	I*2	Duration of simulation TP No. 11 in minutes * 10
TP_12_SEC	I*2	Duration of simulation TP No. 12 in minutes * 10
TP_13_SEC	I*2	Duration of simulation TP No. 13 in minutes * 10
TP_14_SEC	I*2	Duration of simulation TP No. 14 in minutes * 10
TP_15_SEC	I*2	Duration of simulation TP No. 15 in minutes * 10
TP_16_SEC	I*2	Duration of simulation TP No. 16 in minutes * 10
TP_17_SEC	I*2	Duration of simulation TP No. 17 in minutes * 10
TP_18_SEC	I*2	Duration of simulation TP No. 18 in minutes * 10
TP_19_SEC	I*2	Duration of simulation TP No. 19 in minutes * 10

Record type 04: 80 bytes, time-step control card

```
FORMAT (T17,I4,T79,'04')
```

TI_LENGTH_SEC	I*2	Time interval in seconds (range is 60-600). Default is 60. Model output may potentially be generated at the end of time intervals.

Record type 05: 80 bytes, output options card

```
FORMAT (2I4,T52,I1,T79,'05')
```

NTI_OUT_CUM	I*2	Number of time intervals between output of cumulative simulation statistics
NTI_OUT_INT	I*2	Number of time intervals between output of detailed macroscopic simulation statistics

OUTPUT_CODE	I*2	0: no sector statistics requested, 1: sector statistics are requested

Record type 06: 80 bytes, model origin card

```
FORMAT (6I4,T79,'06')
```

LON1	I*2	Whole degrees longitude of case spatial origin
LON2	I*2	Fractional degrees (*.0001) longitude of case spatial origin
LON3	I*2	Fractional degrees (*.00000001) longitude of case spatial origin
LAT1	I*2	Whole degrees latitude of case spatial origin
LAT2	I*2	Fractional degrees (*.0001) latitude of case spatial origin
LAT3	I*2	Fractional degrees (*.00000001) latitude of case spatial origin

LON1, LON2 and LON3, when concatenated, form a 12-digit longitude value with 8 implied decimal places. LAT1, LAT2 and LAT3 similarly form a 12-digit latitude. These fields are constructed this way to allow their use without a decimal place in all integer IDYNEV data management software.

Record type 11: 80 bytes, urban link characteristics

```
FORMAT (2I4,I5,T22,I1,T24,I1,T26,I1,I2,T30,6I1,T37,3I4,I5,I4,I3,
I4,T70,2I1,T79,'11')
```

UPSTREAM_NN	I*2	Upstream node number, i, of subject link (i,j)
DOWNSTREAM_NN	I*2	Downstream node number, j, of subject link (i,j)
LENGTH_OF_LINK	I*2	Length of link, miles * 100
N_FULL_LANES	I*2	Number of full lanes servicing moving traffic (parking lane not included) which extend throughout the length of the link
N_LTP_LANES	I*2	Number of lanes in left-turn pocket
N_RTP_LANES	I*2	Number of lanes in right-turn pocket
PCT_GRADE	I*2	Grade, in percent (range is -9, +9)
CHAN_LANE_1	I*2	Channelization code for lane 1, the right-most (curb) full lane 0 or blank: unrestricted 1: left turn only 2: buses only 3: closed 4: right turn only
CHAN_LANE_2	I*2	Channelization code for lane 2, adjacent to lane 1, if any
CHAN_LANE_3	I*2	Channelization code for lane 3, if any
CHAN_LANE_4	I*2	Channelization code for lane 4, if any

CHAN_LANE_5	I*2	Channelization code for lane 5, if any
CHAN_LANE_6	I*2	Channelization code for lane 6, if any
DOWNSTREAM_NN_L	I*2	Downstream node number, k, of link (j,k) which receives left-turning traffic from the subject link (i,j)
DOWNSTREAM_NN_T	I*2	Downstream node number, m, of link (j,m) which receives through traffic from the subject link (i,j)
DOWNSTREAM_NN_R	I*2	Downstream node number, n, of link (j,n) which receives right-turning traffic from the subject link (i,j)
DOWNSTREAM_NN_DR	I*2	Downstream node number, ± d, of diagonal link (j,d) which receives right-half-turning (+) or left-half-turning (-) traffic from the subject link (i,j)
UPSTREAM_NN_T	I*2	Upstream node number, u, of oncoming link (u,j) servicing through traffic which opposes traffic turning left from the subject link (i,j)
STARTUP_LOST_TIME	I*2	Start-up lost time experienced by first vehicle in queue, tenths-of-a-second (mean)
MEAN_Q_DISCH_H	I*2	Mean queue discharge headway, tenths-of-a-second
FREE_FLOW_SPEED	I*2	Desired, attainable free-flow speed if (i,j) is an internal network link in miles per hour. If (i,j) is an Entry Link, leave blank.
RTOR_CODE	I*2	Right-turn-on-red code. 0: RTOR permitted 1: RTOR prohibited
PED_CODE	I*2	Pedestrian code. This code describes the intensity of pedestrian traffic impeding turning vehicles discharging from the subject link (i,j) at the downstream node, j. 0 or blank: no pedestrian traffic 1: light (100-250 peds/h) 2: moderate (250-500 peds/h) 3: heavy (above 500 peds/h)

Record type 21: 80 bytes, urban street turning movements card

```
FORMAT (6I4,4I1,I4,T79,'21')
```

UPSTREAM_NN_21	I*2	Upstream node number, i, of subject link (i,j)
DOWNSTREAM_NN_21	I*2	Downstream node number, j, of subject link (i,j). Exit links are not specified on this record.

The following 4 fields are required when simulation is run without traffic assignment. Leave blank when traffic assignment is run. The user may specify percent turns for all 4 entries or vehicle counts for all 4 entries; do not mix percents and counts.

PCT_LEFT_NN_TP	I*2	Percentage of vehicles or total vehicles turning left at node, j, for this time period (TP)
PCT_THRU_NN_TP	I*2	Percentage of vehicles or total vehicles traveling through at node, j, for this TP
PCT_RIGHT_NN_TP	I*2	Percentage of vehicles or total vehicles turning right at node, j, for this time period (TP)
PCT_DIAG_NN_TP	I*2	Percentage of vehicles or total vehicles turning diagonally at node, j, for this time period (TP)

The following 4 fields are specified when the Traffic Assignment model is run, if any turn prohibitions are present.

DISCH_PROH_L	I*2	Code (0,1) if a discharging vehicle (is, is not) prohibited from making a left turn from link (i,j)
DISCH_PROH_T	I*2	Code (0,1) if a discharging vehicle (is, is not) prohibited from making a through movement from link (i,j)
DISCH_PROH_R	I*2	Code (0,1) if a discharging vehicle (is, is not) prohibited from making a right turn from link (i,j)
DISCH_PROH_D	I*2	Code (0,1) if a discharging vehicle (is, is not) prohibited from making a diagonal turn from link (i,j)
BLOCK_FACT	I*2	Blockage factor, percent indicating percent that capacity is reduced due to a blockage on the link

Record type 35: 80 bytes, sign or pre-timed signal control timing card

```
FORMAT (7I4,9(1X,I3),T79,'35')
```

NN_OF CONTROL	I*2	Node number identifying intersection at which control is located. This is also the downstream node of all approach links to this intersection.
REF_OFFSET	I*2	Reference offset to signal interval 1, in seconds. If control is a stop or yield sign or no control device is present, leave blank.
UPSTREAM_NN_35_1	I*2	Upstream node number of approach link number 1
UPSTREAM_NN_35_2	I*2	Upstream node number of approach link number 2 (if any)
UPSTREAM_NN_35_3	I*2	Upstream node number of approach link number 3 (if any)

UPSTREAM_NN_35_4	I*2	Upstream node number of approach link number 4 (if any)
UPSTREAM_NN_35_5	I*2	Upstream node number of approach link number 5 (if any)
DUR_SIG_INT(I)	I*2	Duration of signal intervals in seconds. A maximum of 9 intervals may be specified. If control is a stop, yield sign, or perpetual green (or uncontrolled): leave blank.

Record type 36: 80 bytes, sign or pre-timed signal control codes card

```
FORMAT (I4,1X,45I1,T66,2I5,T79,'36')
```

NN_OF CONTROL	I*2	Node number, n, identifying intersection.

9 sets of the following 5 fields. One set of 5 fields for each of 9 signal interval durations. Codes are defined as follows:

0: Yield sign or amber
1: Green
2: Red
3: Red with green right arrow
4: Red with green left arrow
5: Stop sign
6: Red with green diagonal arrow
7: No turns - green thru arrow
8: Red with left and right green arrows
9: No left turn - green thru and right

CC_1(I)	I*2	Control Code for control servicing (i.e., facing) approach link number 1 during interval I.
CC_2(I)	I*2	Control Code for control servicing (i.e., facing) approach link number 2 during interval I.
CC_3(I)	I*2	Control Code for control servicing (i.e., facing) approach link number 3 during interval I.
CC_4(I)	I*2	Control Code for control servicing (i.e., facing) approach link number 4 during interval I.
CC_5(I)	I*2	Control Code for control servicing (i.e., facing) approach link number 5 during interval I.
X_COORDINATE	I*4	X-coordinate of the node in miles * 100. Offset is from west to east using longitude origin from record type 06.
Y_COORDINATE	I*4	Y-coordinate of the node in miles * 100. Offset is from south to north using latitude origin from record type 06.

Record type 49: 80 bytes, evacuation data card

```
FORMAT (2I4,T26,51I1,T79,'49')
```

MAN_EVAC_PCT	I*2	Advisory evacuation factor in percent (1-100). The percentage of vehicles originating within sectors which, having been advised to evacuate, will actually do so.
VOL_EVAC_PCT	I*2	Voluntary evacuation factor in percent (1-100). The percentage of vehicles originating within sectors which, having been advised to shelter, will ignore advisory and actually evacuate.

Note that all of the following assume 3 concentric rings, divided into 16 sectors of 22.5 degrees each. Outer boundary is 10 miles from power plant. Sectors are counted clockwise from north with sector 1 to north in ring 1.

RING_CODE_0_2	I*2	Ring evacuation flag. Code (0,1) if the 0-2 mile ring (I=1) is to be (sheltered, evacuated).
RING_CODE_2_5	I*2	Same as last entry but for 2-5 mile ring (I=2).
RING_CODE_5_BDRY	I*2	Same as last entry but for the 5-mile to EPZ boundary ring (I=3).
RING_SEC_1(I)	I*2	Sector evacuation flag. Code (0,1) if sector 1 of ring I is to be (sheltered, evacuated).
RING_SEC_2(I)	I*2	Sector evacuation flag. Code (0,1) if sector 2 of ring I is to be (sheltered, evacuated).
RING_SEC_3(I)	I*2	Sector evacuation flag. Code (0,1) if sector 3 of ring I is to be (sheltered, evacuated).
RING_SEC_4(I)	I*2	Sector evacuation flag. Code (0,1) if sector 4 of ring I is to be (sheltered, evacuated).
RING_SEC_5(I)	I*2	Sector evacuation flag. Code (0,1) if sector 5 of ring I is to be (sheltered, evacuated).
RING_SEC_6(I)	I*2	Sector evacuation flag. Code (0,1) if sector 6 of ring I is to be (sheltered, evacuated).
RING_SEC_7(I)	I*2	Sector evacuation flag. Code (0,1) if sector 7 of ring I is to be (sheltered, evacuated).
RING_SEC_8(I)	I*2	Sector evacuation flag. Code (0,1) if sector 8 of ring I is to be (sheltered, evacuated).
RING_SEC_9(I)	I*2	Sector evacuation flag. Code (0,1) if sector 9 of ring I is to be (sheltered, evacuated).
RING_SEC_10(I)	I*2	Sector evacuation flag. Code (0,1) if sector 10 of ring I is to be (sheltered, evacuated).
RING_SEC_11(I)	I*2	Sector evacuation flag. Code (0,1) if sector 11 of ring I is to be (sheltered, evacuated).
RING_SEC_12(I)	I*2	Sector evacuation flag. Code (0,1) if sector 12 of ring I is to be (sheltered, evacuated).
RING_SEC_13(I)	I*2	Sector evacuation flag. Code (0,1) if sector 13 of ring I is to be (sheltered, evacuated).
RING_SEC_14(I)	I*2	Sector evacuation flag. Code (0,1) if sector 14 of ring I is to be (sheltered, evacuated).

RING_SEC_15(I)	I*2	Sector evacuation flag. Code (0,1) if sector 15 of ring I is to be (sheltered, evacuated).
RING_SEC_16(I)	I*2	Sector evacuation flag. Code (0,1) if sector 16 of ring I is to be (sheltered, evacuated)

Record type 50: 80 bytes, Entry Link Volume Card
`FORMAT (4I4,T21,4I4,T41,4I4,T79,'50')`

Although the form FD_50_REQUEST only allows access to one entry link at a time, this record can define volumes for up to 3 entry links. Thus the record definition consists of three sets of the following four fields:

UPSTREAM_NN_1	I*2	Upstream node, i, of link (i,j). Value in range (8000 - 8999)
DOWNSTREAM_NN_1	I*2	Downstream node, j, of link (i,j). Value in range (1 - 1999)
VEHICLES_HR_1	I*2	Flow rate expressed in vehicles/hour, entering the network from entry link, (i,j)
PCT_TRUCKS_1	I*2	Percent trucks on the entry link

Record type 51: 80 bytes, source/sink volume card
`FORMAT (16I4,T79,'51')`

Although the form FD_51_REQUEST only allows access to work with one source/sink volume set at a time, this record can define volumes for up to 4 source/sink centroids. Thus, the record definition consists of four sets of the following four fields:

SS_CENTROID	I*2	Source/sink centroid number. Range is 2000 - 2999.
UPSTREAM_51	I*2	Upstream node, i, of link (i,j) which is accessed from/to the source/sink.
DOWNSTREAM_51	I*2	Downstream node, j, of link (i,j).
NET_VOLUME_51	I*2	Net traffic volume entering/leaving link (i,j) from/to source/sink in vehicles/hour. If a discharge to a sink, this quantity is negative; for emission from a source, this quantity is positive.

Record type 52: 80 bytes, load factor card
`FORMAT (4I4,T79,'52')`

AVG_PER_AUTO	I*2	Average number of persons occupying each automobile, in hundredths. If left blank, a default value of 130 (that is, 1.3 persons/vehicle) will be assigned internally.

AVG_PER_POOL	I*2	Average number of persons occupying each car pool vehicle, in hundredths. If left blank, a default value of 350 (that is, 3.5 persons/vehicle) will be assigned internally.
AVG_PER_TRUCK	I*2	Average number of persons occupying each truck, in hundredths. If left blank, a default value of 120 (that is, 1.2 persons/vehicle) will be assigned internally.
AVG_PER_BUS	I*2	Average number of persons occupying each bus, in tenths. If left blank, a default value of 250 (that is, 25 persons/vehicle) will be assigned internally.

Record type 170: 80 bytes, subnetwork delimiter card

```
FORMAT (I4,1X,I3,T78,'170')
```

No form is provided for this record. Fields are unnamed.

unnamed	I*2	Blank if records following the Type 170 record are numbered above 170, that is, they do not define another DYNEV subnetwork. If data for another subnetwork follows, set this entry to '8'.
unnamed	I*2	DYNEV subnetwork number defined by the card types following the 170 record.

Record type 175: 80 bytes, traffic assignment parameters card

```
FORMAT (6I4,T78,'175')
```

EXP_VAL	I*2	Absolute value of the exponent, defining the acceptable threshold, Epsilon, for the relative change in the objective function from one iteration to the next. If left blank, a value of 2 is assigned. Range is 2 - 5.
MAX_INNER_ITER	I*2	Maximum number of iterations to be performed. If left blank, a value of 3 is assigned. The specified value should not exceed 20.
COEF_A	I*2	The value of the coefficient "A" in the BPR impedance equation, multiplied by 100. If the value .23 is desired, the number, 23, must be specified. If left blank, a value of .15 will be assigned internally. The value must be positive and in general not exceed 1 (i.e., this entry should usually be less than 100).
COEF_B	I*2	The value of the coefficient "B" in the BPR impedance equation, multiplied by 10. If the

		value 3.5 is desired, the number, 35, must be specified. If left blank, a value of 4.0 will be assigned internally. The value must be positive and in general not exceed 10 (i.e., this entry should usually be less than 100).
KALMAN_FILTER	I*2	The value of the Kalman filter coefficient which applies to computed values of capacity (10 - 90). If left blank, a value of 50 is assigned.
MAX_OUTER_ITER	I*2	Maximum number of outer iterations to be performed. If left blank, a value of 3 is assigned. The specified value should not exceed 20.

Record type 176: 80 bytes, origin and destination data card

```
FORMAT (I4,T13,16I4,T78,'176')
```
Supports trip-table

or

```
FORMAT (I4,T17,2I4,6(4X,I4),T78,'176')
```
Supports TRAD

Format choice is dependent on whether TRAD was selected in record type 02. The form presentation of this record uses FD_176_REQUEST, if TRAD model is not being used, and FD_276_REQUEST, if it is.

ORIGIN_NN	I*2	Origin node number of the form 2xxx or 8xxx

Trip-Table Option - TRAD model Bypassed:

DESTINATION_1	I*2	Destination node number of the form 2xxx or 8xxx. The traffic volume specified in the following entry is emitted at the origin node and travels to the destination node specified in this entry.
VOL_PRODUCED_1	I*2	Traffic volume, vehicles per hour
DESTINATION_2	I*2	Same as _1 for another destination
VOL_PRODUCED_2	I*2	Traffic volume, vehicles per hour
DESTINATION_3	I*2	Same as _1 for another destination
VOL_PRODUCED_3	I*2	Traffic volume, vehicles per hour
DESTINATION_4	I*2	Same as _1 for another destination
VOL_PRODUCED_4	I*2	Traffic volume, vehicles per hour
DESTINATION_5	I*2	Same as _1 for another destination
VOL_PRODUCED_5	I*2	Traffic volume, vehicles per hour
DESTINATION_6	I*2	Same as _1 for another destination
VOL_PRODUCED_6	I*2	Traffic volume, vehicles per hour
DESTINATION_7	I*2	Same as _1 for another destination
VOL_PRODUCED_7	I*2	Traffic volume, vehicles per hour
DESTINATION_8	I*2	Same as _1 for another destination
VOL_PRODUCED_8	I*2	Traffic volume, vehicles per hour

TRAD Model is Applied:

VOL_PRODUCED	I*2	Total traffic volume emitted form ORIGIN_NN
NODE_1	I*2	Candidate Destination node number, of the form 8xxx for traffic emitted form ORIGIN_NN
NODE_2	I*2	Another candidate destination node number
NODE_3	I*2	Another candidate destination node number
NODE_4	I*2	Another candidate destination node number
NODE_5	I*2	Another candidate destination node number
NODE_6	I*2	Another candidate destination node number
NODE_7	I*2	Another candidate destination node number

Record type 177: 80 bytes, internal centroids card

```
FORMAT (18I4,T78,'177')
```

Although the form FD_177_REQUEST only allows access to one internal centroid at a time, this record can define up to six internal centroids. Thus, the record definition consists of six sets of the following three fields:

CENTROID_NN_1	I*2	Internal centroid - either origin, destination, or both. Range (2000 - 2999).
UPSTREAM_NN_OF_L	I*2	Upstream node of internal link which provides access to/from this internal centroid.
DSTREAM_NN_OF_L	I*2	Downstream node of internal link which provides access to/from this internal centroid.

Record type 178: 80 bytes, destination capacities card

```
FORMAT (18I4,T78,'178')
```

Although the form FD_178_REQUEST only allows access to one destination node at a time, this record can define up to nine destination nodes. Thus, the record definition consists of nine sets of the following two fields:

DESTINATION_NN	I*2	Destination node number. Must be an Exit node - range is 8000 - 8999.
TOTAL_CAPACITY	I*2	The maximum number of vehicle trips in vph that can be accommodated by the destination identified in DESTINATION_NN.

Record type 210: 80 bytes, final card in time period

```
FORMAT (2I4,1X,I3,T78,'210')
```

No form is provided for this record. Fields are unnamed.

unnamed	I*2	Code (0,1) if this Time Period (is, is not) the final Time Period.

unnamed	I*2	Code (0,8) 0: No additional records. 8: Following records are type 11-52 for the next time period
unnamed	I*2	If this is not the final 210 record, this entry should contain the DYNEV subnetwork number whose data follows this record.

Filename: **CnnnIODADV.BIN**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian matrix of Radio Iodine concentrations for each advection (time) step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Curr Iodine
2. EESF_GR->Display Map-Related Graphics->Curr Iodine

Include: MESO$INC:FIELDT.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(21)	R*4	Minimum matrix data value for file 21.
RMAX(21)	R*4	Maximum matrix data value for file 21.

Record R (R = 2, NT): 3844 bytes (one record for each time step)
UNFORMATTED

((R_IODINE(K,32-J),K=1,31),J=1,31)
R*4 Instantaneous concentration in air of iodine isotopes (mCi/cm**3).

Filename: **CnnnLNK.DAT**

Directory: SITEIDYNEV:

Purpose: This file describes IDYNEV network topology. It is used to transfer this information from the network editor to the graphic process that supports it.

Access:
1. IBS -> EMERGENCY ACTIVITIES -> EVACUATION
2. IDYNEV_(NO)GR

Include:
EESF$INC:GRAPHNET.INC
EESF$INC:PREDYNGR.INC
IEMIS$INC:JOB_ENV.INC

Structure: Binary, sequential

Record 1: UNFORMATTED

SITE_NAME	C*40	Extended site name
LON_INIT_SITE	R*8	Longitude of site center
LAT_INIT_SITE	R*8	Latitude of site center
TP	I*4	Time period
DEV_TEK	L*4	If .TRUE. terminal is Tektronix compatible
SITE	C*4	Site code
LL_DYNEV	L*4	If .TRUE. the following fields are provided
LON_DYNEV	R*8	DYNEV longitude of model space origin from record type 06
LAT_DYNEV	R*8	DYNEV latitude of model space origin from record type 06

Record 2: UNFORMATTED

NODES	I*4	Maximum internal node number used
ENTRIES	I*4	Entry node/link count
NEW_EXITS	I*4	Number of exit nodes
LINKS	I*4	Number of links
CENTROIDS	I*4	Number of centroids

Record 3, LINKS*2+1, 2 : UNFORMATTED

NODE_COORD(1,LINK_NODES(1,I))	I*4	Upstream node's X coordinate for link I
NODE_COORD(2,LINK_NODES(1,I))	I*4	Upstream node's Y coordinate for link I
LINK_NODES(1,I)	I*4	Upstream node of link I

Record 4, LINKS*2+2, 2: UNFORMATTED

NODE_COORD(1,LINK_NODES(2,I))		
	I*4	Downstream node's X coordinate for link I
NODE_COORD(2,LINK_NODES(2,I))		
	I*4	Downstream node's Y coordinate for link I
LINK_NODES(2,I)	I*4	Downstream node of link I

Record LINKS*2+3: UNFORMATTED

NODE_COORD(1,1)	I*4	First node's X coordinate
NODE_COORD(2,1)	I*4	First node's Y coordinate
literal	I*4	-1 (Marks the end of the limits)

Record LINKS*2+4: Variable length, this record contains J pairs (one for each entry node): UNFORMATTED

ENTRY_NODE(J)	I*4	Entry node number
ENTRY_NODE_LOC(J)	I*4	Associated internal node number

Record LINKS*2+5: Variable length, this record contains J pairs (one for each exit node): UNFORMATTED

NEW_EXIT_NODE(J)	I*4	Exit node number
NEW_EXIT_NODE_LOC(J)		
	I*4	Associated internal node number

Record LINKS*2+6: Variable length, this record contains J pairs (one for each link): UNFORMATTED

LINK_NODES(1,J)	I*4	Upstream node number of link J
LINK_NODES(2,J)	I*4	Downstream node number of link J

Record LINKS*2+7: Variable length, this record contains J triplets (one for each centroid): UNFORMATTED

CENTROID_NO(J)	I*4	Centroid number
CENTROID_LOC_T(1,J)	I*4	Upstream node number of link to which centroid J is attached
CENTROID_LOC_T(2,J)	I*4	Downstream node number of link to which centroid J is attached

Filename: **CnnnLNK.DAT**

Directory: SITEODYNEV:

Purpose: This file contains link statistics extracted from IDYNEV output to support rapid reporting to the user of evacuation result graphical display and animation.

See Also: CnnnCOD.RPT

Access: IBS->EMERGENCY ACTIVITIES->EVACUATION

Notes: This file contains one record for each link for each output time step produced by the IDYNEV simulation. Record_index = (t-1)*l+il where t is the number of the time step, l is the number of links in the network and il is the specific link index.

Structure: Binary, direct access, fixed length, 28 bytes

Record 1,N:

BUF(1)	R*4	Cumulative vehicle miles on link IL
BUF(2)	R*4	Cumulative vehicles discharged from link IL
BUF(3)	R*4	Movement / travel time
BUF(4)	R*4	Average travel time (seconds/vehicle)
BUF(5)	R*4	Average speed MPH
BUF(6)	R*4	Content (vehicles)
LOS	I*4	Level of service (INTEGER EQUIVALENT)

Filename: **CnnnLOG.RPT**

Directory: SITEOMESORAD:

Purpose: Provides a textual report of input, diagnostic messages, and some output results of the MESORAD model.

Access:
1. MES_(NO)GR->PRINT/DISPLAY REPORTS->Met/Dose Case Debug Report
2. EESF_(NO)GR->Print/Display Reports->Met/Dose Case Debug Report

Include: MESO$INC:*.INC

Notes: The specific formats of each of the records in this file are not provided because records may be written to it from more than 400 locations within MESORAD and the file is not processed by any software in IBS other than to display or print it.

Structure: ASCII, Sequential.

Filename: **CnnnMASTER_CHECKLIST.DAT**

Directory: SITE$IP:

Purpose: This file contains all the tasks listed for IP number nnn. A total of up to 99 tasks is allowed for each of the possible emergency functions.

Access:
1. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES -> CHANGE/USE TASKS
2. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES -> REPORT

Include: IEMIS$INC:IP_MC.INC, IEMIS$INC:USE_TASK_2.INC

Structure: ASCII, Direct Access, 325 bytes/record, with record locking at the task level during operational mode. In planning mode, no record locking exists. The first record lists the number of records contained in the file. Each record contains the emergency function and task ID number.

Record 0: 325 bytes, header
`FORMAT (I6,1X,A23,1X,A5,289X)`

NUM_REC	I*4	Number of records in the file
CDTS	C*23	Current system date/time when file last modified
IP_VERSION	C*5	Current IP version

Record 1-NUM_REC: 325 bytes, task record
`FORMAT (A20,A6,A160,A5,A20,A8,A1,A1,A23,A23,A23,A23,A1,A4,A7)`

ID	C*20	Emergency function
TASK_IDENT	C*6	Unique task number ID (unique for given EF)
MC_TASK_DESC	C*160	Short task description
MC_AGENCY	C*5	Agency code
MC_POSITION	C*20	Position
MC_DURATION	C*8	Estimated task duration
MC_CRUCIAL	C*1	Crucial flag (Y/N)
MC_DONE	C*1	Done flag (Y/N)
MC_DT_AS	C*23	Actual start time
MC_DT_PS	C*23	Planned start time
MC_DT_PC	C*23	Planned completion time
MC_DT_AC	C*23	Actual completion time
MC_OPIP_FLAG	C*1	'O' if OPIP, 'I' if IP
MC_CONSTRAINT	C*4	Signifies if task is fixed in time
MC_MISC	C*7	Misc. space for future expansion

Filename: **CnnnMASTER_NOTES.DAT**

Directory: SITE$IP:

Purpose: This file contains the master note for IP number nnn.

Access:
1. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES -> CHANGE/VIEW IP NOTES
2. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES -> REPORT

Include: IEMIS$INC:IP_XN.INC

Structure: ASCII, Direct Access, 102 bytes/record. It is meant as a general note for the given IP and may only be changed in the planning mode.

Record 0: 102 bytes, header
```
FORMAT (I6,1X,A23,1X,A5)
```

NUM_REC	I*4	Number of records in the file
CDTS	C*23	Current system date/time when file last modified
IP_VERSION	C*5	Current IP version

Record 1-NUM_REC: 102 bytes, IP note record
```
FORMAT (A20, A6, A76)
```

ID	C*20	Emergency function
TASK_IDENT	C*6	Not used for master notes
XN_NOTES	C*76	Actual line of notes

Filename: **CnnnNOD.DAT**

Directory: SITEODYNEV:

Purpose: This file contains sign and signal information extracted from the IDYNEV report CnnnCOD.RPT for quicker access during response to user picks of nodes from graphic display of evacuation results.

See Also: CnnnCOD.RPT

Access: IBS -> EMERGENCY ACTIVITIES -> EVACUATION

Notes: This file contains a block of 12 records for each node in the IDYNEV case nnn. The records are copied from the CnnnCOD.RPT file without modification.

Structure: Binary, direct access, fixed length, 132 bytes

Record 1-12*n:

BUF	C*132	Text extracted from sign or signal report in CnnnCOD.RPT for node n.

Filename: **CnnnOBS.DAT**

Directory: SITEIMESORAD:, IEMIS$SYSF:

Purpose: This file contains weather observations for the MESORAD model. The version in IEMIS$SYSF:, C000OBS.DAT, is a template.

Access: 1. MES_(NO)GR->Update Case Data->Change Weather Data

Include: MESO$INC:OBS.INC

Structure: ASCII, Direct Access.

Record 1,N: 300 bytes (Maximum value of N is 300. Only 147 bytes are currently being used).
`FORMAT (1X,5I2.2,1X,I1,1X,I4,1X,I1,1X,I3,I2,1X,20(L1,I3,I2))`

DDDD(J), J=1,8	I*4	Year, month, day, hour, minute, stability class (A: very unstable - G: very stable), mixing depth (m), precipitation code (0:none 1-3:light, moderate, heavy rain 4-6:light, moderate, heavy snow, or hail if 6).
DDDD(J), J=11,12	I*4	Upper level wind direction, upper level wind speed.

20 Sets of the following (I=1,20)

STATION_STATE(I)	L*4	.TRUE. (station is on).
DDDD(I+12)	I*4	Wind direction (degrees clockwise from north) for station I.
DDDD(I+13)	I*4	Wind Speed meters/second for station I.

Filename: **CnnnOCTm.DAT** (m in range of 1 - 8)

Directory: SITEOOSPM:

Purpose: This file contains sound pressure values for octave band *m* generated by OSPM case *nnn* for each cell in the OSPM model grid.

See Also: CnnnDBA.DAT

Access: 1. OSPM_(NO)GR

Include: OSPM$INC:OSPM.INC

Notes: The frequency for octave band n is:

1. 63Hz
2. 125Hz
3. 250Hz
4. 500Hz
5. 1000Hz
6. 2000Hz
7. 4000Hz
8. 8000Hz

Structure: Binary. See CnnnDBA.DAT for further explanation.

Record 1,(JCOL*IROW): 4 bytes

RSPL(x+1)	R*4	Sound pressure level of cell for octave band n

Filename: **CnnnOSPM.DAT**

Directory: SITEIOSPM:, IEMIS$SYSF:

Purpose: This is the case nnn input file for the Outdoor Sound Propagation Model (OSPM). The version in IEMIS$SYSF:, C000OSPM.DAT, is a template.

Access: 1. OSPM_(NO)GR

Include: OSPM$INC:OSPM.INC

Structure: ASCII, sequential, variable length record

Record 1:

```
FORMAT (A80)
```

TITLE	C*80	Case description

Record 2:

```
FORMAT (*)
```

X0LONG	R*8	Longitude of site center
Y0LAT	R*8	Latitude of site center
RADIUS	R*4	Radius of area to model in miles

Record 3:

```
FORMAT (*)
```

IROW	I*4	Number of rows in which to grid model area
JCOL	I*4	Number of columns in which to grid model area

Record 4:

```
FORMAT (*)
```

NH	I*4	Number of acoustically hard grid points

RecordS 5,4+NH:

```
FORMAT (*)
```

IHARD(IH,1)	I*4	Row index of IHth hard point
IHARD(IH,2)	I*4	Col index of IHth hard point

Record NH+5:

```
FORMAT (*)
```

H1	I*4	Ground level height (in feet)

WD_OSPM	I*4	Wind direction (deg. clockwise from North)
WS1	I*4	Wind Speed (in mph)
T1	I*4	Temperature (deg. F)
RH	I*4	Relative humidity (%)
BP	I*4	Barometric pressure (in mm of Hg)

Record NH+6:

FORMAT (*)

H2	I*4	Upper level height in feet
WS2	I*4	Wind Speed in mph
T2	I*4	Temperature (deg. F)

Record NH+7:

FORMAT (*)

NS	I*4	Number of sirens

Record NH+8,NH+8*NS,2:

FORMAT (*)

SXY_LONG_LAT(IS,1)	R*8	Longitude of siren IS
SXY_LONG_LAT(IS,2)	R*8	Latitude of siren IS
HS(IS)	R*4	Height of siren IS
ITYPE(IS)	I*4	Siren type: 0 - Rotating siren, 1 - omni-directional siren

Record NH+9,NH+9*NS,2: Variable

FORMAT (A)

SIREN_NAMES(IS)	C*25	Name of siren IS (must match one of those in siren library)

Filename: **CnnnOSPM.DMS**

Directory: SITEIOSPM:

Purpose: This file contains siren icon references for OSPM case nnn.

Access: 1. OSPM_GR

Structure: DMS format

Filename: **CnnnPLOT.DAT**

Directory: SITEOD2:

Purpose: This file contains D2 track half-widths at various distances from the dispersion source.

See Also: CnnnDOSAGE.DAT

Access: IBS->EMERGENCY ACTIVITIES->HAZARD ANALYSIS->EXECUTE D2->Run Current D2 Case

Notes: This file, together with CnnnD2INP.DAT, is used to create the CnnnDOSAGE.DAT file. This file always contains dosage records for valid D2 runs; however, concentration records are optional, based on user input (IMA) into the model. Dosage records are listed before concentration records. The following series is repeated until all distances generated by the model have been written. Generally, each section will contain all the information for a given dosage or concentration level; however, time to met change releases will be broken into smaller bits depending on the duration of each met change.

Structure: ASCII, sequential.

Record 1: FORMAT (*)

D_OR_C	C*1	'D' for dosage level, 'C' for concentration
LEV_NDX	I*4	Index of level of interest

Record 2: FORMAT (*)

LEV_DESC	C*80	Description of level

Record 3: FORMAT (*)

NUM_DIST	I*4	Number of distances for given level

Record 4 - NUM_DIST:
FORMAT (*)

WS_DOS_DIST()	R*8	Distance from dispersion origin
WS_DOS_HALF()	R*8	Half-width distance of D2 track at distance WS_DOS_DIST()

Record (NUM_DIST+1) - (NUM_DIST+2):
FORMAT (*)

BLANK_LINE	C*1	Blank line

Filename: **CnnnPLUME.DMS**

Directory: MAP$DATA$D2:

Purpose: This is the graphical representation of the D2 track for case nnn.

Access: This file is created by the PLUME routine after D2 output has been generated. The plume itself is created from the CnnnDOSAGE.DAT (or DOSAGEnnn.DAT) file, and its attributes (if any) are generated from the CnnnSLICE.DAT (or SLICEnnn.DAT) file.

Structure: DMS format

Filename: **CnnnPOL.DAT**

Directory: SITEIMESORAD:, IEMIS$SYSF:

Purpose: This file contains elevations for sectors of a polar coordinate grid used for close in calculations by MESORAD. The version in IEMIS$SYSF:, C000POL.DAT, is a template.

Access: 1. MES_(NO)GR->UPDATE CASE DATA->CHANGE POLAR GRID

Structure: ASCII, Sequential.

Record 1: 80 bytes
FORMAT (A)

HEADER	C*80	File header.

Record I (I=2,37): 79 bytes
FORMAT (1X,3(F10.3,2X,F10.3,4X))

PDIST(1)	R*4	Radial distance from site center to elevation location (meters).
PELEV(J)	R*4	Elevation of jth sector at distance PDIST(1) (meters).
PDIST(2)	R*4	Radial distance from site center to elevation location (meters).
PELEV(J)	R*4	Elevation of jth sector at distance PDIST(2) (meters).
PDIST(3)	R*4	Radial distance from site center to elevation location (meters).
PELEV(J)	R*4	Elevation of jth sector at distance PDIST(3) (meters).

Filename: **CnnnPOSINP.DAT**

Directory: SITEOD2:

Purpose: This file is output from the D2PC model. It is used as input to the PARDOS model.

Access: IBS->EMERGENCY ACTIVITIES->HAZARD ANALYSIS->EXECUTE D2->Run Current D2 Case

Structure: ASCII, sequential.

Record 1: FORMAT (*)

PARDOS_REASON	C*80	Reason for any PARDOS blocking rules

Record 2: FORMAT(*)

PARD_QQQ	R*8	Quantity of run (mg)
PARD_SXS	R*8	Sigma X of run
PARD_SYS	R*8	Sigma Y of run
PARD_SZS	R*8	Sigma Z of run
PARD_TIME	R*8	Time of release (min)

Record 3: FORMAT (*)

IM	I*4	Integer index of stability class
UT	R*8	Wind speed (m/s)
HML	R*8	Height of mixing layer (m)

Record 4: FORMAT (*)

IAGN_NUM	I*4	Integer index of agent

Record 5: FORMAT (*)

ITRUECNT	I*4	Number of distances to be evaluated

Record 6-(5+ITRUECNT): FORMAT (*)

DDIST (I)	R*8	Distances from origin

Record 6+ITRUECNT: FORMAT (*)

BR	R*8	Breathing rate (liters/min)

Record 7+ITRUECNT: FORMAT (A1)

ENTER_DOS	C*1	Answer to prompt to enter dosages. Hardcoded within the model to always answer 'Y'

Record 8+ITRUECNT:

FORMAT (*)		
I_DND	I*4	Number of dosage level to enter

Record 9+ITRUECNT:

FORMAT (*)		
ARR_RECS()	R*8	Dosage levels

Record 10+ITRUECNT:

FORMAT (A1)		
ANOTHER_RUN	C*1	Answer to prompt to enter another run of PARDOS. Hardcoded within the model to always answer 'N'.

Filename: **CnnnPRM.DAT**

Directory: SITEOMESORAD:

Purpose: Provides a report of total whole body and thyroid dose to population for each advection (time) step (current and cumulative) based on IDYNEV model sector statistics for current IDYNEV model case and 31X31 MESORAD advection step matrices containing total whole body and thyroid doses.

Access:
1. MES_(NO)GR->PRINT/DISPLAY REPORTS->Dose Report
2. EESF_(NO)GR->Print/Display Reports->Dose Report

Include: MESO$INC:DATIM.INC
MESO$INC:RSP_COM.INC

Structure: ASCII, Sequential.

Record 1,10: variable length, header

```
FORMAT ('1',A,/1X,A9,2X,A8//,' DOSE TO POPULATION',//,
' Time of Day at     CURRENT 15 MINUTE PERIOD  ',
' .........CUMULATIVE........ ',/,
' End of 15 minute     THYROID      WHOLE BODY ',
'    THYROID      WHOLE BODY  ',/,
'      Period            REM            REM     ',
'      REM           REM      ',/,
' ........... ..... ............ .............',
' ............ ............ ',//)
```

RSP_IDENT	C*50	First line of MESORAD case title.
RDATE	C*9	Current date.
RTIME	C*8	Current time.

Record 11,10+ NT): 73 bytes (one record for each time step)

```
FORMAT ('  ',I2,'-',A,'-',I2,1X,I2,':',I2,4(1X,1PG13.6))
```

DDY	I*4	Day of month for time step R-10.
CMON(DMO)	C*3	Month for time step R-10.
DYR	I*4	Year for time step R-10.
DHR	I*4	Hour of day time step R-10.
DMN	I*4	Minute of hour time step R-10.
TOT_THY_REM	R*4	Total thyroid dose to population for time step R-10 (REM).
TOT_WB_REM	R*4	Total whole body dose to population for time step R-10 (REM).
CUM_TOT_THY	R*4	Total cumulative thyroid dose to population through time step R-10 (REM).

CUM_TOT_WB	R*4	Total cumulative whole body dose to population through time step R-10 (REM).

Filename: **CnnnPUFFTR.BIN**

Directory: SITEOMESORAD:

Purpose: Provides coordinate information on puffs released from MESORAD sources for each advection (time) step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Initial Particle Path
2. EESF_GR->Display Map-Related Graphics->Initial Particle Path

Include: MESO$INC:REL.INC
MESO$INC:PUFFS.INC

Structure: Binary, Direct Access. The first four records are headers. The remaining record types are repeated for each time step.

Record 1: 16 bytes, header
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record puff track data starts in (5).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II9	I*2	Last record of puff track data.

Record 2: 4 bytes, header
UNFORMATTED

OVERALL	R*4	Length of side of model area in miles.

Record 3: 16 bytes, header
UNFORMATTED

(XSOURC(M),YSOURC(M),M=1,2)

R*4 X, Y location of sources 1 and 2 in grid units (8.5 - always grid center).

Record 4: 16 bytes, header
UNFORMATTED

(XSOURC(M),YSOURC(M),M=3,4)

R*4 X, Y location of sources 3 and 4 in grid units (8.5 - always grid center).

Record K (K = NT*5 + NP):		12 bytes (NT is number of time step; NP is number of puff records previously written - the sum of previous maximum J values).
UNFORMATTED		
literal	I*4	100 (record type).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II9	I*2	Number of puffs in this time step.

Record K + I: 16 bytes (I = source index (1,4))
UNFORMATTED

HFLAG	I*4	A record type indicator (200 + I). If no puffs created for Ith source, this value is 200 and remaining fields are 0.
XP(NPUFFS(I))	R*4	X coordinate of last puff of Ith source (grid coordinates).
YP(NPUFFS(I))	R*4	Y coordinate of last puff of Ith source.
RADP(NPUFFS(I))	R*4	Radius of last puff of Ith source.

Record K + 4 + J: 16 bytes (J = puff index)
UNFORMATTED

HFLAG	I*4	A record type indicator (300 + J).
XP(J)	R*4	X coordinate of Jth puff.
YP(J)	R*4	Y coordinate of Jth puff.
RADP(J)	R*4	Radius of Jth puff.

Filename: **CnnnRESOURCE_CHECKLIST.DAT**

Directory: SITE$IP:

Purpose: This file contains all the resources for all the tasks listed for IP number nnn. Up to 99 resources for each of the 99 tasks and 99 tasks for each of the possible emergency functions are permitted.

Access:
1. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES -> CHANGE/USE TASKS -> RESOURCES
2. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES -> REPORT

Include: IEMIS$INC:IP_AC.INC

Structure: ASCII, direct access, 200 bytes/record, with record locking at the task level during operational mode. In planning mode, no record locking exists. The first record lists the number of records contained in the file.

Record 1: 200 bytes, header

```
FORMAT (I6,1X,A23,1X,A5,154X)
```

NUM_REC	I*4	Number of records in the file
CDTS	C*23	Current system date/time when file header was last modified
IP_VERSION	C*5	Current IP version

Record 2-NUM_REC+1: 200 bytes, resource record

```
FORMAT (A20,A6,A8,A32,A10,A40,A8,A8,A40,A1,27X)
```

ID	C*20	Emergency function corresponding to action
TASK_IDENT	C*6	Unique task number ID for given task in given EF
RC_TYPE	C*8	Resource type
RC_DESC	C*32	Resource description
RC_QUANT	C*10	Quantity of given resource
RC_LOC	C*40	Resource location
RC_UNIT	C*8	Unit of the resource
RC_OPER	C*8	Resource operation
RC_LOC2	C*40	Second resource location
RC_STATUS	C*1	Status of the resource

Filename: **CnnnRESOURCE_NOTES.DAT**

Directory: SITE$IP:

Purpose: This file contains all the resource notes for all the tasks listed for IP number nnn. Up to 99 lines of resource notes for each of the 99 tasks and 99 tasks for each of the possible emergency functions are permitted.

Access:
1. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES -> CHANGE/USE TASKS -> RESOURCE NOTES
2. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES -> REPORT

Include: IEMIS$INC:IP_XN.INC

Structure: ASCII, direct access, 102 bytes/record, with record locking at the task level during operational mode. In planning mode, no record locking exists. The first record lists the number of records contained in the file.

Record 0: 102 bytes, header
`FORMAT (I6,1X,A23,1X,A5,66X)`

NUM_REC	I*4	Number of records in the file
CDTS	C*23	Current system date/time when file header last modified
IP_VERSION	C*5	Current IP version

Record 1-NUM_REC: 102 bytes, resource note record
`FORMAT (A20,A6,A76)`

ID	C*20	Emergency function
TASK_IDENT	C*6	ID linking note to given task
XN_NOTES	C*76	Actual line of notes

Filename: **CnnnRSP.DAT**

Directory: SITEIMESORAD:, IEMIS$SYSF:

Purpose: This file contains run specification data for MESORAD. The version in IEMIS$SYSF:, C000RSP.DAT, is a template.

Access: 1. MES_(NO)GR->UPDATE CASE DATA->CCHANGE CONTROL PARAMETERS

Include: MESO$INC:RSP_COM.INC

Structure: ASCII, Sequential

Record 1: 4 bytes
`FORMAT (I4)`

RSP_ITEST	I*4	Testing mode. Bits 0 - 3 are examined for possible action. (Bits correspond to reports from TESTMOV, PUFF, TSTWF, and TIMER respectively.)

Record 2: 40 bytes
`FORMAT (A)`

SITE_NAME	C*40	Path to site. disk:[user.SITES.site.

Record 3: 40 bytes
`FORMAT (A)`

RSP_IDENT	C*50	1st line of case description.

Record 4: 40 bytes
`FORMAT (A)`

RSP_IDENT_2	C*50	2nd line of case description.

Record 5: 64 bytes
`FORMAT (A)`

RSP_METFIL	C*64	Name of weather observation file.

Record 6: 6 bytes
`FORMAT (F6.2)`

RSP_CGSIZE	R*4	Length of side of gridded area to be modeled (kilometers).

Record 7: 4 bytes
FORMAT (I4)

RSP_MCSIG	I*4	Sigma selection (1 - 4 corresponding to use of NSIG, ASIG, BSIG or DSIG routines).

Record 8: 27 bytes
FORMAT (27L1)

RSP_OPTARR L*4(27)
Output option array. If .true. corresponding output files are produced.

Record 9: 10 bytes
FORMAT (5I2)

RSP_STYR	I*4	Simulation start date/time - year.
RSP_STMO	I*4	Simulation start date/time - month.
RSP_STDY	I*4	Simulation start date/time - day.
RSP_STHR	I*4	Simulation start date/time - hour.
RSP_STMN	I*4	Simulation start date/time - minute.

Record 10: 4 bytes
FORMAT (2I2)

RSP_DURHR	I*4	Simulation duration time - hours.
RSP_DURMN	I*4	Simulation duration time - minutes.

Record 11: 64 bytes
FORMAT (A)

CHK_FILE	C*64	Name of checkpoint file.

Record 12: 64 bytes
FORMAT (A)

STA_FILE	C*64	Name of station definition file.

Record 13: 64 bytes
FORMAT (A)

TOP_FILE	C*64	Name of topography (elevation) file.

Record 14: 64 bytes
FORMAT (A)

TER_FILE	C*64	Name of terrain (wind slope) file.

Record 15: 1 byte
FORMAT (L1)

RSP_NUCOUT	L*4	If .true. report on nuclides deposited on ground by grid point.

Record 16: 1 byte
FORMAT (L1)

RSP_DODOSE	L*4	If .true. perform dose calculations.

Record 17: 1 byte
FORMAT (L1)

RSP_PGFLG	L*4	If .true. perform close in calculations using polar grid.

Record 18: 5 bytes
FORMAT (F5.2)

RSP_GSCALE	R*4	Factor by which modeled (gridded) area will be extended for tricking puff movement.

Record 19: 4 bytes
FORMAT (I4)

RSP_NSOURC	I*4	Number of sources to be modeled.

Record 20: 24 bytes
FORMAT (4F6.0)

RSP_ZSOURCE(J)	R*4	Height above ground of sources (meters).

Record 21-44: 64 bytes
FORMAT (A64)

RSP_RELFIL(I)	C*64	Names of release files (6 release files for each of 4 sources).

Record 45: 16 bytes
FORMAT (A1,3I5)

RSP_OUTADV	C*1	If "E" output advection arrays at end of simulation. If "I" also output at end of each advection step.
RSP_STRADV	I*4	If > 0 : Starting hour of advection array output.

RSP_HRADV	I*4	Advection step size - hours.
RSP_MNADV	I*4	Advection step size - minutes.

Record 46: 16 bytes
FORMAT (A1,3I5)

RSP_OUTCUM	C*1	If "E" output cumulative arrays at end of simulation. If "I" also output at end of each advection step.
RSP_STRCUM	I*4	If > 0 : Starting hour of cumulative array output.
RSP_HRCUM	I*4	Cumulative array output frequency - hours.
RSP_MNCUM	I*4	Cumulative array output frequency - minutes.

Filename: **CnnnSnm.DAT**

Directory: SITEIMESORAD:, IEMIS$SYSF:

Purpose: This file contains release specification data for MESORAD. Up to 24 mm files exist for each case (6 for each of 4 sources). The versions in IEMIS$SYSF:, C000Snm.DAT, are templates.

Access: 1. MES_(NO)GR->UPDATE CASE DATA->CHANGE SOURCE DETAILS

Include: MESO$INC:REL.INC
MESO$INC:DOSE.INC

Notes: *n* in filename stands for release number (range 1 - 6); *m* in filename stands for source number (range 1 - 4).

Structure: ASCII, Sequential

Record 1: 60 bytes
`FORMAT (A)`

HEADER	C*60	File header.

Record 2: 17 bytes
`FORMAT (1X,5I2,1X,I3,I2)`

RYR	I*4	Release start date/time - year.
RMO	I*4	Release start date/time - month.
RDY	I*4	Release start date/time - day.
RHR	I*4	Release start date/time - hour.
RMN	I*4	Release start date/time - minute.
RDHR	I*4	Release duration - hour.
RDMN	I*4	Release duration - minute.

The following record type is provided, if no dose calculations are required.

Record 3: variable length
`FORMAT (variable)`

NAME	C*7	Nuclide name.
NUCRATE	R*4	Mass of NAME.

The following record type is provided if dose calculations are required.

Record K (K = 3, N + 2): variable length (maximum value of I is 50)
`FORMAT (variable)`

NAME(I)	C*7	Nuclide name.
NUCRATE(I)	R*4	Release of nuclide NAME(I) (Ci).

Filename: **CnnnSCRIPT.DAT**

Directory: SITEOD2:

Purpose: This file contains the D2 variables passed to the model in the order they were passed to the model.

See Also: CnnnD2LOG.DAT

Access: IBS->EMERGENCY ACTIVITIES->HAZARD ANALYSIS->EXECUTE D2->Run Current D2 Case

Notes: This file differs from CnnnD2LOG.DAT in that this file only lists the inputs to the model, whereas CnnnD2LOG.DAT also echoes all the model's input and output.

Structure: ASCII, sequential, variable.

Record 1-n: 80 bytes (variable format summary information)

Filename: **CnnnSLICE.DAT**

Directory: SITEOD2:

Purpose: This file contains dosage attributes for the track.

See Also: SLICEnnn.DAT

Access:
1. D2_(NO)GR
2. IBS -> ANALYZE TRACK -> PRINT D2 REPORT
3. IBS -> EMERGENCY ACTIVITIES -> HAZARD ANALYSIS -> PRINT D2 REPORT

Structure: ASCII, sequential, variable format, <= 80 bytes/record. It has six lines of header information followed by up to 3*WS_SLICE_MAX dosage attribute records.

Record 1-2: Header lines

```
FORMAT (' NUMBER OF DOSAGE LEVELS: ', 1X, I4, /)
```

WS_SLICE_N	I*4	Number of dosage levels (0 to WS_D2_DC_MAX)

Record 3-(WS_SLICE_N + 2): Dosage levels. One record for each level in the file.

```
FORMAT ('A1,)`,1X,F7.1,1X,'MG-MIN/M3')
```

WS_LABEL	C*1	Letter labeling dosage level
WS_DOSAGE	R*8	Dosage Level

Record (WS_SLICE_N + 3) - (WS_SLICE_N + 7):

```
FORMAT (/'DIST.  TOTAL DOS  TIME (min)   DOSAGE TIME  ...'/
'  (m)  (mg-min/m3)  FOR CLOUD    (%)  (min) ...'/
'                   TIP   TAIL'/
'------- --------- ------ ------ ----- ------ ...')
```

Record WS_SLICE_N + 8 - n:

Record format varies depending on PARDOS output. Only necessary records will be used; for example, if there are 5 levels, Records K1 and K2 will be contained in the file, while Record K3 will not (file will proceed from K2 to K1 for the next distance). If WS_SLICE_N = 0 then only the tip and tail times will be written to the file for each distance. Generically, the format is:

Record K1: (I=1, WS_SLICE_N)

```
FORMAT (F7.0,1X,E9.4,2F7.1,3(2X,A1,1X,F5.1,F7.1))
```

WS_SLICE_DIST(J)	R*8	Distance
WS_SLICE_TD(J)	R*8	Total dosage to the given distance

WS_SLICE_TIP(J)	R*8	Time for cloud tip to reach point
WS_SLICE_TAIL(J)	R*8	Time for cloud tail to reach point
WS_SLICE_LET(I,J)	C*1	Dosage letter for given distance (I=1,3)
WS_SLICE_PN(I,J)	R*8	% of Ith dosage track at given distance (I=1,3)
WS_SLICE_TN(I,J)	R*8	Time for Ith dosage track to reach (I=1,3)

Record K2: (not in file if WS_SLICE_N<4)

```
FORMAT (31X,3(2X,A1,1X,F5.1,F7.1))
```

WS_SLICE_LET(I,J)	C*1	Dosage letter for given distance (I=4,6)
WS_SLICE_PN(I,J)	R*8	% of Ith dosage track at given distance (I=4,7)
WS_SLICE_TN(I,J)	R*8	Time for Ith dosage track to reach (I=4,7)

Record K3: (not in file if WS_SLICE_N<7)

```
FORMAT (31X,3(2X,A1,1X,F5.1,F7.1))
```

WS_SLICE_LET(I,J)	C*1	Dosage letter for given distance (I=7,9)
WS_SLICE_PN(I,J)	R*8	% of Ith dosage track at given distance (I=7,9)
WS_SLICE_TN(I,J)	R*8	Time for Ith dosage track to reach (I=7,9)

Record K4: (not in file if WS_SLICE_N<10)

```
FORMAT (33X,A1,1X,F5.1,F7.1)
```

WS_SLICE_LET(I,J)	C*1	Dosage letter for given distance (I=10)
WS_SLICE_PN(I,J)	R*8	% of Ith dosage track at given distance (I=10)
WS_SLICE_TN(I,J)	R*8	Time for Ith dosage track to reach (I=10)

Filename: **CnnnSTA.DAT**

Directory: SITEIMESORAD:, IEMIS$SYSF:

Purpose: This file contains weather station definition data for MESORAD. The version in IEMIS$SYSF:, C000STA.DAT, is a template.

Access: 1. MES_(NO)GR->UPDATE CASE DATA->CHANGE STATION DEFINITIONS

Include: MESO$INC:METEOR.INC

Structure: ASCII, Sequential.

Record 1: 80 bytes
`FORMAT (A)`

LINE	C*80	File header.

Record K (K = 2, MAXSTA+1): 53 bytes
`FORMAT (1X,A8,18X,2(2X,F7.2),4X,I4)`

NAMST(N)	C*8	Name of met station.
XDIST(N)	R*4	Offset from grid center (kilometers) - Cartesian convention.
YDIST(N)	R*4	Offset from grid center (kilometers) - Cartesian convention.
ZDIST(N)	I*4	Elevation (meters).

Filename: **CnnnSTATIN.BIN**

Directory: SITEOMESORAD:

Purpose: Provides wind speed and direction information on weather stations for each advection (time) step.

Access: 1. MES_(NO)GR

Include: MESO$INC:METEOR.INC

Notes: This file is created by MESORAD but no existing display software uses it. It could be used to provide a display similar to that of the Windfield display, available in MES_GR and EESF_GR but this has not been developed.

Structure: Binary, Direct Access. The first two records are headers. The remaining record types are repeated for each time step.

Record 1: 16 bytes, header
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record station data starts in (3).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II9	I*2	Last record of station data.

Record 2: 8 bytes, header
UNFORMATTED

OVERALL	R*4	Length of side of model area in miles.
MAXSPD	R*4	Maximum speed of wind (miles/hour) - set to fixed value of 45.

Record K (K = 2 + NT + NS): 14 bytes (NT is number of time step; NS is number of station records previously written - the sum of previous maximum J values).

UNFORMATTED

literal	I*4	100 (record type).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
ACTSTA	I*4	Number of active stations in this time step.

Record K + J: 20 bytes (J = active station index)
UNFORMATTED

HFLAG	I*4	A record type indicator (200 + J).
XSTA(J)	R*4	X coordinate of Jth station.
YSTA(J)	R*4	Y coordinate of Jth station.
SFCDIR(J)	I*4	Surface level wind direction at station J (degrees).
SFCSPD(J)	I*4	Surface level wind speed at station J (meters/second).

Filename:	**CnnnSTEP.DAT**
Directory:	SITEOD2:
Purpose:	This file contains interpolated PARDOS time information used for D2 animation.
See Also:	CnnnSLICE.DAT
Access:	1. IBS->ANALYZE TRACK->ANIMATE D2 TRACK 2. IBS->EMERGENCY ACTIVITIES->HAZARD ANALYSIS->ANIMATE D2 TRACK
Include:	IEMIS$INC:D2_STEP.INC
Notes:	This file contains header records followed by one record for each time step. The number of time step records in the file is dependent on the length of the D2 case on which the file is based. The last tail time shown in the D2 case output is used to establish the number of minutes of animation and (at present) one time step record is produced for each minute of animation. The record structure provides for the specification of a time step size of other than one minute, but this size is always set to one minute.
Structure:	Binary, direct access, fixed length, 48 bytes
Record 1:	Header record

	TS_SEC	I*4	Time step length (seconds)
	XDD(1)	R*4	Final distance of lowest calculated dose (meters)
	LAST_TAIL_TIME	I*4	Time in minutes from start of release to passage of plume tail from last calculated point in file CnnnSLICE.DAT.
	constants	R*4	Reserved for future use (9 fields)
Record 2:	WIND	R*4	Wind/plume direction (degrees)
	XORG	R*8	Origin of plume (longitude)
	YORG	R*8	Origin of plume (latitude)
	constants	R*4	Reserved for future use (7 fields)
Record 3,N+2:			
	TIP_DISTANCE	R*4	Distance (meters) of plume tip from origin at the end of time step n.
	TAIL_DISTANCE	R*4	Distance (meters) of plume tail from origin at the end of time step n.
	TS_XDD	R*4	Array of 10 distances (meters) of calculated dosages from origin at the end of time step n.

Filename: **CnnnSUM.RPT**

Directory: SITEODYNEV:

Purpose: This file contains a 1-screen summary of IDYNEV simulation results extracted from CnnnCOD.RPT and reformatted for screen display.

See Also: CnnnCOD.RPT

Access: 1. IBS -> EMERGENCY ACTIVITIES -> EVACUATION

Notes: Variable names are not provided here because the content of this report is obtained by concatenation of strings of text from another report.

Structure: ASCII, sequential, variable length, 80 bytes

Record 1-16:

```
FORMAT (
' IDYNEV   SUBNETWORK STATISTICS'/
' '/
' '/
' VEHICLE-MILES',10X,F10.2/
' VEHICLE-MINUTES',8X,F10.2/
' VEHICLE-TRIPS(EST.)',5X,I6/
' PCT OF VEHS THAT STOPPED',3X,F7.3/
' MOVING/TOTAL TRIP TIME',7X,F5.3/
' AVG. SPEED(MPH)',13X,F5.2
' AVG. QUEUE CONTENT',8X,F6.1,' VEH.'/
' DELAY/VEH',16X,F8.2 ,'SEC.'/
' TOTAL DELAY',11X,F10.1,' MIN'/
' DELAY/VEH-MILE',15X,F4.2,' MIN/V-MILE'/
' TRAVEL TIME/VEH-MILE',9X,F4.2,' MIN/V-MILE/
' '/
' IDYNEV completed processing thru ',I3,1X,I2,1X,I2)
```

Filename: **CnnnTASK_LINKS.DAT**

Directory: SITE$IP:

Purpose: This file contains all the task dependencies for a particular IP.

See Also: CnnnMASTER_CHECKLIST.DAT

Access: IBS->STATUS BOARD->IMPLEMENTING PROCEDURES->CHANGE TASKS->option "L"

Include: IEMIS$INC:IP_TL.INC, IEMIS$INC:USE_TASK_2.INC

Notes: The task links are only used in planning mode.

Structure: ASCII, direct access, 60 bytes/record.

Record 0: 60 bytes, header

```
FORMAT (I6, 1X, A23, 1X, A5, 1X, A23)
```

NUM_REC	I*4	Number of record in the file
CDTS	C*23	Current system date/time when file last modified
IP_VERSION	C*5	Current IP version
T_BASE_DT	C*23	Base time for IP

Record 1-NUM_REC: 60 bytes, task link record

```
FORMAT (A20,A6,1X,A6,1X,A2,1X,A6,A1,16X)
```

TL_EF_NAME	C*20	Emergency function
TL_CUR_ID	C*6	Current task ID
TL_LINK_ID	C*6	ID of linked task
TL_LINK_TYPE	C*2	Type of link "SS","FF","FS"
TL_LAG_MIN	C*6	Number of lag minutes
TL_PRED_OR_SUC	C*1	Is current task predecessor or successor to the linked task?

Filename: **CnnnTASK_NOTES.DAT**

Directory: SITE$IP:

Purpose: This file contains all the task notes for all the tasks listed for IP number nnn. Up to 99 lines of task notes for each of the 99 tasks and 99 tasks for each of the possible emergency functions are permitted.

Access: 1. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES

Include: IEMIS$INC:IP_XN.INC

Structure: ASCII, direct access, 102 bytes/record, with record locking at the task level during operational mode. In the planning mode, there is no locking. The first record lists the number of records contained in the file.

Record 1: 102 bytes, header
`FORMAT (I6,1X,A23,1X,A5,66X)`

NUM_REC	I*4	Number of records in the file
CDTS	C*23	Current system date/time when file header was last modified
IP_VERSION	C*5	Current IP version

Record 2-NUM_REC: 102 bytes, task note record
`FORMAT (A20,A6,A76)`

ID	C*20	Emergency function
TASK_IDENT	C*6	ID linking note to given task
XN_NOTES	C*76	Actual line of notes

Filename: **CnnnTER.DAT**

Directory: SITEIMESORAD:, IEMIS$SYSF:

Purpose: This file contains terrain data to be used in modifying the wind field. The version in IEMIS$SYSF:, C000TER.DAT, is a template.

Access: 1. MES_(NO)GR->UPDATE CASE DATA->CHANGE TERRAIN

Include: MESO$INC:WFMOD.INC

Structure: ASCII, Sequential.

Record 1: 40 bytes
FORMAT (A)

LINE	C*40	File header.

Record K (K = 2, N): 21 bytes
FORMAT (1X,I2,1X,I2,1X,F4.0,1X,F4.0,1X,F4.0)

I	I*4	Grid X index for coefficients (1 - 16: WEST - EAST).
J	I*4	Grid Y index for coefficients (1 - 16: SOUTH - NORTH).
TANGLE(I,J)	R*4	Degrees from horizontal (0.0 - 180: NE=45, N=90, NW=135, W=0, SW=45, S=90, SE=135, E=0).
COEFF1(I,J,1)	R*4	Coefficient for transport to the right (0=vertical, 1=flat).
COEFF1(I,J,2)	R*4	Coefficient for transport to the left (0=vertical, 1=flat).

Filename: **CnnnTOP.DAT**

Directory: SITEIMESORAD:, IEMIS$SYSF:

Purpose: This file contains ground level elevation to describe Cartesian elevation data. The version in IEMIS$SYSF:, C000TOP.DAT, is a template.

Access: 1. MES_(NO)GR->UPDATE CASE DATA->CHANGE CARTESIAN GRID

Include: MESO$INC:TOPOGR.INC

Structure: ASCII, Sequential.

Record 1: 80 bytes
FORMAT (A)

LINE	C*80	File header.

Record K (K = 2, 32): 125 bytes (31 records corresponding to index J below)
FORMAT (1X,31I4)

(TTOPO(I,J),I=1,31)	I*4	Elevation above sea level (meters).

Filename: **CnnnTOTWBD.ADV**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of total whole body dose from inhalation, ground, and cloud shine for each advection (time) step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Curr Dose - Total Whole Body
2. EESF_GR->Display Map-Related Graphics->Curr Dose - Total Whole Body

Include: MESO$INC:DOSE.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((DOSARR(K,32-J,N-6),K=1,31),J=1,31)
R*4 Dose levels in Cartesian grid (REM) (N=19).
((PGDARR(I,J,N-6),I=1,36),J=1,3)
R*4 Dose levels in polar grid (REM) (N=19).

Filename: **CnnnTOTWBD.CUM**

Directory: SITEOMESORAD:

Purpose: Provides a 31X31 Cartesian and 3*36 polar coordinate matrix of total whole body dose from inhalation, ground, and cloud shine for each advection (time) step - cumulative from start to current time step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Cumul Dose - Total Whole Body
2. EESF_GR->Display Map-Related Graphics->Cumul Dose - Total Whole Body

Include: MESO$INC:DOSE.INC

Structure: Binary, Direct Access.

Record 1: 28 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.
OVERALL	R*4	Length of side of model area in miles.
RMIN(N)	R*4	Minimum matrix data value for file N.
RMAX(N)	R*4	Maximum matrix data value for file N.

Record R (R = 2, NT): 4276 bytes (one record for each time step)
UNFORMATTED

((DOSARR(K,32-J,N-6),K=1,31),J=1,31)
R*4 Dose levels in Cartesian grid (REM) (N=20).
((PGDARR(I,J,N-6),I=1,36),J=1,3)
R*4 Dose levels in polar grid (REM) (N=20).

Filename:	**CnnnTV.DAT**		
Directory:	SITEODYNEV:		
Purpose:	This file contains time stamps and link volume statistics extracted from CnnnCOD.RPT.		
See Also:	CnnnCOD.RPT		
Access:	IBS->EMERGENCY ACTIVITIES->EVACUATION		
Include:	EESF$INC:NODEPLOT.INC		
Structure:	Binary, sequential, variable length, max length unknown		
Record 1:	**LAST_TIME**	I*4	Number of time steps in output report
	NLINK_TIME	I*4	Number of links
	NTIME(J)	C*22	Time stamp array (one entry for each time stamp)
Record 2,N+1:			
	LINKS	I*4	Number of links in case nnn
	V_MAX	R*4	Maximum vehicle volume (VPH) for each link N.

Filename: **CnnnV.DAT**

Directory: SITEIMESORAD:, IEMIS$SYSF:

Purpose: This file contains evacuation data (sector statistics) produced by IDYNEV and used to compute dose to population. The version in IEMIS$SYSF:, C000V.DAT, is a template.

Access: MES_(NO)GR

Include: MESO$INC:TOPOGR.INC

Notes: The portion of the system that uses this data is still limited to the use of 15-minute time periods. Record indexing is based on this concept.

Structure: Binary, Direct Access.

Record 1: 16 bytes
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record sector data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of sector data.

Record K (K = 2, n): 1600 bytes (one record for each 15 minute time step)
UNFORMATTED

V	R*4(20,20)	Volume of vehicles in sectors, both evacuating and waiting, reduced by specified sheltering factors.

Filename: **CnnnVA.DMS**

Directory: MAP$DATA$VA:

Purpose: This file is the graphical representation of the Vapor contours calculated by the CHEMS model for the nnn case scenario. LIMITVA.BIN points to the current case.

See Also: MAP$DATA$VA:LIMITVA.BIN

Access: 1. CHEMS_(NO)GR

Structure: DMS Format

Filename: **CnnnWINDFD.BIN**

Directory: SITEOMESORAD:

Purpose: Provides 16X16 matrices of wind speed and direction (wind fields) for each advection (time) step.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Windfield
2. EESF_GR->Display Map-Related Graphics->Windfield

Structure: Binary, Direct Access. The first record is a header. The remaining record types are repeated for each time step.

Record 1: 28 bytes, header
UNFORMATTED

II1	I*2	Day data starts (0).
II2	I*2	Hour data starts (0).
II3	I*2	Minute data starts (15).
II4	I*2	Record windfield data starts in (2).
II5	I*2	Day data ends.
II6	I*2	Hour data ends.
II7	I*2	Minute data ends.
II8	I*2	Last record of station data.
OVERALL	R*4	Length of side of model area in miles.
literal	R*4	0.0 (Minimum speed of wind (miles/hour).
MAXSPD	R*4	Maximum speed of wind (miles/hour) - set to fixed value of 45.

Record K (K = NT*2): 1024 bytes (NT is number of time step)
UNFORMATTED

((BINDIR(K,17-J),K=1,16),J=1,16)
R*4 Calculated wind direction (degrees) at center of each cell in 16X16 met grid.

Record K + 1: 1024 bytes
UNFORMATTED

((BINSPD(K,17-J),K=1,16),J=1,16)
R*4 Calculated wind speed (meters/second) at center of each cell in 16X16 met grid.

Filename: **CnnnXmm.BIN**

Directory: SITEOMESORAD:

Purpose: Provides a report of thyroid and total whole body doses projected at specified checkpoints for each advection (time) step.

Access:
1. MES_(NO)GR->PRINT/DISPLAY REPORTS->Dose Report
2. EESF_(NO)GR->Print/Display Reports->Dose Report

Include: MESO$INC:RSP_COM.INC
MESO$INC:DATIM.INC
MESO$INC:CHECK.INC

Notes: The *mm* suffix in filename corresponds to a checkpoint number. One file is written for each checkpoint specified.

Structure: ASCII, Sequential. The first 14 records are a header. The next record type is repeated for each time step having a non-zero dose rate at checkpoint mm. If no time period has a non-zero dose, the last record type is written instead.

Record 1,14: Variable length, header

```
       FORMAT ('1',A/1X,A /1X, A9, 2X,A8//,
+      ' ALL TIMES THAT DO NOT APPEAR HAVE VALUES OF ZERO',//,
+      ' DOSE AT CHECKPOINT ',A8,5X,'CHECKPOINT # ',I2//,
+      ' Time of Day at    .......CURRENT 15 ',
+      'MINUTE TIME PERIOD..... ',/,
+      ' End of 15 minute      I-131        ',
+      ' PLUME        GROUND      ',/,
+      '      Period          microCi/cc     ',
+      'mrem/hr       mrem/hr    ',/,
+      ' ........... ..... ............. ',
+      '............. ............. ',//)
```

RSP_IDENT	C*50	First line of MESORAD case description.
RSP_IDENT_2	C*50	Second line of MESORAD case description.
RDATE	C*9	Current date.
RTIME	C*8	Current time.
CHKPNT(J)	C*8	Checkpoint name.
J	I*4	Checkpoint number.

Record K (K = 14 + NT): 59 bytes. (NT ranges from 1 to number of time steps having non-zero doses at checkpoint mm).

```
FORMAT ('  ',I2,'-',A,'-',I2,' ',I2,':',I2,3(1X,1PG13.6))
```

DDY	I*2	Day of month of time step.
MON(DMO)	C*3	Month of time step.

DYR	I*2	Year of time step.
DHR	I*2	Hour of time step.
DMN	I*2	Minute of time step.
THYC	R*4	Iodine-131 air concentration (microCuries/cc) for current time step.
PLMDR	R*4	Dose rate from plume (mrem/hr) for current time step.
GRDDR	R*4	Dose rate from deposition on the ground (mrem/hr) for current time step.

Record 15: 27 bytes. (This record only appears if no time period provides a non-zero dose to checkpoint mm.)

```
FORMAT ('NO DOSE AT THIS CHECKPOINT')
```

Filename: CALLLIST.DAT

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: Transferred from onpost to offpost. This free-format file contains the people or agencies the onpost expects to call, if an event occurs.

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> CALL LIST
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. ONPOSTSIM
4. EVENT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes: When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file has WS_CL_MAX (100) records with person/agency and phone numbers for each.

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K1 (K1=J+2;J=1,N;1<=N<=WS_CL_MAX):
FORMAT (A80)

WS_CL_DESC(J)	C*80	Person/agency/phone #s

Filename: **CAS_SUM.DAT**

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: This file is sent from onpost to offpost, and provides a summary of onpost casualties.

See Also: OFF_CAS.DAT

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> CASUALTY SUMMARY
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. ONPOSTSIM
4. EVENT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes: A similar file, OFF_CAS.DAT, is sent from offpost to onpost but it differs slightly in format and content. When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, variable format, <= 80 bytes/record
This file has 2 header records followed by 9 records with a 33-character description and a number, followed by up to WS_CS_MAX (5) records describing the event. Generically, the format for the first nine records is: `FORMAT (A33,I7)`

Record 1-2: File header and description
```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record 3-11:
```
FORMAT ('Fatalities',T32,':',1X,I7/
'Hospital Admissions',T32,':',1X,I7/
'Outpatient Care',T32,':',1X,I7/
```

```
'RMAT Total',T32,':',1X,I7/
'Chem Agent Exposure',T32,':',1X,I7/
'Fracture',T32,':',1X,I7/
'Anxiety reaction',T32,':',1X,I7/
'Non related',T32,':',1X,I7/
'Number of civilian casualties:',T32,':',1X,I7)
```

WS_CS_FAT	I*4	Fatalities
WS_CS_HA	I*4	Hospital admissions
WS_CS_OC	I*4	Outpatient care casualties
WS_CS_T	I*4	Total casualties
WS_CS_CAE	I*4	Agent exposure casualties
WS_CS_F	I*4	Fracture casualties
WS_CS_AR	I*4	Anxiety reaction casualties
WS_CS_NR	I*4	Unrelated casualties
WS_CS_NCC	I*4	Civilian casualties

Record K (K=J+11;J=1,N;0<=N<=WS_CS_MAX):

```
FORMAT (A80)
```

WS_CS_REPORT(J)	C*80	Comment

Filename: **CASEINDEX.DAT**

Directory: SITE$DIR:, IEMIS$SYSF:

Purpose: This file is used to maintain an index of case numbers in use for all models for a given site.

Access: 1. IBS

Include: IEMIS$INC:JOB_ENV.INC

Notes: The version of this file in IEMIS$SYSF: is a template and is named CASEINDEX.DAT_TEMPLATE.

Structure: ASCII, sequential, with 10 records for each of 6 models

Each 10-record set contains 100 single character flags showing whether an associated case exists, is checked out, or is a Maximum Credible Event (MCE) case, followed by the model name to which the cases apply. 1000 case flags are provided for each model. Case index range (I) is 0:999. Thus, within each 10 record set, the structure is INDEX_RECORDS(I,J); I=N-100,N-1 where N is the record index within the set multiplied by *100. J, the index of the model, is fixed within a 10-record set and ranges from 1 to 6 for the 6 models.

Record 1,60:

```
FORMAT (100A1,A8)
```

INDEX_RECORDS(I,J)	C*1	I=N-100:N-1 (see structure above) Y - Yes, this case exists. N - No case exists. A - Case exists, but is currently checked out by a user. M - Case exists, but modifications to case will not be allowed. This case is a maximum credible event (MCE) reference case.
MODEL_NAMES(J)	C*8	"MESORAD", "IDYNEV", "CHEMS", "OSPM", "D2", "IP" - Indicates to which model this record applies.

Filename: **CASEINDEX.DOC**

Directory: SITE$DIR:

Purpose: This file provides the user with a easy-to-read format of the CASEINDEX.DAT file. Only the existing cases are listed.

See Also: CASEINDEX.DAT, CASEINDEX.LST

Access: 1. SHOWCNX

Structure: ASCII, sequential, variable length, 11 bytes/record

Record 1-N:

FORMAT(I3, 1X, A8)

CASE	N/A	000 through 999
NAME	N/A	"MESORAD", "IDYNEV", "CHEMS", "OSPM", "D2", "IP" - Indicates to which model this record applies.

Filename: **CASEINDEX.LST**

Directory: SITE$DIR:

Purpose: This file provides the user with a easy-to-read format of the CASEINDEX.DAT file.

See Also: CASEINDEX.DAT, CASEINDEX.DOC

Access: 1. LISTCNX

Structure: ASCII, sequential, variable length, 11 bytes/record

Record 1-N:

FORMAT(A8, ' (', A1, '): ', I3)

NAME	N/A	"MESORAD", "IDYNEV", "CHEMS", "OSPM", "D2", "IP" - Indicates to which model this record applies.
FLAG	N/A	Case existence flag Y - Yes, this case exists. N - No case exists. A - Case exists, but is currently checked out by a user. M - Case exists, but modifications to case will not be allowed. This case is a maximum credible event (MCE) reference case.
CASE	N/A	000 through 999

Filename: CCS_TRACK.DAT

Directory: SITE$DIR:

Purpose: The model case tracking file - maintains a list of checked out model cases and the users who have checked them out.

Access: 1. IBS-> Setup -> Job Environment -> Copy Case or Site File(s)

Include: IEMIS$INC:CCS_TRACK.INC

Notes: Checkin and checkout are available in planning mode only.

Structure: ASCII, direct access, 80 bytes/record.

Record 1-n:

```
FORMAT(A12,1X,A4,1X,A8,1X,I3.3,1X,A23,20X,A5)
```

USER	C*(*)	User Id
SITE	C*(*)	Site user checked in/out to
T_MODEL	C*8	Model of case user checked in/out
ICASE	I*4	Case number user checked in/out
WORK_CDTS	C*23	Timestamp when checked in/out
CCS_TRACK_VERS	C*5	File format version number

Filename: **CHEM.DAT**

Directory: IEMIS$CHEMS:

Purpose: Maintains a list of all chemical codes for which chemical property files of the form xxxPROP.DAT exist.

See Also: xxxPROP.DAT

Access: 1. CHEMS_(NO)GR

Structure: ASCII, Sequential

Record 1,4: 80 bytes
FORMAT (A3,A77)

A1	C*3	Dummy.
A2	C*77	Dummy.

Record 5,n: 80 bytes
FORMAT (A3,A77)

A1	C*3	Chemical code.
A2	C*77	Property string.

Filename: **COLORTBL.DAT**

Directory: IEMIS$SYSF:

Purpose: This file links the IBS pen number with the device-specific color table parameters.

Access: This file is accessed during the initialization process of most graphic programs in IBS.

Notes: The logical MAP$COLOR$TABLE is used to reference this file.

Structure: ASCII, sequential, variable length record

Record 1-NCOLORS:

```
FORMAT (3(F4.0,1X),I1)
```

Comments describing the color defined by the combinations of red, green, and blue indicated in the record may be added after the 16th byte of the record and are not read by the software.

RED	R*4	Intensity of red for this color. Values 0 through 100 permissible.
GREEN	R*4	Intensity of green for this color. Values 0 through 100 permissible.
BLUE	R*4	Intensity of blue for this color. Values 0 through 100 permissible.
IBLINK	I*1	Indicates if this is a blinking color. Values are 1 for blinking and 0 for no blinking.

Filename:	**COMBINED.LOG**
Directory:	OFF$DIR:, SYS$LOGIN:
Purpose:	This file contains important mail selected from VAX Mail and extracted entries from SPOCC_ACTIVITY.LOG, SPOCC_EVENT.LOG and WATCH_EVENT.LOG that meet user criteria for use in reports.
Access:	1. IBS -> MESSAGE BOARD -> CREATE/VIEW COMMON LOG 2. IBS -> MESSAGE BOARD -> CREATE/VIEW PRIVATE LOG
Notes:	Common log reports are located in OFF$DIR: while private log reports are located in SYS$LOGIN:.
Structure:	ASCII, sequential, variable record length
Record n:	101 bytes `FORMAT (A14,I7,A80)`

DATE14	C*14	Date/Time stamp of record creation (YYMMDDHHMMSSHH)
I	I*4	Index of DETAIL_RECORD within message
DETAIL_RECORD(I)	C*80	Line from VAX Mail message

Filename: CR_RISK_AREA.DAT

Directory: SITE$OFF:

Purpose: This file contains the known polygon identifiers comprising the current risk area.

Access:
1. This file is accessed after a risk area has been defined and is saved. This happens within RISK AREA ANALYSIS after the user has created and saved a risk area. This file is also read on startup of the IBS application in order to display the current risk area.

Include: IEMIS$INC.RISK_AREA.INC

Structure: ASCII, direct access, 64 bytes/record, shared and locked in operational mode. Contains the Risk Area Polygon record and one or more records which identify the EPZs and IRZs at risk.

Record 0: header

```
FORMAT (I6,1X,A23,1X,A5,28X)
```

NUM_RECS	I*4	Number of records in the file
DATE_TIME	C*23	Date and time last updated
VERSION	C*5	Version of this file

Record 1: risk area polygon

```
FORMAT (I4,1X,A40,19X)
```

Contains the id and name of the polygon used to establish the risk area.

RISK_POLY_ID	I*4	Unique known polygon id which defines the risk polygon
RISK_POLY_NAME	C*40	Unique risk polygon name from known polygons file

Record 2-n: EP Zones

```
FORMAT (I4,1X,A5,54X)
```

Contains the EP zones that are at risk.

EPZ_ID	I*4	Unique EP zone ID from known polygons file
EPZ_NAME	C*5	Unique EP zone Name from known polygons file

Filename: **CR_RISK_AREA.DMS**

Directory: SITE$OFF:

Purpose: This file contains the areas comprising the current risk area and is the DMS equivalent of the CR_RISK_AREA.DAT file.

See Also CR_RISK_AREA.DAT

Access: 1. IBS -> EMERGENCY ACTIVITIES -> DIRECTION/CONTROL -> RISK AREA ANALYSIS

This file is created by the Risk Area Analysis function, but is not used by any part of IBS thereafter. It is merely a snapshot of the current risk area.

Structure: DMS format. Locked for write in operational mode.

Filename: **CRASHFLG.DAT**

Directory: SITEICHEMS:, IEMIS$SYSF:

Purpose: This file provides a description of missing data in CHEMS model execution.

Access: 1. CHEMS_(NO)GR

Include: CHEMS$INC:CRASHFLG.INC

Notes: Portions of the CHEMS software are only provided as a binary object library. The file is necessary for the execution of the CHEMS models. Data structures in CHEMS provide for up to 25 pairs of ERRPRP values. The version in IEMIS$SYSF: is a template and is accessed by NEWSITE.

Structure: ASCII, sequential

Record 1: Variable
FORMAT (*)

ICRASH	I*4	Signal; if 2 then warn user of the possibility of missing chemical property values

Record I; 0≤I≤25 : Variable
FORMAT (*)

ERRPRP(I,1)	R*4	Unknown
ERRPRP(I,2)	R*4	Unknown

Filename:	**D2_MASTER_FILE.DAT**
Directory:	SITEID2:
Purpose:	This file lists several D2 input values for each D2 case. These values are used for criteria searches when selecting a D2 case.
See Also:	FIXD2MSTR in the *IBS Utilities Guide*.
Access:	1. FIXD2MSTR 2. IBS-> EMERGENCY ACTIVITIES -> CHANGE TRACK CASE -> Choose D2 case by SEARCH FOR D2 CRITERIA
Include:	IEMIS$INC:CHOOSE_D2_CASE_2.INC
Notes:	If this information is not intact, criteria searches for D2 will not work. See FIXD2MSTR in the *IBS Utilities Guide*.
Structure:	ASCII, direct access, 172 bytes/record.
Record 1-n:	172 bytes, D2 case input summary record `FORMAT (A1, 1X, A1, 1X, A80, 1X, A3, 1X, A3, 1X, A2, 1X, A1, 1X, E13.6, 1X, E13.6, 1X, E13.6, 1X, I10, 1X, I10, 1X, I10)`

D2_EXISTS	C*1	Flag specifying input exists
D2_OUT_EXISTS	C*1	Flag specifying output exists
D2_DESCRIP	C*80	Description for D2 case
D2_MUN	C*3	Munition code
D2_REL	C*3	Release code
D2_AGENT	C*2	Agent code
D2_STB	C*1	Stability code
D2_WND	R*8	Wind speed
D2_WND_DIR_LST	R*8	Wind direction
D2_QUANT	R*8	Quantity in mg
D2_L_DIST	I*4	(not used)
D2_M_DIST	I*4	(not used)
D2_H_DIST	I*4	(not used)

Filename: **D2INPnnn.DAT**

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: Transferred from onpost to offpost. This file is the one used for input to the D2 model for D2 case number *nnn*. It will be available as the basis for a new offpost D2 case.

See Also: CnnnD2INP.DAT, OFF$SENDDIR:D2INPnnn.DAT

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> D2 INPUT
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. ONPOSTSIM
4. EVENT

Include: IEMIS$INC:HEADER.INC

Notes: This file is identical to CnnnD2INP.DAT, except for the addition of the transfer header. When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, variable format, <= 80 bytes/record.

Record 1-2: File header and description

```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record N: See CnnnD2INP.DAT

Filename: **D2INPnnn.DAT**

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file is a copy of CnnnD2INP.DAT, with the addition of the transfer header records. It is not used by EMIS.

See Also: CnnnD2INP.DAT, ONSITE$DIR:D2INPnnn.DAT

Access:
1. IBS -> MESSAGE BOARD -> CREATE IBS -> ONPOST RPT
2. IBS -> MESSAGE BOARD -> RESEND IBS -> ONPOST RPT

Include: IEMIS$INC:HEADER.INC

Notes: This file is identical to CnnnD2INP.DAT, except for the addition of the transfer header.

Structure: ASCII, sequential, variable format, <= 80 bytes/record.

Record 1-2: File header and description
`FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)`

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record N: See CnnnD2INP.DAT

Filename: **D2LOGnnn.DAT**

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: Transferred from onpost to offpost. This file is an output report from D2 for case number *nnn*. Its format is quite variable, so it will be treated as a free-format file. Except for extracting some header information, it is not used by IBS, although it may be printed or displayed.

See Also: CnnnD2LOG.DAT, OFF$SENDDIR:D2LOGnnn.DAT

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> D2 LOG
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. ONPOSTSIM
4. EVENT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes : This file is identical to CnnnD2LOG.DAT, except for the addition of a file transfer header line placed by the onpost system. When the EVENT program finds this file in ONSITE$DIR, it processes it and then moves it to OFF$DIR for use by other applications. The file is a free format log file, hence most of the data is ignored. Refer to CnnnD2LOG.DAT for more information.

Structure: ASCII, sequential, variable format, <= 80 bytes/record
This file contains up to WS_D2LOG_MAX (1000) records of anything.

Record 1-2: File header and description
`FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)`

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K1 (K1=J+2;J=1,N;1<=N<=WS_D2LOG_MAX):
`FORMAT (A80)`

WS_D2LOG_DESC(J)	C*80	Variable contents

Filename: **D2LOGnnn.DAT**

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file is a copy of CnnnD2LOG.DAT, with the addition of the transfer header records. It is not used by EMIS.

See Also: CnnnD2LOG.DAT, ONSITE$DIR:D2LOGnnn.DAT

Access:
1. IBS -> MESSAGE BOARD -> CREATE IBS -> ONPOST RPT
2. IBS -> MESSAGE BOARD -> RESEND IBS -> ONPOST RPT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes : This file is identical to CnnnD2LOG.DAT, except for the addition of a file transfer header line. Refer to CnnnD2LOG.DAT for more information.

Structure: ASCII, sequential, variable format, <= 80 bytes/record

Record 1-2: File header and description

```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record N: variable format summary information

Filename: **DA.DMS**

Directory: MAP$DATA$DA:

Purpose: This file is the DMS file that contains the user-defined damage assessment polygons for DAMAGE ASSESSMENT. These polygons are used to define contamination areas and national defense areas.

Access: IBS -> EMERGENCY FUNCTIONS -> DAMAGE ASSESSMENT

Include: IEMIS$INC:RISK_AREA.INC

Structure: DMS format

Filename:	**DLGXDMS.ATT**
Directory:	IEMIS$SYSF:
Purpose:	DLGXDMS.ATT is the attribute lookup table used for converting USGS DLG attribute codes to DMS attribute codes.
Access:	DLGXDMS.ATT is accessed by the DLGXDMS DLG to DMS file conversion utility. The attribute lookup table will also be accessed within IBS when DLG file data is transferred from the onpost system to the offpost system and converted to DMS file format.
Structure:	ASCII, sequential, ≤ 91 bytes/record

Each record contains a field for the DLG attribute major/minor code and fields for the corresponding DMS attribute major/minor/parameter values and attribute description. The DLG major and minor attribute values are combined as a single integer value: (DLGX_DLGATT = 10000*(DLG_MAJOR_CODE) + DLG_MINOR_CODE). If no equivalent DMS attribute value for a DLG attribute is present, the DMS major/minor/parameter values are set to zero. The DLGX_DMSPEN (DMS pen number) and DLGX_DMSLINETYPE (DMS line type) fields (see the following) are currently not used, but are reserved for future use.

Record n: ≤91 byte record (variable)

```
FORMAT (I7,1X,I2,1X,I4,1X,I6,1X,I3,1X,I3,1X,A)
```

DLGX_DLGATT	I*4	Combined DLG major/minor codes
DLGX_DMSMJR	I*4	DMS major attribute
DLGX_DMSMNR	I*4	DMS minor attribute
DLGX_DMSPARM	I*4	DMS parameter attribute
DLGX_DMSPEN	I*4	DMS pen number (not currently used)
DLGX_DMSLINETYPE	I*4	DMS line type (not currently used)
DLGX_DMSATTDESC	C*60	DMS attribute description

Filename: **DOSAGnnn.DAT**

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: Transferred from onpost to offpost. This file contains the data used within IBS to create the graphical D2 track for case nnn.

See Also: CnnnDOSAGE.DAT, OFF$SENDDIR:DOSAGnnn.DAT

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> D2 DOSAGE
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. ONPOSTSIM
4. EVENT

Include: IEMIS$INC:HEADER.INC, IEMIS$INC:WATCHSIM.INC

Notes: This file is identical to CnnnDOSAGE.DAT except for the addition of the file transfer header. The Plume routine creates the plume based on this file and the SLICEnnn.DAT file. When the EVENT program finds this file in ONSITE$DIR, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, variable format, <= 80 bytes/record. There are four control records, up to WS_D2_DC_MAX (10) level description records, followed by up to WS_DOS_MAX (50) records of dosage data, one separator record of zeros, and one record with the final distances for each track.

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record N: See CnnnDOSAGE.DAT for the record formats.

Filename: **DOSAGnnn.DAT**

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file is a copy of CnnnDOSAGE.DAT, with the addition of the transfer header records. It is not used by EMIS.

See Also: CnnnDOSAGE.DAT, ONSITE$DIR:DOSAGnnn.DAT

Access:
1. IBS -> MESSAGE BOARD -> CREATE IBS -> ONPOST RPT
2. IBS -> MESSAGE BOARD -> RESEND IBS -> ONPOST RPT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes: This file is identical to CnnnDOSAGE.DAT except for the addition of the file transfer header.

Structure: ASCII, sequential, variable format, <= 80 bytes/record.

Record 1-2: File header and description
`FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)`

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record N: See CnnnDOSAGE.DAT for the record formats.

Filename: **EI.DMS**

Directory: MAP$DATA$EI:

Purpose: This file is the graphical representation of the environmental samples.

See Also: ENVIRON.DAT

Access:
1. Displayed by most of the graphics programs in IBS
2. EVENT
3. ONPOSTSIM

Notes: The Event program updates the environmental data when a new ENVIRON.DAT file arrives from the post.

Structure: DMS Format

Filename: **ENVIRON.DAT**

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: Transferred from onpost to offpost. This file contains environmental sightings and instrument readings to be displayed as icons by IBS.

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> ENVIRONMENT
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. ONPOSTSIM
4. EVENT

Include: IEMIS$INC:HEADER.INC, IEMIS$INC:WATCHSIM.INC

Notes: The include file reference only provides space for ONPOSTSIM records. It does not define the individual variables. The record definitions which follow correspond to the SIM_RECORD array.

When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file has up to WS_ENV_MAX (10) detail records.

Record 1-2: File header and description
```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K (K=J+2;J=1,N;1<=N<=WS_ENV_MAX):
```
FORMAT (A40,A16,2F12.7)
```

WS_ENV_CSR(J)	C*40	Chemical sensor reading or observation
WS_ENV_DEV(J)	C*16	Device type (HUMAN, MASS SPEC, ACAMS, MINICAMS, or BUBBLER)
WS_ENV_LON(J)	R*8	Longitude
WS_ENV_LAT(J)	R*8	Latitude

Filename: **EVAC_SEARCH_HEADERS.DAT**

Directory: SITEIDYNEV:

Purpose: This file gives evacuation header information not available within the IDYNEV model.

Access: This file is created at the time of IDYNEV execution. The user is prompted for the needed information.

Include: SPOCC$INC:EVH.INC

Notes: For the most part, the header information is used in searches of the scenario table and in picking new IDYNEV cases. If the user does not input the information, the information will not be there for these searches.

Structure: ASCII, direct access, 1325 bytes/record

Record n: `FORMAT (I3.3,1X,F6.2,2X,F5.1,2X,F4.1,2X,A10,1X,A1,2X,I3,256A5)`

EVH_CASE_NUMB	I*4	IDYNEV case number
EVH_LOAD_TIME	R*4	Time to load the network
EVH_PRCT_EVAC	R*4	Percent of population expected evacuated at the end of the run
EVH_PEP_P_VEH	R*4	Average number of people per vehicle
EVH_POP_CLASS	C*10	Type of population (WEEKDAY, WEEKNIGHT, WEEKEND, SEASONAL)
EVH_CASE_TYPE	C*1	Type of case S = Simulation T = Traffic Assignment/Distribution B = Both
EVH_NUM_ZONES	I*4	Number of evacuation zones (0..EVH_MAX_ZONES)
EVH_EVAC_ZONZ(I)	C*5	Evacuation zones (EPZs)

Filename: EVENT.LOG

Directory: INFO$DIR:

Purpose: EVENT program activity log. Only found in INFOMANAGER.

Access:
1. IBS -> STATUS BOARD -> VIEW BATCH LOG
2. EVENT

Structure: ASCII, sequential, variable length.

Filename: FACILITY.DAT

Directory: OFF$DIR:

Purpose: This file contains information regarding facilities.

Access:
1. IBS -> RESOURCE MANAGEMENT
2. RMMGR

This file is used in conjunction with RESOURCE MANAGEMENT, which can create and update it. The standalone utility RMMGR can do the same.

Include: IEMIS$INC:FAC.INC

Notes: Routine M_FAC_READ reads the file. The parameter FAC_MAX limits the # of records in the file.

Structure: ASCII, direct access, 256 bytes/record, shared and locked. This file contains *n* records, each with information for a single facility.

Record n: 256 bytes

```
FORMAT (A40,A25,A40,A20,A2,A10,A20,A12,A1,A13,I4,L1,A20,A5,A20,
23X)
```

This record contains information for a single facility.

FAC_NAME	C*40	Facility name (also a known point name)
FAC_TYPE	C*25	Facility type (references an item in the icon control file)
FAC_ADDR	C*40	Address
FAC_CITY	C*20	City
FAC_STATE	C*2	State abbreviation
FAC_ZIP	C*10	ZIP code
FAC_CNAMEL	C*20	Last name of contact person
FAC_CNAMEF	C*12	First name of contact person
FAC_CNAMEM	C*1	Middle initial of contact person
FAC_PHONE	C*13	Office phone # for contact person
FAC_CAP	I*4	Capacity: # people facility will hold
FAC_MA	L*4	Does a mutual aid agreement exist?
FAC_MA_NUM	C*20	Mutual aid agreement #
FAC_AG_CODE	C*5	Agency of contact person
FAC_POSISH	C*20	Position (job title) of contact person

Filename: FACILITY_TRACK.DAT

Directory: OFF$DIR:

Purpose: This file contains tracking/load information for facilities.

Access:
1. IBS -> RESOURCE MANAGEMENT
2. RMMGR
3. LOCATE

This file is used in conjunction with RESOURCE MANAGEMENT and LOCATE, both of which can update it, although LOCATE cannot create it. The standalone utility RMMGR can also create and update it.

Include: IEMIS$INC:FT.INC

Notes: Routine M_FT_READ reads the file. The parameter FT_MAX limits the number of records in the file.

Structure: ASCII, direct access, 64 bytes/record, shared and locked. This file contains *n* records, each with information for a single facility.

Record n: 64 bytes
`FORMAT (A40,A1,3I6)`

This record contains information for a single facility.

FT_NAME	C*40	Facility name (also a known point name)
FT_ACTIVE	C*1	Is facility active? (Y,N)
FT_REC	I*4	# people who have come to the facility
FT_REM	I*4	# people still at the facility
FT_REL	I*4	# people released from the facility

Filename: FTP.COM_*

Directory: IEMIS$SYSF:

Purpose: Two different commercial software packages can be used to communicate with the onpost system using File Transfer Protocol (FTP): Pathway Access, formerly called Win/TCP, and TCP ware. Each uses a slightly different syntax, as specified in FTP.COM_WINTCP and FTP.COM_TCPWARE, respectively. The appropriate file should be copied to FTP.COM_TEMPLATE, where IBS can access it.

Access:
1. EVENT
2. IBS -> MESSAGE BOARD -> CREATE IBS -> ONPOST REPORT

Notes: The database administrator should copy the appropriate format (FTP.COM_TCPWARE or FTP.COM_WINTCP) into FTP.COM_TEMPLATE.

Structure: ASCII, sequential, variable length, 80 bytes/record.

Filename: ICON.CTL

Directory: MAP$ICONS:

Purpose: This is the icon control file for a map database in IBS. It contains the list of available icons and the names of the files that are needed to plot each icon.

See Also: *.DMS [MAP$ICONS:]

Access: This file is accessed during the initialization phase of most GIS applications in IBS.

Include: MAP$DEVELOP$INC:MAPROOT.INC

Notes: MAP$ICON$CTL is the logical for ICON.CTL.

Structure: ASCII, sequential, 78 bytes/record. This file contains *n* records, each with the attribute, icon description, and file name for each of the IBS icons.

Record n:

```
FORMAT (I2,I5,1X,A25,A40,A1,I4)
```

TT_MAJOR	I*4	Major attribute code for icon
TT_MINOR	I*4	Minor attribute code for icon
CON_DESC	C*25	Icon description
CON_FILE	C*40	Path and file name for icon
CON_FR_TYP	C*1	Icon facility/resource type: 'F' = FACILITY icon, 'R' = RESOURCE icon, ' ' = NON-FACILITY/RESOURCE icon
CON_FR_SPOT	I*4	Indicates the order in which resources should be plotted around a facility, clockwise starting at 12 o'clock, in concentric circles--6 icons in the 1st, or inner circle, 12 in the 2nd, 18 in the 3rd, and so on.

Filename: IGL_*.DAT

Directory: IEMIS$SYSF:

Purpose: Contains a list of igloos, and their corresponding latitude/longitude locations. This is used to find source locations for D2 cases. There is one file for each of the eight depots.

Access: IBS -> EMERGENCY ACTIVITIES -> HAZARD ANALYSIS -> EXECUTE D2 -> Modify D2 Case Input

Include: IEMIS$INC:D2_IGL.INC

Structure: ASCII, sequential, variable length. This file contains one header record, followed by a list of igloo records.

Record 1:

FORMAT (1X,A4,1X,A3,1X,A5,1X,A23)

D2_IGL_SITE	C*4	IBS site name (i.e. UMDA)
D2_IGL_D2_SITE	C*3	D2PC site name (i.e. UAD)
D2_IGL_VER	C*5	Version of igloo file
D2_IGL_DT	C*23	Date of file creation

Record 2-n:

FORMAT (F9.6,1X,F11.6,1X,A12,A46)

D2_IGL_LAT()	R*8	Latitude of igloo
D2_IGL_LON()	R*8	Longitude of igloo
D2_IGL_CODE()	C*12	Code name of igloo
D2_IGL_DESC()	C*46	Description of igloo

Filename: IMPMAIL.LOG

Directory: OFF$DIR:

Purpose: This file contains important mail selected from VAX Mail for use in reports. It is created by extracting messages within mail (MAIL > EXTRACT n OFF$DIR:IMPMAIL.LOG). Then when a log report is created, messages in this file which meet the search criteria are put into the log report.

Access: 1. IBS -> MESSAGE BOARD -> CREATE COMMON/PRIVATE LOG REPORT

Structure: ASCII, sequential, variable record length

Record n: 80 bytes

```
FORMAT (A)
```

RECORD	C*80	Line from VAX Mail message

Filename: JOB_ENV.DAT

Directory: SYS$LOGIN:

Purpose: Contains parameters that determine the user's operating environment.

Access: Virtually all the programs within the IBS system read it. IBS -> SETUP -> JOB ENVIRONMENT is the main module for changing the values in the file, but many other IBS function do also, especially changing sites, cases, or map layers.

Include:
INCLUDE 'IEMIS$INC:JOB_ENV.INC'
INCLUDE 'IEMIS$INC:JEFMT.INC'

Structure: ASCII, sequential, variable length, 80 bytes/record.

Record 1: 80 bytes, Header
FORMAT(A50,1X,A23,1X,A5)

JE_DESC	C*50	File description
JE_CDTS	C*23	Date file was created
JE_VERS	C*5	Format version (V2.00)

Record 2: 73 bytes, IBS SITE
FORMAT(10X,3(5X,A15,1X))

JE_IBS_DISK	C*15	Disk name (disk or OPERATIONAL)
JE_IBS_USER	C*15	User name (user or OPERATIONAL)
JE_IBS_SITE	C*15	Site name (site or OPERATIONAL)

Record 3: 52 bytes, D2 VARS
FORMAT(10X,6(4X,I2,1X))

JE_DEF_TMP_UNITS	I*4	D2: Temperature units
JE_DEF_VEL_UNITS	I*4	D2: Wind Speed units
JE_DEF_QQQ_UNITS	I*4	D2: Quantity units
JE_DEF_HML_UNITS	I*4	D2: Ht of Mixing Layer units
JE_DEF_PMM_UNITS	I*4	D2: PMM units
JE_DEF_TIM_UNITS	I*4	D2: Time units

Record 4: 53 bytes, COLORS
FORMAT(10X,11X,I2,1X,12X,I2,1X,12X,I2)

JE_MENU_COLOR	I*4	Dflt color of graphical menu items
JE_ALPHA_BACKG	I*4	Dflt color of alpha background
JE_ALPHA_FOREG	I*4	Dflt color of alpha foreground

Record 5: 57 bytes, GRID

```
FORMAT(10X,5X,I1,1X,10X,I1,1X,11X,I2,1X,12X,I2)
```

JE_GRID_TYPE	I*4	Dflt grid type (1=LL_D,2=LL_DMS)
JE_GRID_TYPE_LINE	I*4	Dflt type of grid lines (0-7)
JE_GRID_COLOR_LINE	I*4	Dflt color of grid lines (0-15)
JE_GRID_COLOR_LABEL	I*4	Dflt color of grid labels (0-15)

Record 6: 44 bytes, MISC1

```
FORMAT(10X,6X,I1,1X,6X,I1,1X,8X,I1,1X,6X,I2)
```

JE_DEF_GPICK	I*4	Dflt graphical selection method
JE_DEF_APICK	I*4	Dflt alpha selection method
JE_DEF_RPTDEST	I*4	Dflt report destination
JE_DEF_UNITS	I*4	Dflt units

Record 7: 58 bytes, MISC2

```
FORMAT(10X,9X,A4,1X,9X,A4,1X,10X,A10)
```

JE_OLD_SITE	C*4	Old site name
JE_POP_CODE	C*4	Population code
JE_TERM_TYPE	C*10	Terminal Type (TEKTRONIX, TEKPCTGRAF)

Record 8: 56 bytes, MODELCTL

```
FORMAT(10X,10X,I2,1X,11X,I2,1X,11X,A8)
```

JE_MODEL_NUM	I*4	# of models (6)
JE_MODEL_CURR	I*4	Current model # (1-6)
JE_MODEL_NAME	C*8	Current model name

Record 9 - JE_MODEL_NUM+8: 28 bytes, MODEL

```
FORMAT(10X,A4,1X,A4,1X,A8)
```

JE_MODEL_CCASE_PLAN(I)	C*4	Current model case
JE_MODEL_BCASE_PLAN(I)	C*4	Base model case
JE_MODEL_NAMES(I)	C*8	Model name

Record JE_MODEL_NUM+9: 14 bytes, NTOPO

```
FORMAT(10X,I4)
```

JE_TOPO_NUM	I*4	# of topo records

Record JE_MODEL_NUM+10 - JE_MODEL_NUM+JE_TOPO_NUM+9: 73 bytes, TOPO

```
FORMAT(10X,A20,1X,A16,2F8.2)
```

JE_TOPO_CAT(I)	C*20	Topo class
JE_TOPO_LOG(I)	C*16	Topo logical
JE_TOPO_MNX(I)	R*4	Min distance (miles)
JE_TOPO_MXX(I)	R*4	Max distance (miles)

Record JE_MODEL_NUM+JE_TOPO_NUM+10: 14 bytes, NASSGNED

```
FORMAT(10X,I4)
```

JE_ASS_NUM	I*4	# of case assignment records

Record JE_MODEL_NUM+JE_TOPO_NUM+11 -
JE_MODEL_NUM+JE_TOPO_NUM+JE_ASS_NUM+10: 64 bytes, ASSIGNED

```
FORMAT(10X,5X,A15,1X,5X,A4,1X,6X,A8,1X,5X,I3)
```

JE_ASS_DISK(I)	C*15	Disk for assigned case
JE_ASS_SITE(I)	C*4	Site for assigned case
JE_ASS_MODL(I)	C*8	Model for assigned case
JE_ASS_CASE(I)	I*4	# for assigned case

Filename: KEY.DAT

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:, OFF$SENDDIR:

Purpose: Sent from onpost to offpost or from offpost to onpost. These files will contain keywords that will inform the remote system of the purpose of the data transmitted, or request that the system take a particular action..

Access:
1. IBS -> MESSAGE BOARD -> CREATE IBS TO ONPOST REPORT
2. IBS -> MESSAGE BOARD -> RESEND IBS TO ONPOST REPORT
3. IBS -> MESSAGE BOARD -> VIEW IBS TO ONPOST REPORT
4. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT
5. ONPOSTSIM
6. EVENT

Include: IEMIS$INC:HEADER.INC

Notes: When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file contains up to KEY_MAX (10) sections of up to KEY_DESC_MAX (100) lines, each preceded by a "key" which determines how the records in that section are to be interpreted. The following keys have been defined:

DATA_OK	File was processed OK (ACK)
DATA_BAD	Data in file was bad
FILE_BAD	File was corrupt or unreadable
FILE_UNKNOWN	Unknown file (invalid report type)
FILE_MISSING	Missing file (referenced in XFERLIST)
PLEASE_ECHO	Are you there? (PING)
IS_AN_ECHO	Im here (ACK)

Record 1-2: File header and description

```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K1 (K1=N+2; 1<=N<=KEY_MAX):

FORMAT (I4,1X,A32)

KEY_NREC	I*4	# of records accompanying KEY
KEY_CODE	C*32	KEY which determines how accompanying records are to be interpreted

Record K2 (K2=J+K1; J=1,KEY_NREC; 0<=KEY_NREC<=KEY_DESC_MAX):

FORMAT (A80)

KEY_DESC(J)	C*80	Contents

Definition for DATA_OK, DATA_BAD, FILE_BAD, FILE_UNKNOWN, FILE_MISSING:

KEY_DESC(1) = filename
KEY_DESC(2,3) = Header of file (if possible)
KEY_DESC(4) = Date
KEY_DESC(5+) = Error message, description

Definition for IS_AN_ECHO:

KEY_DESC(1) = Date/Time

Filename: KNOWN_POINTS.DAT

Directory: OFF$DIR:

Purpose: This file contains information regarding known points.

Access:
1. IBS -> RESOURCE MANAGEMENT
2. RMMGR
3. Other IBS functions

This file is used in conjunction with RESOURCE MANAGEMENT, which can create and update it. The stand-alone utility RMMGR can do the same.

Include: IEMIS$INC:KP.INC

Notes: Routine M_KP_READ reads the file. The parameter KP_MAX limits the number of records in the file.

Structure: ASCII, direct access, 80 bytes/record, shared and locked. This file contains *n* records, each with information for a single known point.

Record n: 80 bytes
```
FORMAT (A40,I4,6X,2F15.10)
```

This record contains information for a single known point.

KP_NAME	C*40	Known point name (location)
KP_ID	I*4	Unique id, used as a link to DMS attributes
KP_LON	R*8	Longitude
KP_LAT	R*8	Latitude

Filename: KNOWNPTS.DAT

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: Transferred from onpost to offpost. This file contains known points -- locations with known coordinates. It will primarily contain tower locations, but any other known points could also be in it.

See Also: KNOWN_POINTS.DAT, OFF_KP.DAT

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> KNOWN POINTS
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. ONPOSTSIM
4. EVENT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes: When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, fixed format, 80 bytes/record. This files contains up to WS_KP_MAX (1000) records, each with information for a single known point.

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K (K=J+2;J=1,N;1<=N<=WS_KP_MAX):
FORMAT (A40,1X,I6,1X,I2,1X,I4,1X,2F12.7)

WS_KP_NAME(J)	C*40	Name of known point. ("TOWER:", 1X,A8 is format for Weather Tower known points, where the A8 is the 8 character Tower ID.
WS_KP_ID(J)	I*4	Unique ID for known point
WS_KP_MAJOR(J)	I*4	Attribute major code

WS_KP_MINOR(J)	I*4	Attribute minor code
WS_KP_LON(J)	R*8	Longitude
WS_KP_LAT(J)	R*8	Latitude

Filename: KNOWN_POLYGONS.DAT

Directory: OFF$DIR:

Purpose: This file contains the names and ids of the IBS known polygons. The known polygons reference specific types of polygons used by the IBS for specific functions. The known polygons include EPZs and user-defined polygons (risk area, search/rescue, contamination, and national defense area polygons).

Access:
1. IBS -> EMERGENCY ACTIVITIES -> DIRECTION CONTROL -> RISK AREA ANALYSIS
2. IBS -> EMERGENCY ACTIVITIES -> DAMAGE ASSESSMENT
3. IBS -> EMERGENCY ACTIVITIES -> SEARCH/RESCUE

The file is accessed within RISK AREA ANALYSIS when adding/deleting EPZs to the risk area or when defining a risk polygon. It is also accessed within SEARCH/RESCUE when creating user-defined search/rescue polygons and DAMAGE ASSESSMENT when creating contamination and national defense area polygons.

Include: IEMIS$INC:RISK_AREA.INC

Structure: ASCII, direct access, 64 bytes/records, shared and locked. Contains one header record and *n* known polygon records.

Record 0: Header

```
FORMAT (I4,1X,A23,36X)
```

KPL_LAST	I*4	Last used polygon ID
KPL_CDTS	C*23	Date last polygon was created

Record n: Known polygon

```
FORMAT (I4,1X,A40,19X)
```

KPL_ID	I*4	Unique known polygon ID
KPL_NAME	C*40	Known polygon name

Filename: KNOWN_ROUTES.DAT

Directory: OFF$DIR:

Purpose: This file associates the known route codes with known route names and the DMS files in which they are stored. Known routes are generally evacuation routes but may also be any proposed set of transportation routes for any purpose.

Access:
1. IBS -> EMERGENCY ACTIVITIES -> EVACUATION
2. IBS -> MESSAGE BOARD -> IBS -> ONPOST REPORTS

Include: SPOCC$INC:KR.INC

Structure: ASCII, sequential, variable length records. One record is required for each known route on the system.

Record 1-N:

```
FORMAT (I4, A40)
```

KR_ID	I*4	Mono-incremental identification code
KR_NAME	C*40	Known name of route

Filename: LABEL.DAT

Directory: IEMIS$CHEMS:

Purpose: Provide a labels list for properties to be added to the property database by batch program.

Access: Current version of the system does not appear to reference this file.

Structure: ASCII, Sequential.

Record 1,N: 68 bytes
FORMAT (A3,5X,A60)

NAME(I)	C*3	Chemical code.
TEMP(I)	C*60	Property of NAME(I).

Filename: LIMITxx.BIN

Directory: MAP$DATA$xx:

Purpose: Identifies the DMS files that comprise the topography, and their geographic limits. In the case of D2, EX, FI, VA, and WD, where the DMS files represent multiple model scenarios, the LIMITxx.BIN identifies the current scenario. The SEELIMITS utility converts the LIMITxx.BIN into ASCII format.

See Also: LIMITxx.DAT

Access: Accessed by all of the graphics programs that access topographies.

Structure: Binary, sequential, fixed length, 48 bytes/record.

Record N: UNFORMATTED

FN	C*16	DMS file name
XMIN	R*8	Minimum Longitude
XMAX	R*8	Maximum Longitude
YMIN	R*8	Minimum Latitude
YMAX	R*8	Maximum Latitude

Filename: **LOGIN.COM**

Directory: SYS$LOGIN:

Purpose: This file is a command procedure for initializing a user's environment.

See also: LOGIN.COM_TEMPLATE

Access:
1. NEWUSER
2. SYS$MANAGER:SYLOGIN.COM

The NEWUSER program creates the file by copying the template file, IEMIS$SYSF:LOGIN.COM_TEMPLATE. This file is executed during login. The user may edit this file to modify the default configuration, subject only to the limits of permissions and VMS knowledge, keeping in mind that IBS may no longer work properly.

Structure: DCL command

Filename: LOGIN.COM_TEMPLATE

Directory: IEMIS$SYSF:

Purpose: This file is merely a template command procedure used when creating new user accounts.

See Also: LOGIN.COM

Access: NEWUSER copies this file to the new user's login directory and renames it to LOGIN.COM.

Notes: The system manager/DBA should configure this file for each site, keeping in mind that IBS may not work properly if the file is changed. For example, the template file has a command that sets the user's terminal type to VT100. Other terminal types will adversely affect data entry from the IBS forms package.

Structure: DCL command

Filename: MAIL_SYSTEMS.DAT

Directory: IEMIS$SYSF:

Purpose: This file contains the node, directory name, and password of all other offpost systems to which mail will be routed. It allows the EVENT program to keep MAILLIST.DAT on the other systems current. EVENT will route MLUPDATE.DAT to each of the systems listed. Unlike OTHER_SYSTEMS.DAT, which is empty on all offpost systems except the one that communicates directly with the onpost system, MAIL_SYSTEMS.DAT should be the same on all offpost systems. EVENT will recognize the name of the system it is on, and will not attempt to update it.

See Also: MAILLIST.DAT, MLUPDATE.DAT, OTHER_SYSTEMS.DAT, POST_SYSTEM.DAT

Access: 1. EVENT

Notes: This file must be created and maintained by the database administrator using a text editor. Because it contains passwords for user accounts, it must be protected from prying eyes. The file protections are (S:RWED,O:RWED,G,W).

If the password (or user name) changes on a destination node, this file must be updated. If the password (or user name) on your system changes, the DBAs for each of the destination node must be informed so they can update their files. If data is not arriving at a destination node, the first thing to check is that the user name and password are correct. Log on to the destination node using either FTP or RLOGIN.

Structure: ASCII, sequential, 80 bytes/record. This file contains *n* records with the node, user name, password, and directory names of other offpost systems.

Record n: 80 bytes
FORMAT (A80)

NODE	C*80	Node, user name, and password in the form of: node"username password"::EVNDB:[ONSITE]

Filename: MAILLIST.DAT

Directory: IEMIS$SYSF:

Purpose: This file is the list of all users to whom mail can be sent using VMS mail. It is used within IBS to create distribution lists. It is maintained between all of the organizations at a site by EVENT.

See Also: MAIL_SYSTEMS.DAT, MLUPDATE.DAT

Access:
1. IBS -> MESSAGE BOARD (various mail functions; read-only)
2. MLMGR (used to update the master mail list)
3. MAIL_LIST
4. EVENT

Include: IEMIS$INC:ML.INC, SPOCC$INC:MAIL_LIST.INC

Structure: ASCII, sequential, variable length, 80 bytes/record

Record 1: FORMAT(A)

VERSION	C*5	Mail list file version number

Record N,N+1: A pair of records for each user
FORMAT (A20,1X,A12,1X,A1,1X,A5,1X,A20,/,A80)

ML_NAME_L	C*20	Last name of the person to send mail to
ML_NAME_F	C*12	First name of person
ML_NAME_M	C*1	Middle initial of person
ML_AGENCY	C*5	Agency for which person works
ML_POSISH	C*20	Position at agency
ML_ADDRESS	C*80	Mail address

Filename:	MASTERDIR.DAT
Directory:	IEMIS$SYSF:
Purpose:	This file is used to maintain information needed to provide online help to users of COPYCASESITE and related functions.
Access:	1. FIXMSTRDIR 2. IBS -> SETUP -> JOB ENVIRONMENT -> COPY CASE SITE 3. NEWSITE
Include:	IEMIS$INC:LOG_QUAL.INC
Notes:	This file is created using program FIXMSTRDIR. A set of six records is required for each site identified in file IEMIS$SYSF:ALLSITE.DAT.
Structure:	ASCII, sequential, variable length, 80 bytes/record

Record 1:

```
FORMAT (T6,A6,T18,A12,T36,A4,T46,A15)
```

RNODE(I)	C*6	Node name on which site is located
RUSER(I)	C*12	User directory containing site
RSITE(I)	C*4	Site code
RDISK(I)	C*6	Disk device on which site is located

Record 2:

```
FORMAT (10A8)
```

MODELS(J)	C*8	Models directories for specified site

Record 3:

```
FORMAT (20A4)
```

POPS(J)	C*4	Population data set codes for specified site

Record 4-6:

```
FORMAT (40A2)
```

TOPOS(J)	C*2	Topography directory codes for specified site. A total of 120 topographies are allowed (40 on each of 3 records).

Filename: MEM_LOG.DAT

Directory: MAP$MEMLOG:

Purpose: This file contains summary information saved by certain programs detailing totals and averages relating to GIS objects.

Access: Some programs automatically calculate summary information on various types of GIS objects and save it to the file.

Notes: Some of the more important programs do NOT summarize and report.

Structure: ASCII, sequential, 132 bytes/record. This file contains 3 header records followed by *n* summary records. Each time a program writes to the file, it adds a single summary record.

Record 1-3: 132 bytes, header

```
FORMAT (A132)
```

The header records will not be documented.

Record n: 132 bytes, Summary

```
FORMAT (1X,A23,1X,A12,1X,A12,10I8,'  ')
```

This record contains summary information.

CDTS	C*23	Date/time record added to file
USER	C*12	Name of user running program
TUTILITY	C*12	Name of program doing summary
NUMFILE	I*4	# of GIS files read
NUM_S	I*4	# of arcs
NUM_D	I*4	# of points
NUM_A	I*4	# of attributes
NUM_L	I*4	# of lines
NUM_P	I*4	# of polygons
AVG_PTS	I*4	Avg # of pts/arc
MAX_PTS	I*4	Max # of pts/arc
AVG_ATT	I*4	Avg # of attributes/object
MAX_ATT	I*4	Max # of attributes/object

Filename: MLUPDATE.DAT

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:, OFF$SENDDIR:

Purpose: Onpost < = > Offpost. This file tells the receiving system what updates need to be made to the Mail List. The contents of the file will instruct the receiving system to add, modify, or delete any number of user records.

See Also: MAILLIST.DAT, MAIL_SYSTEM.DAT

Access:
1. IBS -> MESSAGE BOARD -> CREATE IBS TO ONPOST REPORT
2. IBS -> MESSAGE BOARD -> RESEND IBS TO ONPOST REPORT
3. IBS -> MESSAGE BOARD -> VIEW IBS TO ONPOST REPORT
4. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT
5. EVENT

Include: IEMIS$INC:HEADER.INC

Notes: When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, variable format, < = 80 bytes/record. Each section in this file contains the information necessary to update the mail list for a single user.

Record 1-2: File header and description

```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K1:

```
FORMAT (A3)
```

MLU_OPERATION	C*3	"ADD", "MOD", or "DEL"

Record K2-K3: New data (blank if DELeting)

```
FORMAT (A20,1X,A12,1X,A1,1X,A5,1X,A20,/,A80)
```

MLU_N_NAME_L	C*20	New last name of person to Add/Modify
MLU_N_NAME_F	C*12	New first name of person to Add/Modify
MLU_N_NAME_M	C*1	New middle initial of person to Add/Modify
MLU_N_AGENCY	C*5	New agency of person to Add/Modify
MLU_N_POSISH	C*20	New position of person to Add/Modify
MLU_N_ADDRSS	C*80	New e-mail address of person to Add/Modify

Record K4-K5: User to Delete or Modify (blank if ADDing)

```
FORMAT (A20,1X,A12,1X,A1,1X,A5,1X,A20,/,A80)
```

MLU_O_NAME_L	C*20	Last name of person to Modify/Delete
MLU_O_NAME_F	C*12	First name of person to Modify/Delete
MLU_O_NAME_M	C*1	Middle initial of person to Modify/Delete
MLU_O_AGENCY	C*5	Agency of person to Modify/Delete
MLU_O_POSISH	C*20	Position of person to Modify/Delete
MLU_O_ADDRSS	C*80	E-mail address of person to Modify/Delete

Filename: NODE_POS.DAT

Directory: SITEODYNEV:

Purpose: This file contains position data and link identification information of nodes extracted from Tiger Trails data by utility TTXTTDAT.

See Also: NODE_POS.SRT

Access:
1. IEMIS -> MODEL LIBRARY -> GENERATE EVACUATION NETWORK -> EXTRACT LINKS FROM TIGER
2. REA_GR -> GENERATE A FIRST-CUT EVACUATION CASE -> EXTRACT LINKS FROM TIGER

Notes: A similar file, NODE_POS.SRT, is sorted on location.

Structure: ASCII, sequential, fixed format, 43 bytes/record
This file has one record for each string endpoint extracted from Tiger Trails data.

Record 1-n:

FORMAT (2(1X,F15.10),1X,I8,I2)

XD	R*8	Longitude of sting end point.
YD	R*8	Latitude of sting end point.
JLINK		I*4Index of string from which point taken.
END		I*41 or 2 for end one or two of link.

Filename: NODE_POS.SRT

Directory: SITEODYNEV:

Purpose: This file contains position data and link identification information of nodes extracted from Tiger Trails data by utility TTXTTDAT. It is sorted using the entire record as the key in ascending order.

See Also: NODE_POS.DAT

Access:
1. IEMIS -> MODEL LIBRARY -> GENERATE EVACUATION NETWORK -> EXTRACT LINKS FROM TIGER
2. REA_GR -> GENERATE A FIRST-CUT EVACUATION CASE -> EXTRACT LINKS FROM TIGER

Notes: A similar file, NODE_POS.DAT, is unsorted.

Structure: ASCII, sequential, fixed format, 43 bytes/record
This file has one record for each string endpoint extracted from Tiger Trails data.

Record 1-n:

FORMAT (2(1X,F15.10),1X,I8,I2)

XD	R*8	Longitude of sting end point.
YD	R*8	Latitude of sting end point.
JLINK		I*4Index of string from which point taken.
END		I*41 or 2 for end one or two of link.

Filename: NOTIFY.DAT

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: Transferred from onpost to offpost. This file provides notification of a chemical event at the depot. After an event has occurred and the offpost has been notified, any future notifications will be used to upgrade or downgrade the event. The actual event time will not change, however. The onpost event ID will be the same for related events, but must be different for an unrelated event, not that the offpost system will be able to handle two events at one time.

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> NOTIFICATION
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. IBS -> STATUS BOARD -> ONPOST EVENT STATUS
4. ONPOSTSIM
5. EVENT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes: When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file has 8 records of miscellaneous information followed by up to WS_NOT_MAX (10) records describing the event.

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record 3-10:

```
FORMAT ('Current Date/Time',T33,':',1X,A23/
'Event   Date/Time',T33,':',1X,A23/
'Classification',T33,':',1X,A20/
'Onpost Event ID',T33,':',1X,I5/
'Onpost Event Local ID',T33,':',1X,A12/
'Onpost Essential Population',T33,':',1X,I4/
'Onpost Non-Essential Population',T33,':',  1X,I4/
'Onpost Team Population',T33,':',1X,I4)
```

WS_NOT_DT_1	C*23	Current time (dd-mmm-yyyy hh:mm:ss.hh)
WS_NOT_DT_2	C*23	Event time (dd-mmm-yyyy hh:mm:ss.hh)
WS_NOT_CLASS	C*20	Event classification (COMMUNITY EMERGENCY, POST ONLY EMERGENCY, LIMITED EMERGENCY, NON-SURETY EMERGENCY)
WS_NOT_EV_ID	I*4	Unique onpost event ID
WS_NOT_MCE	C*12	Local ID
WS_NOT_E_POP	I*4	Onpost essential population
WS_NOT_N_POP	I*4	Onpost non-essential population
WS_NOT_T_POP	I*4	Onpost team population

Record K (K=J+10; J=1,WS_NOT_MAX):

```
FORMAT (A80)
```

WS_NOT_DESC(J)	C*80	Description of the event, which should probably include the PAR

Filename: NUC.DAT

Directory: IEMIS$MESORAD:

Purpose: Provides data on the characteristics of radionuclides which may be included in MESORAD source terms.

Access: 1. MES_(NO)GR

Include: MESO$INC:NUC.INC

Structure: ASCII, Sequential. The first 8 records are a header. The next 7 record types are repeated for each radionuclide. This is followed by an end of data flag record which may be followed by one or more records containing notes.

Record 1: 80 bytes, header
`FORMAT (A)`

LINE	C*80	Nuclide data file identification header.

Record 2,10: variable length, header

```
FORMAT ('F 3 nuclide halflife    parent       %frompar'//
'     daughter     depo.vel.  ',/,
T13,'gammal       yieldl       gamma2       yield2',/,
T13,'gamma3       yield3       gamma4       yield4',/,
T13,'gamma5       yield5',/,
T13,'lung     wbinhal   thyroid   wbground sicdrf',/,
T13,'ING_WB   ING_THY   BIVEG     BIMEAT   FI MILK   FI MEAT',/,
T13,'BETA')
```

literal	C*(*)	Format literal only.

Record K (K = 4 + N*7): 65 bytes (N ranges from 1 to number of radionuclides defined in the file.)

`FORMAT ('T 3 ',A7,T13,E12.0,T26,A7,T39,E12.0,T52,A7,T65,A1)`

NUCLIDE(I)	C*7	Name of nuclide being defined.
HALFLIFE(I)	R*4	Halflife (days) of nuclide being defined.
PARENTNAME(I)	C*7	Name of nuclide parent.
PARENTYIELD(I)	R*4	% daughter produced from parent.
DAUGHTERNAME(I)	C*7	Name of nuclide daughter.
DEPFLG(I)	C*1	If "T", nuclide will deposit on ground.

Record K2 (K2 = 5 + N*7): 63 bytes (N ranges from 1 to number of radionuclides defined in the file.)

```
FORMAT (11X, 4(1X,E12.0))
```

ENERGY(1,I)	R*4	1st Gamma energy of nuclide being defined.
FIELD(1,I)	R*4	1st yield of nuclide being defined.
ENERGY(2,I)	R*4	2nd Gamma energy of nuclide being defined.
FIELD(2,I)	R*4	2nd yield of nuclide being defined.

Record K3 (K3 = 6 + N*7): 63 bytes (N ranges from 1 to number of radionuclides defined in the file.)

```
FORMAT (11X, 4(1X,E12.0))
```

ENERGY(3,I)	R*4	3rd Gamma energy of nuclide being defined.
FIELD(3,I)	R*4	3rd yield of nuclide being defined.
ENERGY(4,I)	R*4	4th Gamma energy of nuclide being defined.
FIELD(4,I)	R*4	4th yield of nuclide being defined.

Record K4 (K4 = 7 + N*7): 37 bytes (N ranges from 1 to number of radionuclides defined in the file.)

```
FORMAT (11X, 2(1X,E12.0)
```

ENERGY(5,I)	R*4	5th Gamma energy of nuclide being defined.
FIELD(5,I)	R*4	5th yield of nuclide being defined.

Record K5 (K5 = 8 + N*7): 70 bytes (N ranges from 1 to number of radionuclides defined in the file.)

```
FORMAT (10X, 6(2X,G8.0))
```

LPIDF(I)	R*4	Lung dose factor for nuclide being defined.
WBIDF(I)	R*4	Whole body inhalation dose factor for nuclide being defined.
TPIDF(I)	R*4	Thyroid dose factor for nuclide being defined.
WBGCDF(I)	R*4	Whole body ground dose factor for nuclide being defined.
SICDF(I)	R*4	Semi-infinite cloud dose factor for nuclide being defined.
BONEDF(I)	R*4	Bone dose factor for nuclide being defined.

Record K6 (K6 = 9 + N*7): 70 bytes (N ranges from 1 to number of radionuclides defined in the file)

```
FORMAT (10X, 6(2X,G8.0))
```

ING_DOSE_FACT_WB(I)	R*4	Whole body ingestion dose factor for nuclide being defined.
ING_DOSE_FACT_THY(I)	R*4	Thyroid ingestion dose factor for nuclide being defined.
B(I)	R*4	Concentration ratio for plant uptake of nuclide being defined from soil (pCi/kg wet mass of plant per pCi/kg dry soil).
BA(I)	R*4	Concentration ratio for pasture plant uptake of nuclide being defined from soil (pCi/kg dry mass of plant per pCi/kg dry soil).
FM(I)	R*4	Transfer coefficient of nuclide being defined from daily intake of forage by the cow to milk (pCi/L per pCi/day).
FF(I)	R*4	Transfer coefficient of nuclide being defined from daily intake of forage by beef cattle to meat (pCi/kg per pCi/day).

Record K7 (K7 = 10 + N*7): 20 bytes (N ranges from 1 to number of radionuclides defined in the file)

```
FORMAT (12X, G8.0)
```

BETADF(I)	R*4	Beta dose factor for nuclide being defined.

Record K8 (K8 = 11 + N*7): 11 bytes, end of data flag (N is the number of radionuclides defined in the file)

```
FORMAT ('END OF DATA')
```

literal	C*11	Format literal only.

Record K9 (K9 = K8 + 1,N2): 80 bytes (N2 is the number of note records following data definition.)

```
FORMAT (A)
```

NOTE(NX)	C*80	Notes on data derivation.

Filename: NUC_CHERNOBYL.DAT

Directory: IEMIS$MESORAD:

Purpose: Provides data on the characteristics of radionuclides that may be included in MESORAD source terms. This file supported studies of Chernobyl accident.

See Also: NUC.DAT

Access: 1. MES_(NO)GR

Include: MESO$INC:NUC.INC

Notes: This file can be copied to IEMIS$SYSF:NUC.DAT for studies of accidents requiring the Chernobyl data. Remember that all users of the system will be using whatever data is placed in NUC.DAT.

Structure: ASCII, Sequential.
It is identical to that of NUC.DAT.

Filename: NUC_FEMA.DAT

Directory: IEMIS$MESORAD:

Purpose: Provides data on the characteristics of radionuclides that may be included in MESORAD source terms. This file is a backup file for the standard FEMA NUC.DAT file.

See Also: NUC.DAT

Access: 1. MES_(NO)GR

Include: MESO$INC:NUC.INC

Notes: This file can be copied to IEMIS$SYSF:NUC.DAT to restore the standard set of FEMA nuclide definitions for MESORAD. Remember, all users of the system will be using whatever data is placed in NUC.DAT.

Structure: ASCII, Sequential.
It is identical to that of NUC.DAT.

Filename: OFF_CAS.DAT

Directory: OFF$SENDDIR:

Purpose: This file is sent from offpost to onpost, and provides a summary of offpost casualties.

See Also: CAS_SUM.DAT

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> OFFPOST CASUALTY REPORT
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:OFFRPT.INC

Notes: A similar file, CAS_SUM.DAT, is sent from the onpost to the IBS but it differs slightly in format and content.

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file has 9 records of miscellaneous information followed by up to OS_CS_MAX (5) records describing the event.

Record 1-2: File header and description

```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record 3-11:

```
FORMAT ('Fatalities',T25,':',1X,I4/
'Hospital Admissions ',T25,':',1X,I4/
'Outpatient care',T25,':',1X,I4/
'Total',T25,':',1X,I4/
'Chemical agent exposure',T25,':',1X,I4/
'Fracture',T25,':',1X,I4/
'Anxiety reaction',T25,':',1X,I4/
'Non-related',T25,':',1X,I4/
'Civilian casualties',T25,':',1X,I4)
```

FCAS_FATALITIES	I*4	Fatalities
FCAS_ADMINS	I*4	Hospital Admissions
FCAS_OUTPATIENT	I*4	Casualties receiving outpatient care
FCAS_TOT_CAS	I*4	Total offpost casualties
FCAS_CHEM_AGENT	I*4	Casualties exposed to a chemical agent
FCAS_FRACTURE	I*4	Casualties with bone fractures
FCAS_ANXIETY	I*4	Casualties with anxiety reactions
FCAS_NONRELATED	I*4	Casualties unrelated to chemical agent
FCAS_CIV_CASUAL	I*4	Total offpost civilian casualties

Record K (K=J+11;J=1,N;1<=N<=OS_CS_MAX):

```
FORMAT (A80)
```

FCAS_DESC(J)	C*80	Unusual event description

Filename: **OFF_EVR.DAT**

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file contains planned evacuation routes. It is accompanied by one or more graphical files with the routes.

See Also: generic.ATT, generic.DLG

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> OFFPOST EVACUATION ROUTES
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:OFFRPT.INC

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file contains up to OS_EVR_MAX (20) codes for all the evacuation routes, along with the name of the graphical file covering each route.

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K (K=J+2;J=1,N;1<=N<=OS_EVR_MAX):
FORMAT (I4,1X,A40,A32,3X)

FEVR_CODE(J)	I*4	Known route code
FEVR_NAME(J)	C*40	Known route name
FEVR_FILE(J)	C*32	Graphical file for evacuation route

Filename: **OFF_KP.ATT**

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file contains attribute data and accompanies the Known Point report, together with an associated DLG file.

See Also: OFF_KP.DAT, OFF_KP.DLG

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> OFFPOST KNOWN POINTS
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Structure: ATT file

Filename: OFF_KP.DAT

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file contains known points -- locations with known coordinates. It contains point data describing such things as reception centers, traffic control points, and roadway network nodes. It will accompany a graphical file with the same information.

See Also: KNOWN_POINTS.DAT, KNOWNPTS.DAT, OFF_KP.ATT, OFF_KP.DLG

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> OFFPOST KNOWN POINTS
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:OFFRPT.INC

Structure: ASCII, sequential, fixed format, 80 bytes/record. This files contains up to OS_KP_MAX (1000) records, each with information for a single known point.

Record 1-2: File header and description
```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K (K=J+2;J=1,N;1<=N<=OS_KP_MAX):
```
FORMAT (A40,1X,I6,1X,I2,1X,I4,1X,2F12.7)
```

FKP_NAME(J)	C*40	Name of known point
FKP_ID(J)	I*4	Unique ID for known point
FKP_MAJOR(J)	I*4	Attribute major code
FKP_MINOR(J)	I*4	Attribute minor code
FKP_LON(J)	R*8	Longitude
FKP_LAT(J)	R*8	Latitude

Filename: **OFF_KP.DLG**

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file contains known points and is a graphical version of the OFF_KP.DAT file. It contains point data describing such things as reception centers, traffic control points, and roadway network nodes. A label (the name of the known point) is attached to each point via the ATT file.

See Also: OFF_KP.DAT, OFF_KP.ATT

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> OFFPOST KNOWN POINTS
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Structure: DLG file

Filename: OFF_KPL.DAT

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file contains the names of known polygons (areas, zones, etc.), along with the name of the graphical file for each.

See Also: off_kpl.DMS, generic.ATT, generic.DLG

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> OFFPOST KNOWN POLYGONS
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:OFFRPT.INC

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file contains up to OS_KPL_MAX (1000) records, each with information about a single known polygon.

Record 1-2: File header and description
`FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)`

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K1 (K=J+2;J=1,N;1<=N<=OS_KPL_MAX):
`FORMAT (I4,1X,A40,A32,3X)`

FKPL_CODE(J)	I*4	Known Polygon ID
FKPL_NAME(J)	C*40	Known Polygon name
FKPL_FILE(J)	C*32	Graphical file

Filename: OFF_PA.DAT

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file contains the protective action orders given together with the affected areas and is accompanied by a number of graphical files.

See Also: generic.ATT, generic.DLG

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> PROTECTIVE ACTION REPORT
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:OFFRPT.INC

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file has 3 records with miscellaneous information followed by four sections of records with the following information:
1. names of the areas to be sheltered OS_PA_S_MAX (256)
2. names of the areas to be evacuated OS_PA_E_MAX (256)
3. descriptions of available resources and services OS_PA_DESC_MAX (100)
4. names of reception centers OS_PA_RC_MAX (200)

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record 3: FORMAT (A20,1X,A12,1X,A1,5(1X,I4))

FPA_AUTHORITY_L	C*20	Last name of person who gave the order
FPA_AUTHORITY_F	C*12	First name...
FPA_AUTHORITY_M	C*1	Middle initial...
FPA_NUM_PEOPLE	I*4	Total # of people to evacuate
FPA_NUM_SHELT	I*4	# of areas to be sheltered (NS)
FPA_NUM_EVAC	I*4	# of areas to be evacuated (NE)

FPA_NUM_DESC	I*4	# of services (ND)
FPA_NUM_RC	I*4	# of reception centers (NR)

Record 4: FORMAT (I4,1X,A40,A32,3X)

FPA_KR_CODE	I*4	Evacuation route (known route) code
FPA_KR_NAME	C*40	Evacuation route name
FPA_KR_FILE	C*32	Graphical file containing road closures and traffic routing as protective action measures

Record 5: FORMAT (A80)

FPA_EXPLAIN	C*80	Explanation of road closures and traffic routing as protective action measures

Record K1 (K1=J+5;J=1,NS;1<=NS<=OS_PA_S_MAX):
FORMAT (I4,1X,A40,A32,3X)

FPA_KPL_CODE_S(J)	I*4	EPZ code
FPA_KPL_NAME_S(J)	C*40	EPZ to be sheltered
FPA_KPL_FILE_S(J)	C*32	Graphical file for area to be sheltered

Record K2 (K2=J+NS+5;J=1,NE;1<=NE<=OS_PA_E_MAX):
FORMAT (I4,1X,A40,A32,3X)

FPA_KPL_CODE_E(J)	I*4	EPZ code
FPA_KPL_NAME_E(J)	C*40	EPZ to be evacuated
FPA_KPL_FILE_E(J)	C*32	Graphical file for area to be evacuated

Record K3 (K3=J+NS+NE+5;J=1,ND;1<=ND<=OS_PA_DESC_MAX): FORMAT (A80)

FPA_SERVICES(J)	C*80	Description of available emergency resources and services

Record K4 (K4=J+NS+NE+ND+5;J=1,NR;1<=NR<=OS_PA_RC_MAX):
FORMAT (A40,1X,A1,1X,I4,1X,I4)

FPA_RC_NAME(J)	C*40	Reception center name (known point)
FPA_RC_ACTIVE(J)	C*1	Reception center status (Y=Active, N=Inactive)
FPA_RC_LOAD(J)	I*4	Reception center load
FPA_RC_CAP(J)	I*4	Reception center capacity

Filename: **OFF_PEVR.DAT**

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file contains proposed evacuation routes for the onpost and is accompanied by a number of graphical files.

See Also: generic.ATT, generic.DLG

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> PROPOSED EVACUATION ROUTES
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:OFFRPT.INC

Structure: ASCII, sequential, fixed format, <= 80 bytes/record. This file contains the names of up to OS_PEVR_MAX (20) routes to be used when evacuating, along with the name of the graphical file for each route.

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K (K=J+2;J=1,N;1<=N<=OS_PEVR_MAX):
FORMAT (I4,1X,A40,A32,3X)

FPEVR_CODE(J)	I*4	ID of route for evacuating
FPEVR_NAME(J)	C*40	Route for evacuating
FPEVR_FILE(J)	C*32	Graphical file showing evacuation routes

Filename: OFF_PIMR.DAT

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file contains proposed routes for the onpost for incoming military response teams. It is accompanied by a single graphical file.

See Also: generic.ATT, generic.DLG

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> PROPOSED MILITARY ROUTES
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:OFFRPT.INC

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file contains 1 source location record with the name of the file with the proposed route, and up to OS_PIMR_MAX (20) destination location records.

Record 1-2: File header and description

FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record 3:

FORMAT (I4,1X,A40,A32,3X)

FPIMR_S_ID	I*4	Known point ID
FPIMR_S_KP	C*40	Location (known point) from where incoming military response teams come
FPIMR_FILE	C*32	Graphical file with proposed route

Record K1 (K1=J+3;J=1,N;1<=N<=OS_PIMR_MAX):

FORMAT (I4,1X,A40)

FPIMR_D_ID(J)	I*4	Known point ID
FPIMR_D_KP(J)	C*40	Location (known point) to where incoming military response teams go

Filename: **OFF_RCD.DAT**

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file contains a directory of reception center contacts.

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> RECEPTION CTR DIRECTORY
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:OFFRPT.INC

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file has information for up to OS_RCD_MAX (200) reception centers, each taking 2 records. The first record has the name and address of the reception center, the second has the contact persons name and phone number.

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K1 (K1=2*J+1;J=1,N;1<=N<=OS_RCD_MAX):
FORMAT (A40,A40)

FRCD_RC_NAME(J)	C*40	Reception center name (known point)
FRCD_RC_ADDR(J)	C*40	Reception center address

Record K2 (K2=2*J+2;J=1,N;1<=N<=OS_RCD_MAX):
FORMAT (A13,1X,A20,1X,A12,1X,A1)

FRCD_RC_PHON(J)	C*13	Reception center phone number
FRCD_NAME_L(J)	C*20	Last name of contact person
FRCD_NAME_F(J)	C*12	First name...
FRCD_NAME_M(J)	C*1	Middle initial...

Filename: OFF_RCR.DAT

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file summarizes the current load at reception centers and the number of people who have passed through them.

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> RECEPTION CTR REPORT
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:OFFRPT.INC

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file has two header records, OS_RCR_MAX (200) detail records with information about one reception center per line, followed by a single summary record.

Record 1-2: File header and description
`FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)`

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record 3-4: `FORMAT ('Reception', 32X, 'Number', 4X, 'Number', 4X, 'Number', /, 'Center', 35X, 'Received', 2X, 'Released', 2X, 'Remaining)`

Record K1 (K1=J+4;J=1,NC;1<=NC<=OS_RCR_MAX): `FORMAT (A40,I6,4X,I6,4X,I6)`

FRCR_NAME(J)	C*40	Reception center name (known point)
FRCR_REC(J)	I*4	# of people received at reception center
FRCR_REL(J)	I*4	# of people released from reception center
FRCR_REM(J)	I*4	# of people remaining at reception center

Record K2 (K2=NC+5): `FORMAT ('Total in facilities',21X,I6,4X,I6,4X,I6)`

FRCR_REC_TOT	I*4	Total # of people received at all reception centers

FRCR_REL_TOT	I*4	Total # of people released from all reception centers
FRCR_REM_TOT	I*4	Total # of people remaining at all reception centers

Filename: OFF_SHLT.DAT

Directory: OFF$SENDDIR:

Purpose: This file is sent from offpost to onpost and summarizes health-related statistics for people at shelters, reception centers, hospitals, morgues, etc.

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> OFFPOST SHELTER REPORT
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:OFFRPT.INC

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file has 3 summary records followed by up to OS_SHLT_MAX (200) pairs of records, the first with the shelter name and location, and the second with its status, load, and capacity.

Record 1-2: File header and description
`FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)`

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record 3-5:
```
FORMAT ('# for chemical agent exposure',T31,':',1X,I4)/
'# for physical trauma',  T31,':',1X,I4/
'# for anxiety',T31,':',1X,I4)
```

FSHLT_CHEM_EXP	I*4	# requiring care for chemical agent exposure
FSHLT_PHYS_TRAUMA	I*4	# requiring care for physical trauma
FSHLT_ANXIETY	I*4	# requiring care for anxiety

Record K1 (K1=2*J+4;J=1,NS;1<=NS<=OS_SHLT_MAX): `FORMAT (A40,A40)`

FSHLT_NAME(J)	C*40	Shelter name (known point)
FSHLT_AREA(J)	C*40	Area (EPZ) in which shelter is located

Record K2 (K2=2*J+5;J=1,NS;1<=NS<=OS_SHLT_MAX):

```
FORMAT ('Active?:',1X,A,1X,'Load:',1X,I4,1X,'Capacity:',1X,I4)
```

FSHLT_ACTIVE(J)	C*1	Status (Y=ACTIVE, N=INACTIVE) of shelter
FSHLT_LOAD(J)	I*4	Current load (# of evacuees) at shelter
FSHLT_CAP(J)	I*4	# of people that can be sheltered

Filename: OFF_STAT.DAT

Directory: OFF$SENDDIR:

Purpose: Transferred from offpost to onpost. This file contains several separate pieces of information pertaining to the offpost response. The file includes information on:

1. event status
2. Local scenario ID and authority
3. Proposed Protective Active Decision and comments
4. Hospitals for onpost casualties

Access:

1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> OFFPOST STATUS REPORT
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:OFFRPT.INC

Structure: ASCII, sequential, fixed format, <= 80 bytes/record. This file contains 3 records of miscellaneous information, OS_STAT_DESC_MAX (5) records of comments which should probably include the proposed protective action decision, followed by up to OS_STAT_MAX (10) records of hospital information. The first hospital is the primary hospital for onpost casualties. If it is full or inactive, then the next hospital becomes the primary hospital, and so on.

Record 1-2: File header and description

```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record 3:

```
FORMAT ('Offpost Status:',1X,A4,1X,A4,4X,'Level:',1X,A20,24X)
```

FSTAT_MODE	C*4	OPER=Operational, EXER=Exercise
FSTAT_FAZE	C*4	PREP=Planning, EVNT=Event in progress
FSTAT_CLASS	C*20	Event classification (COMMUNITY EMERGENCY, POST ONLY EMERGENCY, LIMITED EMERGENCY, NON-SURETY EMERGENCY)

Record 4: FORMAT ('Local ID:' 1X, A12, 3X, 'Whose Authority:' 1X, A20, 1X, A12, 1X, A1)

FSTAT_MCE	C*12	Local ID code
FSTAT_BOSS_L	C*20	Last name of person responsible for selecting case
FSTAT_BOSS_F	C*12	First name
FSTAT_BOSS_M	C*1	Middle initial

Record 5: FORMAT (A5,1X,A20,2X,4A13)

FSTAT_AGENCY	C*5	Agency
FSTAT_POSISH	C*20	Position
FSTAT_PHONEO	C*13	Phone # (Office)
FSTAT_PHONEH	C*13	Phone # (Home)
FSTAT_PHONEE	C*13	Phone # (Emergency)
FSTAT_PHONEB	C*13	Phone # (Beeper)

Record K1 (K1=J+5;J=1,OS_STAT_DESC_MAX): FORMAT (A80)

FSTAT_PPAD(J)	C*80	Comments, probably including the proposed protective action decision

Record K2 (K2=J+10;J=1,N;1<=N<=OS_STAT_MAX):
FORMAT (A40,1X,I4,1X,I4,1X,A1)

FSTAT_HOSPITAL(J)	C*40	Hospital (known point) for onpost casualties
FSTAT_LOAD(J)	I*4	Current load at hospital
FSTAT_CAP(J)	I*4	Capacity of hospital
FSTAT_STATUS(J)	C*1	Y=Active, N=Inactive

Filename: OFF_TC.DAT

Directory: OFF$SENDDIR:

Purpose: This file is sent from offpost to onpost, and contains a status report on traffic control points. The traffic control point names are all known points and link to entries in the Known Points report.

See Also: OFF_KP.DAT

Access:
1. IBS -> MESSAGE BOARD -> CREATE/RESEND/VIEW IBS TO ONPOST REPORT -> TRAFFIC CONTROL REPORT
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:OFFRPT.INC

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This report has two sections, each with a control header record, and containing up to OS_TC_MAX (50) data records per section.

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record 3: FORMAT (I4,1X,'Manned Traffic Control Points)

FTC_NUM_MANNED	I*4	# of staffed traffic control points (NM)

Record K2 (K2=J+3;J=1,NM;0<=NM<=OS_TC_MAX): FORMAT (A40)

FTC_MANNED(J)	C*40	Name (known point) of staffed traffic control point

Record K3 (K3=NM+4): FORMAT (I4,1X,'Unmanned Traffic Control Points)

FTC_NUM_UNMANNED	I*4	# of unstaffed traffic control points (NU)

Record K4 (K4=J+NM+4;J=1,NU;0<=NU<=OS_TC_MAX): FORMAT (A40)

FTC_UNMANNED(J)	C*40	Name (known point) of unstaffed traffic control point

Filename: OFF_TO_ON.DAT

Directory: IEMIS$SYSF:

Purpose: This file provides a mapping between onpost names and associated operational site names.

Access:
1. IBS -> SETUP -> CHANGE SITE
   When the operational site is changed, the onsite name associated with the new operational site must be set.

Include: IEMIS$INC:OFF_TO_ON.INC

Structure: ASCII, sequential, 9 bytes/record. This file contains *n* records each with an offpost site name and an onpost name.

Record n: 9 bytes
```
FORMAT (A4,1X,A4)
```

This record contains an offpost site name and an onpost name.

OFF_SITE	C*4	Offpost site name (operational directory)
ON_SITE	C*4	Onpost site name

Filename: **OTHER_SYSTEMS.DAT**

Directory: IEMIS$SYSF:

Purpose: This file contains the node and directory names of other offpost systems to which data from the onpost system on the current offpost system will be routed. This means the onpost system needs to send data to only one offpost system. That offpost system will then route the data to the other offpost systems.

See Also: MAIL_SYSTEMS.DAT, POST_SYSTEM.DAT

Access: 1. EVENT

Notes: This file must be created and maintained by the database administrator using a text editor. Because it contains passwords for user accounts, it must be protected from prying eyes. The file protections are (S:RWED,O:RWED,G,W).

If the password (or user name) changes on a destination node, this file must be updated. If the password (or user name) on your system changes, the DBAs for each of the destination node must be informed so they can update their files. If data is not arriving at a destination node, the first thing to check is that the user name and password are correct. Log on to the destination node using either FTP or RLOGIN.

Structure: ASCII, sequential, 80 bytes/record. This file contains *n* records with the node and directory names of other offpost systems.

Record n: 80 bytes
FORMAT (A80)

NODE	C*80	Node, user name, and password in the form of: node"username password"::EVNDB:[ONSITE]

Filename: P_T_UPDATE.DAT

Directory: SITE$OFF:

Purpose: Allows the People Track component to inform other nodes of the location of sheltered people.

See Also: PEOPLE_TRACK.DAT

Access:
1. LOCATE
2. EVENT

Include: SPOCC$INC:LOCATE.INC, SPOCC$FRM:LOCATE_C1.INC, IEMIS$INC:FAC.INC

Notes: Due to the sensitive nature of the people track data, this file is typically not transferred between offpost systems.

Structure: ASCII, sequential, variable length, 80 bytes/record.

Record 1-3:
```
FORMAT('CURRENT DATE / TIME: ',A23,/,
'FACILITY_TYPE: ',A16,/, 'LOCATION: ',A40)
```

CDTS	C*23	System Date and Time
FAC_TYPE(FNDX)	C*25	Facility Type
FAC_NAME(FNDX)	C*40	Facility Name

Record 4:
```
FORMAT(13HFamily name: ,A32,3X,10HOperator: ,A12)
```

FAMILY_TARGET	C*32	Family Name
OPERATOR	C*12	Operator's Name

Record 5:
```
FORMAT(9HMan:     ,A12,X,A1,X,A3,1H-,A2,1H-,A4)
```

MAN_F_TARGET	C*12	Man's First Name
MAN_M_TARGET	C*1	Middle Initial
MAN_S1_TARGET	C*3	Social Security Number Prefix
MAN_S2_TARGET	C*2	SSN Middle
MAN_S3_TARGET	C*4	SSN Suffix

Record 6:
```
FORMAT(9HWoman:   ,A12,X,A1,X,A3,1H-,A2,1H-,A4)
```

WOMAN_F_TARGET	C*12	Woman's First Name
WOMAN_M_TARGET	C*1	Middle Initial
WOMAN_S1_TARGET	C*3	Social Security Number Prefix
WOMAN_S2_TARGET	C*2	SSN Middle
WOMAN_S3_TARGET	C*4	SSN Suffix

Record 7:	FORMAT(9HChild: ,A12,X,A1,X,A3,1H-,A2,1H-,A4)		
	CHILD_F_TARGET	C*12	Child's First Name
	CHILD_M_TARGET	C*1	Middle Initial
	CHILD_S1_TARGET	C*3	Social Security Number Prefix
	CHILD_S2_TARGET	C*2	SSN Middle
	CHILD_S3_TARGET	C*4	SSN Suffix
Record 8:	FORMAT(9HFMNARF: ,A12,X,A1,X,A3,1H-,A2,1H-,A4)		
	NARF_F_TARGET	C*12	Family Member Not at Regestered Facility - First Name
	NARF_M_TARGET	C*1	Middle Initial
	NARF_S1_TARGET	C*3	Social Security Number Prefix
	NARF_S2_TARGET	C*2	SSN Middle
	NARF_S3_TARGET	C*4	SSN Suffix
Record 9:	FORMAT(22HPre-disaster Address: ,A40)		
	ADR_TARGET	C*40	Pre-disaster Address
Record 10:	FORMAT(6HCity: ,A20,X,7HState: ,A2,3X, 5HZip: ,A5,1H-,A4)		
	CITY_TARGET	C*20	Pre-disaster Address City
	STATE_TARGET	C*2	State
	ZIP_1_TARGET	C*5	Zip code
	ZIP_2_TARGET	C*4	Zip code suffix
Record 11:	FORMAT(31HPre-disaster Telephone Number: ,1H(,A3,1H),A3,1H-,A4)		
	PHONE_1_TARGET	C*3	Pre-disaster Telephone Area Code
	PHONE_2_TARGET	C*3	Telephone Exchange
	PHONE_3_TARGET	C*4	Telephone Number
Record 12-13:	FORMAT(14HInquiry from: ,A12,X,A1,X,A32,/, 14HRelationship: ,A20,X,17HMode of Inquiry: ,A9)		
	I_FIRST_NAME	C*12	Inquirer's first name
	I_M_I	C*1	Middle initial
	I_LAST_NAME	C*32	Last name
	I_RELATIONSHIP	C*20	Relationship to family
	MODE_OF_INQUIRY	C*9	Mode used to inquire: 'PHONE', 'TWX', 'RADIO', 'LETTER', or 'PERSON'

Filename: PAT.DAT

Directory: IEMIS$CHEMS:

Purpose: To provide a list of chemicals to be converted from HACS format to TMS format.

Notes: Only used in initial conversion. Not required by IBS users. Code no longer exists to use this file.

Structure: ASCII, Sequential.

Record 1: 3 bytes
FORMAT(A3)

READNAME	C*3	Chemical code.

Filename: PEOPLE_TRACK.DAT

Directory: SITE$OFF:

Purpose: Database of people at emergency facilities.

Access: 1. LOCATE

Include: SPOCC$FRM:LOCATE_A.INC

Structure: ASCII, Direct Access, fixed length, 64 bytes/record. Fifteen records, and a maximum of 10 people are allowed for each family. The value of J for a new family is 2 + 15 * NFAMILIES.

Record 1: Number of families

```
FORMAT(I10)
```

NFAMILIES	I*4	Number of families in the file

Record J:

```
FORMAT(A32)
```

FAMILY_NAME	C*32	Last Name

Record J+1 for 10:

```
FORMAT(A12,A1,A3,A2,A4,A3,A1,A1)
```

F_N(I)	C*12	First Name
M_I(I)	C*1	Middle Initial
SSN_1(I)	C*3	First part of SSN
SSN_2(I)	C*2	2nd part of SSN
SSN_3(I)	C*4	3rd part of SSN
AGE(I)	C*3	Age
P_S(I)	C*1	
		'H' = 'HOSPITALIZED'
		'O' = 'OUTPATIENT CARE'
		'F' = 'FATALITY'
P_H(I)	C*1	
		'C' = 'CHEMICAL AGENT'
		'F' = 'FRACTURE'
		'A' = 'ANXIETY RELATED'
		'N' = 'NON RELATED'

Record J+11:

```
FORMAT(A40)
```

PRE_DISASTER_ADR	C*40	Pre-Disaster Address

Record J+12:

```
FORMAT(A20,A2,A5,A4)
```

CITY	C*20	City of residence

STATE	C*2	State abbreviation
ZIP_1	C*5	5 digit zip code
ZIP_2	C*4	4 digit zip code suffix

Record J+13:

FORMAT(A3,A3,A4,A1,A20,20X)

PRE_D_PHONE_1	C*3	Area Code
PRE_D_PHONE_2	C*3	Telephone Exchange
PRE_D_PHONE_3	C*4	Telephone Number
CONFIDENTIAL	C*1	Y if confidential
CDTS(1:20)	C*23	System date and time

Record J+14:

FORMAT(A40,A13)

LOCATION	C*40	Shelter Location
FACILITY_PHONE	C*13	Shelter Phone Number

Filename: PERMITI.DAT

Directory: IEMIS$SYSF:

Purpose: This file defines the operations for which the user has privilege while using IBS.

Access:
1. PDMGR
2. PD_ADD
3. PD_DELETE

Include: IEMIS$INC:PERMISSION.INC

Notes: The value in PERMIT(i) is used to set PERMISSION_GRANTED(i), which is equivalent to PRIV_*, the actual variable to which the software refers. IBS imposes no limit on the number of records in the file.

Structure: ASCII, sequential, 132 bytes/record. This file contains *n* records, one for each user, with *m* optional comment records interspersed throughout. Comment lines beginning with NOT - A NAMEP are used by the PDMGR program to generate form help.

Record 1: Header
```
FORMAT (*)
```

Record 2-n: 132 bytes, user privileges record.
```
FORMAT (A12,A120)
```

This record contains a user name field, followed by 120 Yes/No flags telling whether the user has the privilege to perform the associated activity. If for some reason a flag is not set to Y or N, it is assumed to be N.

USER_NAME	C*12	User name (login name)
PERMIT	C*120	Privilege flags (Y/N)

#	Privilege Name	Privilege Description	Plan	Oper
1	D2	Run D2 Model	Y	-
2	IDYNEV	Run IDYNEV Model	Y	N
3	CHEMS	Run CHEMS Models	Y	N
4	OSPM	Run OSPM Model	Y	N
5	MESORAD	Run Mesorad Model	Y	N
6	REA	Run Regional Evacuation Analysis	Y	N
7	EESF	Run Emergency Exercise Simulation Facility	Y	N
11	IBS	Run IBS and Offspring	-	-
12	NEWSITE	Run NEWSITE to Create a New Site	-	-
13	BUILDNET	Run BUILDNET (IDYNEV First Cut Approximation)	Y	N

#	Privilege Name	Privilege Description	Plan	Oper
14	GEDIT	Run GEDIT (Graphical Editor)	Y	-
15	MPDISPLAY	Run MPDISPLAY (Map Display)	-	-
16	DELSITE	Run DELSITE to Delete a Site	-	-
17	NEWLOG	Run NEWLOG to Create New Log Journals	-	-
18	ARCHIVE	Run SITEBKP to Backup a Site	Y	-
21	SW_MANAGER	Create/send IBS to Onpost reports	-	-
22	WS_MANAGER	Create/send Onpost to IBS reports (ONPOSTSIM)	-	-
23	PER_MANAGER	Edit Personnel Database	-	-
24	POP_MANAGER	Edit Special Population Database	-	-
25	LOC_MANAGER	Edit People Location Database	N	-
26	RA_MANAGER	Edit Risk Area Analysis Database	-	-
27	RM_MANAGER	Edit Facility & Resource Database	-	-
28	LGD_MANAGER	Edit the Common Legend	N	-
29	PD_MANAGER	Run PDMGR to Edit the Permissions Database	-	-
30	CNX_MANAGER	Edit the shared CaseIndex file	N	-
31	D2_ANIM	Do D2 Animation	Y	-
32	PLANNER	Do Planning Functions	-	-
33	COMBINED_LOG	Create the Combined Log	-	-
34	ADD_TO_LOG	Annotate the Log Journal	-	-
35	CHG_OPER_SITE	Change the Operational Site	-	-
36	RESTORE_SHARED	Restore Shared Data from Backup	-	-
37	DECLARE_EVENT	Declare an Event	N	-
38	CI_EDIT	Check in/out from the shared CaseIndex	-	-
39	D2_SELECT	Select D2 case	-	-
40	IDYNEV_SELECT	Select Evacuation case	-	-
41	IP_CREATE	Create an IP	Y	N
42	IP_SELECT	Select a new Common IP	Y	-
43	IP_CHANGE	Change the Common IP	Y	-
44	IP_USE	Use (Checkoff) Tasks in the Common IP	N	-
45	IP_VALIDATE	Validate an IP	Y	-
46	IP_LINK	Link an IP to D2 & IDYNEV Cases	Y	N
51	COPY_TO_INFO	Copy Data to Infomanager	-	-
52	COPY_TO	Copy Data to Some Other User	-	-
53	COPY_FROM	Copy Data from Some Other User	-	-
61	EOC_ACTIVATION	Do EOC Activation Functions	-	-
62	SEARCH_RESCUE	Do Search & Rescue Functions	-	-
63	DAMAGE_ASSESS	Do Damage Assessment Functions	-	-
81	EF	Emergency Functions (1 of 40)	Y	-

Filename: **PERMITI.DFT**

Directory: IEMIS$SYSF:

Purpose: A list of user types (e.g. NORMAL, ALL, PLANNER, PNL) and the privileges to be initially assigned to a user who is in that type.

See Also: PERMITI.DAT

Access: NEWUSER

Notes: NEWUSER prompts for user type, and it assigns the new user the privileges assigned to the selected user type.

Structure: ASCII, sequential, variable length, 132 bytes/record. See PERMITI.DAT for more detail.

Record 1: Header, including filename, date, time, and version number.

Record 2-n: FORMAT(A12,A120)

USER_TYPE	C*12	User Classification
PERMIT	C*120	Privilege Flag (Y/N)

Filename: PERSONNEL.DAT

Directory: OFF$DIR:

Purpose: This file contains information for emergency response personnel.

Access: This file is used in conjunction with PERSONNEL BOARD and RESOURCE MANAGEMENT which can be used to create and update it. The standalone utility RMMGR can do the same.

Include: IEMIS$INC:PER.INC

Notes: Routine M_PER_READ reads the file. The parameter PER_MAX limits the number of records in the file.

Structure: ASCII, direct access, 128 bytes/record, shared and locked. This file contains *4n* records (four records are required to hold the information for each person): 1) Name and telephone information; 2) Address, agency, and availability; 3) Schedule; and 4) Positions held.

Record 4n-3: 128 bytes (n=1,PER_MAX)
`FORMAT (A23,A20,A12,A1,A20,4A13)`

This record contains name and telephone information for a single person.

PER_DATE	C*23	Date/Time of last update to record
PER_NAMEL	C*20	Last name of person
PER_NAMEF	C*12	First name of person
PER_NAMEM	C*1	Middle initial of person
PER_TITLE	C*20	Job title
PER_PHONEO	C*13	Office phone #
PER_PHONEH	C*13	Home phone #
PER_PHONEB	C*13	Beeper phone #
PER_PHONEE	C*13	Emergency phone #

Record 4n-2: 128 bytes (n=1,PER_MAX)
`FORMAT (A40,A20,A2,A10,A5,A1,50X)`

This record contains address, agency, and availability information for a single person.

PER_ADDR	C*40	Address
PER_CITY	C*20	City
PER_STATE	C*2	State abbreviation
PER_ZIP	C*10	ZIP code
PER_AG_CODE	C*5	Agency acronym or abbreviation
PER_AVAIL	C*1	Is person available? (Y,N)

Record 4n-1: 128 bytes (n=1,PER_MAX)
FORMAT (7(A2,':',A2,'-',A2,':',A2,' '))
This record contains schedule information for a single person.

PER_BEG_HH(i)	C*2	Hour person arrives at work on day i
PER_BEG_MM(i)	C*2	Minute person arrives at work on day i
PER_END_HH(i)	C*2	Hour person leaves work on day i
PER_END_MM(i)	C*2	Minute person leaves work on day i

Record 4n: 128 bytes (n=1,PER_MAX)
FORMAT (5(A1,A20))

This record contains information about the positions/jobs a single person holds.

PER_STATUS(i)	C*1	Is person currently responsible for this position? (Y,N)
PER_POSISH(i)	C*20	Positions for which person can be responsible

Filename: PLUMELABS.DAT

Directory: IEMIS$MESORAD:

Purpose: Provides labels that are displayed as legend information when MESORAD plumes are displayed graphically.

Access:
1. MES_GR->MESORAD GRAPHIC OPTIONS->Windfield
2. EESF_GR->Display Map-Related Graphics->Windfield

Include: EESF$INC:BGMAP.INC

Structure: ASCII, Sequential. A set of three records is provided for each MESORAD graphic output selection.

Record K (K = I*3 - 2): 36 bytes (I is the MESORAD graphic type index - maximum value of I is 19).

```
FORMAT (A,A,I1,A26)
```

TAG(I)	C*6	Root file name for graphic type I input (as used by Make_filename).
EXT(I)	C*3	Extension to root file name for graphic type I input (as used by Make_filename).
K	I*4	Flag indicating plume scaling for graphic type I is logarithmic if not equal to 0.
LAB(1,I)	C*26	First line of label for graphic type I.

Record K2 (K2 = K + 1): 26 bytes

```
FORMAT (A26)
```

LAB(2,I)	C*26	Second line of label for graphic type I.

Record K3 (K3 = K + 2): 26 bytes

```
FORMAT (A26)
```

LAB(3,I)	C*26	Third line of label for graphic type I.

Filename: **POPCODES.DAT**

Directory: IEMIS$SYSF:, SITE$DIR:

Purpose: This file is used to maintain a list of population/evacuation model centroid cross-reference tables.

Access:
1. IBSSH -> Models -> REA
2. IBSSH -> Models -> BUILDNET

Include: IEMIS$INC:JOB_ENV.INC

Structure: ASCII, sequential

Record 1,MAXIMUM_POPS: 20 bytes
FORMAT (A20)

POP_STRING(I)	C*20	The first four characters of the POP_STRING are known as the POP_CODE and are used in constructing the name of the file containing the associated cross-reference table.

Filename: POSITION_LOOKUP.DAT

Directory: OFF$DIR:

Purpose: This file contains a list of agency acronyms or abbreviations together with positions (jobs) people can hold at each agency.

Access:
1. IBS -> RESOURCE MANAGEMENT
2. IBS -> PERSONNEL BOARD
3. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES -> USE/CHANGE TASK
4. RMMGR

Include: IEMIS$INC:POS.INC

Notes: Routine M_POS_READ reads the file. The parameter POS_MAX limits the number of records in the file.

Structure: ASCII, direct access, 32 bytes/record, shared and locked. This file contains *n* records, each with an agency acronym or abbreviation followed by a position/job title.

Record n: 32 bytes

```
FORMAT (A5,A20)
```

This record contains an agency acronym or abbreviation followed by a position/job title.

POS_AG_CODE	C*5	Agency acronym or abbreviation
POS_NAME	C*20	Position or job title

Filename: POST_SYSTEM.DAT

Directory: IEMIS$SYSF:

Purpose: Provides communication parameters necessary to communicate with the post. Only the primary IBS node communicates with the post. This file is empty at all other nodes.

See Also: MAIL_SYSTEMS.DAT, OTHER_SYSTEMS.DAT

Access:
1. EVENT
2. IBS -> MESSAGE BOARD -> CREATE/RESEND IBS -> ONPOST REPORT

Include: IEMIS$INC:EVENT.INC

Notes: Only the primary IBS node communicates with the post. All other IBS nodes receive post communication through the primary node. Because the onpost system may be case sensitive, all characters in this file should be lower case.

This file must be created and maintained by the database administrator using a text editor. Because it contains passwords for user accounts, it must be protected from prying eyes. The file protections are (S:RWED,O:RWED,G,W).

If the account name or password of the onpost machine is changed, this file must be updated. If data is not arriving onpost, the first thing to check is that the username and password are correct: Log on to the onpost computer using either FTP or RLOGIN.

Structure: ASCII, sequential, variable length, 39 bytes/record.

Record 1: `FORMAT(A12, 1X, A12, 1X, A12)`

HOST	C*12	Network Node Name
USER	C*12	Account to log on through
PASS	C*12	Password for the account

Filename: **RESOURCE.DAT**

Directory: OFF$DIR:

Purpose: This file contains information regarding resources.

Access:
1. IBS -> RESOURCE MANAGEMENT
2. IBS -> RMMGR

Include: IEMIS$INC:RES.INC

Notes: Routine M_RES_READ reads the file. The parameter RES_MAX limits the # of records in the file.

Structure: ASCII, direct access, 128 bytes/record, shared and locked. This file contains *n* records, each with information for a single resource type and description combination at a single facility. Resources that have not yet been assigned to any facility will have a blank facility name.

Record n: 128 bytes
```
FORMAT (A8,A32,I10,A40,A8,A23)
```

This record contains information for a single resource type and description combination at a single facility.

RES_TYPE	C*8	Resource type (references an item in the icon control file)
RES_DESC	C*32	Resource description
RES_QUANT	I*4	Amount of resource at facility
RES_LOC	C*40	Facility name
RES_UNIT	C*8	Units in which resource is measured
RES_DATE	C*23	Date/Time resource was placed at the facility

Filename: RM.DMS

Directory: MAP$DATA$RM:

Purpose: This file is the graphical representation of the location of the facilities and their resources. It is created on the fly when a user leaves Resource Management and may be used as a normal map topography elsewhere in the system. It is not used within Resource Management.

Access: 1. IBS -> RESOURCE MANAGEMENT

Structure: DMS Format

Filename: **RP.DMS**

Directory: MAP$DATA$RP:

Purpose: This file contains the user-defined risk polygons for RISK AREA ANALYSIS.

Access: This file is accessed at the start of RISK AREA ANALYSIS and when the risk polygons are modified by the user with menu functions DRAW RISK POLYGON and DELETE RISK POLYGON.

Include: IEMIS$INC:RISK_AREA.INC

Structure: DMS format

Filename: RPTSUBJ.DAT

Directory: IEMIS$SYSF:

Purpose: Contains search criteria for producing custom reports from the journal and important mail files. Entries must be added or modified manually by editing this file. Then when creating a log report, the entries in this file, together with the predefined ones, can be selected as search criteria. All journal and important mail entries which match at least one of the search criteria will be placed in the log report.

Access:
1. IBS -> MESSAGE BOARD -> CREATE COMMON LOG REPORT
2. IBS -> MESSAGE BOARD -> CREATE PRIVATE LOG REPORT

Include: SPOCC$INC:CMBLOG.INC

Structure: ASCII, sequential, variable length, 40 bytes/record.

Record 1-2: Header
```
FORMAT(' MESSAGE CATEGORY'/' -------------------------')
```

Record 3-n:
```
FORMAT(1X,A25)
```
CMB_CAT_NAME(I) C*25 Message categories (pre-defd + user-defd)

Filename: SCEN_INDEX.DAT

Directory: SITE$IP:

Purpose: This file provides an index to the scenario table so that specific records can be accessed, obviating the need to read the entire file. It was created because the size of the SCEN_TABLE.DAT can become quite large and slow to read.

See Also: SCEN_TABLE.DAT

Access:
1. IBS -> EMERGENCY ACTIVITIES -> HAZARD ANALYSIS -> EXECUTE D2 -> CREATE SCENARIO FOR CURRENT D2 CASE
2. IBS -> STATUS BOARD -> SELECT IP

Notes: This file contains a 10-character code to index a given scenario record in SCEN_TABLE.DAT. The code is as follows:

(1:1) Population pattern 1 --> SEASONAL
2 --> WEEKDAY
3 --> WEEKNIGHT
4 --> WEEKEND
(2:4) Dispersion (D2) case number
(5:7) Evacuation (IDYNEV) case number
(8:10) IP number associated with record

When the appropriate code is found, the current record number is also the record number to be accessed in SCEN_TABLE.DAT. The SCEN_INDEX.DAT is a sequential file to be scanned quickly, and the SCEN_TABLE.DAT is a direct access file to be read on a record-by-record basis after the appropriate index has been found.

Structure: ASCII, sequential, 10 bytes/records. This file contains *n* records, each an index as described previously.

Record n:

```
FORMAT (A10)
```

IP_INDEX_CODE C*10 Index code described above

Filename: SCEN_TABLE.DAT

Directory: SITE$IP:

Purpose: This file is used to link model data with a particular IP. After an IP has been created, the planner will link the IP to several scenarios which might lead to the implementation of the IP. At the time of an event, the INFOMANAGER will search the database to find which scenario best fits the current situation. The IP linked to this scenario will then be used.

Access:
1. PLANNING MODE
   IBS -> EMERGENCY ACTIVITIES -> HAZARD_ANALYSIS -> EXECUTE D2

2. OPERATIONAL MODE
   A. IBS -> STATUS BOARD -> IMPLEMENTING PROCEDURES CHANGE IP
   B. IBS -> STATUS BOARD -> CHANGE ONPOST WORK PLAN -> CHANGE IP

Include: IEMIS$INC:SCEN_EQUIV.INC, IEMIS$INC:SCEN_TABLE.INC

Structure: Direct access, 1551 bytes/record. See SCEN_INDEX.DAT for information on accessing this file.

Record 1-n:

```
FORMAT (I3,I3,I4,1X,A9,1X,A8,1X,A2,A3,A3,F6.1,F5.1,1X,E13.6,A1,
I3,F6.2,1X,F6.2,F5.2,A80,A80,A13,A10,I3,256(A5))
```

F_IP	I*3	IP number linked with record
F_OFFPOST_D2	I*3	Offpost D2 case
F_ONPOST_D2	I*4	Onpost D2 case
F_D2LOG_DATE	C*9	Date of D2LOG creation
F_D2LOG_TIME	C*8	Time of D2LOG creation
F_AGENT	C*2	D2 agent
F_RELEASE	C*3	D2 release
F_MUNITION	C*3	D2 munition
F_WIND_DIRECTION	R*8	D2 wind direction
F_WIND_SPEED	R*8	D2 wind speed
F_QUANTITY	R*8	D2 quantity of release
F_STABILITY_CLASS	C*1	D2 stability class
F_EVAC_CASE_NUMBER	I*4	IDYNEV case number
F_EVAC_PCT	R*8	IDYNEV percent evacuated
F_EVAC_LOAD_TIME	R*8	IDYNEV load time
F_EVAC_PER_CAR	R*8	IDYNEV people per car
F_SCENARIO_DESCRIPTION		

	C*80	Short IP record description
F_SCEN_D2_DESC	C*80	Short D2 record description
F_POPULATION_TYPE	C*13	Population type
F_BLANKS	C*10	For future use
F_NUM_ZONES	I*4	Number of zones for given scenario
F_ZONES (256)	C*5	Zone names themselves

Filename: SETUP.COM

Directory: SITE$DIR:

Purpose: This file is a command procedure for assigning logical variables associated with a site and its volatile map topographies.

Access: NEWSITE, IBS applications. This file is accessed during startup of most IBS applications in the environment initialization phrase. The user may edit the file to reassign logical names to refer to different directories, control files, or topographies than the default configuration. It can also be executed from the DCL command level, though four parameters must be specified: disk, username, site, and offsite. For example,

@disk:[user.SITES.site]SETUP disk user site offsite

or

@SITE$DIR:SETUP disk user site offsite

The NEWSITE program creates the file by copying the template file from IEMIS$SYSF:SETUP.COM_TEMPLATE, renaming and modifying it to execute the appropriate map database association file: mapDB:[map]SETUPDB.COM.

Structure: DCL commands

Filename: SETUP.COM_TEMPLATE

Directory: IEMIS$SYSF:

Purpose: This file is a command procedure for assigning logical variables associated with a site and its volatile map topographies. It is used as a template and copied to SITE$DIR:SETUP.COM when a user creates a new site.

See Also: SITE$DIR:SETUP.COM

Access: This file is accessed from the NEWSITE program.

Structure: DCL commands

Filename: **SETUPDB.COM**

Directory: MAP$DATA$MAIN:

Purpose: This file is a command procedure for assigning system logical variables associated with the map topographies.

See Also: IEMIS$SYSF:SETUPDB.COM_TEMPLATE

Access: This file is accessed from the DCL command line via a symbol, xxxxDATATBASE or SYSxxxxDATABASE, where xxxx is the name of the map database. A database administrator may edit the file to reassign logical names to refer to different directories, control files, or topographies other than the default configuration.

The MAPDBGEN program creates this file by copying the template IEMIS$SYSF:SETUPDB.COM_TEMPLATE, and renaming and modifying it to point to the new map database.

Notes: The xxxxDATABASE symbol defines the map logical for your process; the SYSxxxxDATABASE symbol defines the map logicals for all users.

Structure: DCL commands

Filename: SETUPDB.COM_TEMPLATE

Directory: IEMIS$SYSF:

Purpose: This file is a command procedure for assigning system logical variables associated with the map topographies. It is used as a template by the MAPDBGEN program when creating a new map database.

See Also: MAP$DATA$MAIN:SETUPDB.COM

Structure: DCL commands

Filename: SIGEVENT.DAT

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: Transferred from onpost to offpost. This file reports significant onpost events.

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> SIGNIFICANT EVENTS
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. ONPOSTSIM
4. EVENT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes: When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file contains up to WS_SIG_MAX (100) records describing significant activities during an event.

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K (K=J+2;J=1,N;1<=N<=WS_SIG_MAX): FORMAT (A80)

WS_SIG_DESC(J)	C*80	Description of significant events

Filename: SIREN.LIB

Directory: IEMIS$SYSF:

Purpose: Output levels and rotating characteristic of known sirens.

Access: 1. OSPM_(NO)GR

Include: OSPM$INC:OSPM.INC

Structure: ASCII, sequential, fixed length, 82 bytes/record.

Record 1-N:

FORMAT(A,8(F7.1),A)

SIREN_NAME	C*25	Name of the siren.
HZ2	R*4	Siren output at 63 Hz.
HZ3	R*4	Siren output at 125 Hz.
HZ4	R*4	Siren output at 250 Hz.
HZ5	R*4	Siren output at 500 Hz.
HZ6	R*4	Siren output at 1000 Hz.
HZ7	R*4	Siren output at 2000 Hz.
HZ8	R*4	Siren output at 4000 Hz.
HZ9	R*4	Siren output at 8000 Hz.
CTYPE(SIREN_TYPE)	C*1	Type of siren R - Rotating N - Nonrotating

Filename: **SITE.DAT**

Directory: IEMIS$SYSF:

Purpose: This file contains a list of all depot sites.

See also: SYS$LOGIN:SITE.DAT

Access: This file should be used as a template by the database administrator when creating new map databases and by users when creating a new site. The depot site of interest can be extracted and used by everyone as the default.

Structure: see SYS$LOGIN:SITE.DAT

Filename: **SITE.DAT**

Directory: SYS$LOGIN:, INFO$DIR:

Purpose: The user's version of this file, SYS$LOGIN:SITE.DAT contains a list of planning sites owned by a particular user. The INFOMANAGER version, INFO$DIR:SITE.DAT, contains the list of operational sites.

See Also: IEMIS$SYSF:SITE.DAT_TEMPLATE

Access: Most of the models and map-based programs read this file during the environment initialization phase to get more information about the particular site specified in the user's JOB_ENV.DAT.

Include: IEMIS$INC:SITES.INC

Notes: If JOB_ENV.DAT points to the operational site, INFO$DIR:SITE.DAT is used; otherwise, SYS$LOGIN:SITE.DAT is used.

Structure: ASCII, sequential, fixed length, 98 bytes/record. This file contains two header records and *n* site records.

Record 1-2: 98 bytes, header

```
FORMAT (A)
```

The header records specify the version of the file and identify the various columns.

Record n+2: 98 bytes, site records

```
FORMAT (1X,A4,1X,A15,1X,A1,1X,A4,1X,2F10.4,1X,F7.2,1X,A40)
```

This record contains information for a single user site.

SITE_CODE	C*4	Site abbreviation (directory name)
SITE_DISK	C*5	Disk on which site is located

SITE_TYPE	C*1	R = Real site (Infomanager only) E = Exercise site (Infomanager only) P = Planning site (Not Infomanager)
SITE_OTHR	C*4	If a Real site has an associated Exercise site, this is the exercise site. Otherwise, it is blank. If it is an Exercise or Planning site, it is the Real site under Infomanager.
SITE_LONG	R*8	Longitude of site center
SITE_LATI	R*8	Latitude of site center
SITE_SIZE	R*8	Radius in miles of evacuation area
SITE_NAME	C*40	Full name of site

Filename: SITE.DAT

Directory: SYS$LOGIN:, INFO$DIR:

Purpose: The user's version of this file, SYS$LOGIN:SITE.DAT contains a list of planning sites owned by a particular user. The INFOMANAGER version, INFO$DIR:SITE.DAT, contains the list of operational sites.

See Also: IEMIS$SYSF:SITE.DAT_TEMPLATE

Access: Most of the models and map-based programs read this file during the environment initialization phase to get more information about the particular site specified in the user's JOB_ENV.DAT.

Include: IEMIS$INC:SITES.INC

Notes: If JOB_ENV.DAT points to the operational site, INFO$DIR:SITE.DAT is used; otherwise, SYS$LOGIN:SITE.DAT is used.

Structure: ASCII, sequential, fixed length, 98 bytes/record. This file contains two header records and *n* site records.

Record 1-2: 98 bytes, header
`FORMAT (A)`

The header records specify the version of the file and identify the various columns.

Record n+2: 98 bytes, site records
`FORMAT (1X,A4,1X,A15,1X,A1,1X,A4,1X,2F10.4,1X,F7.2,1X,A40)`

This record contains information for a single user site.

SITE_CODE	C*4	Site abbreviation (directory name)
SITE_DISK	C*5	Disk on which site is located
SITE_TYPE	C*1	R = Real site (Infomanager only) E = Exercise site (Infomanager only) P = Planning site (Not Infomanager)
SITE_OTHR	C*4	If a Real site has an associated Exercise site, this is the exercise site. Otherwise, it is blank. If it is an Exercise or Planning site, it is the Real site under Infomanager.
SITE_LONG	R*8	Longitude of site center
SITE_LATI	R*8	Latitude of site center
SITE_SIZE	R*8	Radius in miles of evacuation area
SITE_NAME	C*40	Full name of site

Filename: SITE.DAT_TEMPLATE

Directory: IEMIS$SYSF:

Purpose: This file is merely a template containing no sites and is used when creating new user accounts.

See also: SYS$LOGIN:SITE.DAT

Access: The NEWUSER.COM program copies this file to the new user's login directory and renames it to SITE.DAT. The implication is the user initially has no sites and JOB_ENV.DAT points to the operational site.

Filename: SITE_EF.DAT

Directory: OFF$DIR:

Purpose: This file defines the emergency functions which can be used during the creation and use of implementing procedures.

Access:
1. IBS(IEMIS_INIT→SM_EF_READ)
2. SM_INIT
3. SM_READ

The database administrator can create and edit this file using a text editor. If any changes are made, then re-initialization of shared memory must occur before those changes take effect. After ensuring that no one is using the system, use either of the following commands:

$ RUN IEMIS$EXE:SM_INIT which re-initializes all of shared memory, or
$ RUN IEMIS$EXE:SM_READ which re-initializes just the emergency function portion of shared memory.

If the DBA makes some modifications and wants all new operational sites on the system to use them, then the file should be copied to IEMIS$SYSF:SITE_EF.DAT_TEMPLATE, replacing the original file.

Include: IEMIS$INC:SM_BOTH.INC

Notes: After this file is created for a particular site, items should NOT be removed or modified or the implementing procedures will be corrupted.

Structure: ASCII, sequential, 20 bytes/record. This file contains as many as 40 records, each with the name of an emergency function for which plans will be created.

Record n: 20 bytes, Emergency Function Name Record
FORMAT (A20)

SM_EF_NAME	C*20	The name of the emergency function

Filename: SITE_EF.DAT_TEMPLATE

Directory: IEMIS$SYSF:

Purpose: This file defines the default emergency functions that can be used during the creation and use of implementing procedures.

See Also: SITE_EF.DAT

Access: The NEWSITE program uses this file as a template when creating a new site, copying it, if necessary, to OFF$DIR, and renaming it SITE_EF.DAT. The DBA can then edit the file.

Filename: **SLICEnnn.DAT**

Directory: OFF$SENDDIR:

Purpose: This file is sent from offpost to onpost and is a copy of CnnnSLICE.DAT, with the addition of the transfer header records. It is not used by EMIS.

See Also: CnnnSLICE.DAT, ONSITE$DIR:SLICEnnn.DAT

Access:
1. IBS -> MESSAGE BOARD -> CREATE IBS -> ONPOST RPT
2. IBS -> MESSAGE BOARD -> RESEND IBS -> ONPOST RPT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes: This file is identical to CnnnSLICE.DAT except for the file transfer header lines.

Structure: ASCII, sequential, variable format, <= 80 bytes/record.

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record N: See CnnnSLICE.DAT

Filename: **SLICEnnn.DAT**

Directory: OFF$SENDDIR:

Purpose: This file is sent from offpost to onpost and is a copy of CnnnSLICE.DAT, with the addition of the transfer header records. It is not used by EMIS.

See Also: CnnnSLICE.DAT, ONSITE$DIR:SLICEnnn.DAT

Access:
1. IBS -> MESSAGE BOARD -> CREATE IBS -> ONPOST RPT
2. IBS -> MESSAGE BOARD -> RESEND IBS -> ONPOST RPT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes: This file is identical to CnnnSLICE.DAT except for the file transfer header lines.

Structure: ASCII, sequential, variable format, <= 80 bytes/record.

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record N: See CnnnSLICE.DAT

Filename: **SLICEnnn.DAT**

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: This file is sent from onpost to offpost and contains dosage/time attributes for the D2 track. This file is written by the PARDOS routine.

See Also: CnnnSLICE.DAT, OFF$SENDDIR:SLICEnnn.DAT

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> D2 SLICE
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. ONPOSTSIM
4. EVENT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes: This file is identical to CnnnSLICE.DAT except for the file transfer header lines placed by the onpost system. This file is read from the PLUME routine. When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, variable format, <= 80 bytes/record.

Record 1-2: File header and description
```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record N: See CnnnSLICE.DAT

Filename: SPECIAL_POPULATION.DAT

Directory: OFF$DIR:

Purpose: This file contains the special population data about the number of people and the type of special populations (transportation-dependent, etc.) at certain facilities.

Access:
1. IBS Special Population operations (ADD, UPDATE, DELETE SPECIAL POPULATION) within DIRECTION/CONTROL
2. SPECIAL POPULATION SUMMARY within DIRECTION/CONTROL
3. RISK AREA POPULATION SUMMARY within RISK AREA ANALYSIS
4. During deletion of a facility that has special population data associated with it.

Include: IEMIS$INC:SP.INC

Structure: ASCII, direct access, 256 bytes/record, shared and locked

Record 1: 256 bytes
FORMAT (A40,I5,A1,A1,A60,A1,A60,A1,A60)

Contains special population data for one facility.

SP_NAME	C*40	Facility name
SP_COUNT	I*4	Population count at facility
SP_ISFLAG	C*1	Isolated population flag (Y/N)
SP_TDFLAG	C*1	Transportation dependent flag (Y/N)
SP_TDNOTE	C*60	Transportation dependent notes
SP_ANFLAG	C*1	Alert/notification flag (Y/N)
SP_ANNOTE	C*60	Alert/notification notes
SP_OTFLAG	C*1	Other needs flag (Y/N)
SP_OTNOTE	C*60	Other needs notes

Filename: SPOCC_ACTIVITY.LOG

Directory: OFF$DIR:

Purpose: This file maintains a journal of database changes for the IBS system for the local node. This file contains pointers to detail records contained in SPOCC_ACTIVITY.LOG_DETAIL.

See Also: SPOCC_ACTIVITY.LOG_DETAIL

Access: 1. IBS -> MESSAGE BOARD -> CREATE/VIEW COMMON/PRIVATE LOG REPORT

Structure: ASCII, direct access, fixed length, 128 bytes/record

Record 1:

```
FORMAT (I6,A)
```

IREC_1+1	I*4	Pointer to next unused record in file
BLANKS	C*122	Space filler

Record 2-n:

```
FORMAT (4A,2I6,A)
```

TSOURCE	C*20	Function or user that made the log entry
TLDATETIME	C*23	Current date and time (VMS format)
TADATETIME	C*23	Actual date and time of activity
TMSGTYPE	C*25	Primary key
FIRST	I*4	Pointer to first record in associated detail file
LAST	I*4	Pointer to last record in associated detail file
TUNUSED	C*25	Space filler

Filename: **SPOCC_ACTIVITY.LOG_DETAIL**

Directory: OFF$DIR:

Purpose: This file maintains a journal of database changes for the IBS system for the local node. It contains the detail records referenced by SPOCC_ACTIVITY.LOG.

See Also: SPOCC_ACTIVITY.LOG

Access: 1. IBS -> MESSAGE BOARD -> CREATE/VIEW COMMON/PRIVATE LOG REPORT

Notes: Applications can write as many detail records for a given log entry as desired.

Structure: ASCII, direct access, fixed length, 80 bytes/record

Record n:

```
FORMAT (A)
```

DETAILS(I)	C*80	Free format text

Filename: SPOCC_EVENT.LOG

Directory: OFF$DIR:

Purpose: This file maintains a journal of reports sent to the onpost system from the local IBS node. It contains pointers to detail records contained in SPOCC_EVENT.LOG_DETAIL.

See Also: SPOCC_EVENT.LOG_DETAIL

Access:
1. IBS -> MESSAGE BOARD -> VIEW IBS -> ONPOST RPT
2. IBS -> MESSAGE BOARD -> CREATE/VIEW COMMON/PRIVATE LOG REPORT

Structure: ASCII, direct access,fixed length, 128 bytes

Record 1:

```
FORMAT (I6,A)
```

TREC_1+1	I*4	Pointer to next unused record in file
BLANKS	C*122	Space filler

Record 2-n: 128 bytes

```
FORMAT (4A,2I6,A)
```

TSOURCE	C*20	Function or user that made the log entry
TLDATETIME	C*23	Current date and time (VMS format)
TADATETIME	C*23	Actual date and time of activity
TMSGTYPE	C*25	Primary key
FIRST	I*4	Pointer to first record in associated detail file
LAST	I*4	Pointer to last record in associated detail file
TUNUSED	C*25	Space filler

Filename: **SPOCC_EVENT.LOG_DETAIL**

Directory: OFF$DIR:

Purpose: This file provides a journal of reports sent to the onpost system from the local IBS node. It contains the detail records referenced by SPOCC_EVENT.LOG.

See Also: SPOCC_EVENT.LOG

Access:
1. IBS -> MESSAGE BOARD -> VIEW IBS -> ONPOST RPT
2. IBS -> MESSAGE BOARD -> CREATE/VIEW COMMON/PRIVATE LOG REPORT

Notes: Applications can write as many detail records for a given log entry as desired.

Structure: ASCII, direct access, fixed length, 80 bytes/record

Record 1-n:

FORMAT (A)

DETAILS(I)	C*80	Free format text

<u>Filename</u>: **SR.DMS**

<u>Directory</u>: MAP$DATA$SR:

<u>Purpose</u>: This DMS file contains the user-defined search and rescue polygons for SEARCH/RESCUE.

<u>Access</u>: This file is accessed at the start of SEARCH/RESCUE, and when the search and rescue polygons are modified by the user with menu functions DRAW SEARCH AREAS and DELETE SEARCH AREAS.

<u>Include</u>: IEMIS$INC:RISK_AREA.INC

<u>Structure</u>: DMS format

Filename: **T.DAT**

Directory: MAP$DATA$xx:

Purpose: T.DAT contains a list of the DMS files in the topography. It is used to create the LIMITxx.BIN file, and is of no use after that file has been created. IBS does not access the T.DAT file.

See Also: LIMITxx.BIN

Access: The T.DAT file can be created by using the text editor or, if all the DMS files within the directory are part of the topography, it can be created automatically, along with the LIMITxx.BIN file, by using the LIMITXX3 utility. The CREATELIMITS utility reads the T.DAT file and generates the LIMITxx.BIN file.

Notes: The objective is to create the LIMITxx.BIN file. The two methods to create LIMITXX.BIN are: create T.DAT and read it with the CREATELIMITS utility, or use the LIMITXX3 utility to create both the T.DAT and LIMITxx.BIN files.

Structure: ASCII, sequential, variable length, 16 bytes/record. Only the filename is required; disk and directory are not necessary. One filename is permitted per line.

Filename: **TOPOCTL.FIL**

Directory: MAP$DATA$MAIN:

Purpose: TOPOCTL.FIL is the map database control file. It contains the file names and descriptions of the regions within the map database and the name of the icon control file.

See Also: MAP$DATA$MAIN:*.DMS, MAP$ICONS:ICON.CTL

Access: The map database control file is read at start up of most IBS applications through a call to CONTROL_FILE_READER.

Include: MAP$DEVELOP$INC:MAPGENVAR.INC

Notes: System logical for TOPOCTL.FIL is MAP$CONTROL:

Structure: ASCII, sequential, variable length records. Identifies the file names and descriptions of the regions within the DMS database and the name of the icon control file. The low-detail DMS files specified by the REGION fields of TOPOCTL.FIL (see below) are not currently used by the IBS. They provide a low-detail overview of a database region.

Record 1: Variable record length

```
FORMAT (2I5,1X,A)
```

ICON_FLAG	I*4	Icon flag field. If 1, the icon control file will be specified in record 2.
NREGION	I*4	Number of REGIONS contained in the file (max = 5)
MAIN_DIR	C*20	Top-level directory for the DMS database (usually MAP$DATA$MAIN:)

Record 2: Variable record length. Contains the name of the icon control file, if ICON_FLAG ≠ 0 (see previous record description). If ICON_FLAG = 0, this record is not included in the file.

```
FORMAT (A)
```

ICON_CTL	C*40	Name of icon control file (usually MAP$ICON$CTL:)

Record 3: Variable record length. Contains the file names and descriptions of the regions for the map database. There are NREGION (see above) records of record 3 format in the file.

```
FORMAT (A20,1X,A20)
```

REGION (1,I)	C*20	Name of region file name
REGION (2,I)	C*20	Description of region

Filename: **TOPODESC.FIL**

Directory: MAP$DATA$MAIN:

Purpose: TOPODESC.FIL is the DMS topography description file. It contains the descriptions, logical names of topographic file directories, and topographic codes for the topographic map layers of the IBS.

Access: The topography description file is accessed at start up of most IBS applications through a call to TOPO_FILE_READER.

Include: MAP$DEVELOP$INC:MAPGENVAR.INC

Notes: System logical for TOPODESC.FIL is MAP$TOPOGRAPHY.

Structure: ASCII, sequential, variable length records. Identifies the map topographies available to the IBS.

Record 1: Header (ignored)

Record 2-n: 48 bytes

```
FORMAT (1X,A26,A20,A2)
```

Each record provides information for a single map topography.

TOPO_DESC	C*26	Topography description. If the last character of the field is a *v*, it denotes a volatile topography.
TOPO_LOG_NAME	C*20	Logical name of topography directory.
TOPO_ABRV	C*2	2-character abbreviation code for topography.

Filename: TOWERSEL.DAT

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: Transferred from onpost to offpost. This file tells which weather observation data to use and must accompany the WEATHER.DAT file.

See Also: WEATHER.DAT

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> TOWER SELECTION
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. ONPOSTSIM
4. EVENT

Include:
IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC
IEMIS$INC:WEATHER.INC

Notes: When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, fixed format, 80 bytes/record
This file has up to WS_TS_MAX (7) records, one for each type of tower data.

Record 1-2: File header and description
```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K (K=J+2;J=1,N;1<=N<=WS_TS_MAX):
```
FORMAT (A8,1X,A8,1X,A8,54X)
```

WS_TS_TOWER_TYPE(J)	C*8	Type of data gathered at tower (Keys from D2 input file: TWR_STAB, TWR_WSPD, TWR_WDIR, TWR_TEMP, TWR_ATMP, TWR_CLDH, TWR_HMLR)

WS_TS_TOWER_ID(J)	C*8	Tower ID. In KNOWNPTS, ("TOWER:", 1X, A8), where the A8 is this 8 character ID.
WS_TS_TOWER_CLS(J)	C*8	Cluster ID

Filename: UNITS.DAT

Directory: IEMIS$CHEMS:

Purpose: To provide indexes to UNITSI.DAT data.

Access: 1. CHEMS_(NO)GR

Include: CHEMS$INC:UNITS.INC

Structure: ASCII, Sequential, variable record length

Record 1: Variable length
FORMAT (*)

LENGTH	I*4	Index of length conversion factors in UNITSI.

Record 2: Variable length
FORMAT (*)

LVOLUME	I*4	Index of volume conversion factors in UNITSI.

Record 3: Variable length
FORMAT (*)

LPRESS	I*4	Index of pressure conversion factors in UNITSI.

Record 4: Variable length
FORMAT (*)

LSPEED	I*4	Index of speed conversion factors in UNITSI.

Record 5: Variable length
FORMAT (*)

MASS	I*4	Index of mass conversion factors in UNITSI.

Record 6: Variable length
FORMAT (*)

MASSDOT	I*4	Index of mass rate conversion factors in UNITSI.

Record 7: Variable length
FORMAT (*)

	LHEAT	I*4	Index of heat conversion factors in UNITSI.
Record 8:	Variable length FORMAT (*)		
	LHEATDOT	I*4	Index of heat rate conversion factors in UNITSI.
Record 9:	Variable length FORMAT (*)		
	LHEATFLX	I*4	Index of heat flux conversion factors in UNITSI.
Record 10:	Variable length FORMAT (*)		
	LAREA	I*4	Index of area conversion factors in UNITSI.
Record 11:	Variable length FORMAT (*)		
	LTEMP	I*4	Index of temperature conversion factors in UNITSI.
Record 12:	Variable length FORMAT (*)		
	LCONC	I*4	Index of concentration conversion factors in UNITSI.

Filename: UNITSI.DAT

Directory: IEMIS$CHEMS:

Purpose: To provide conversion factors to change various units of measure to SI system units.

Access: 1. CHEMS_(NO)GR

Include: CHEMS$INC:UNITSI.INC

Structure: ASCII, Sequential, variable record length

Record 1: Variable length
FORMAT (*)

IUNITS	I*4	Number of units records to follow.

Record 2,iunits+1: Variable length
FORMAT (*)

CONV((J,K),K=1,6)	I*4	Conversion factors to SI units from 6 common units.

Filename: WATCH_EVENT.LOG

Directory: OFF$DIR:

Purpose: This file maintains a journal of reports sent from the onpost system to the local IBS node. It contains pointers to detail records contained in WATCH_EVENT.LOG_DETAIL.

See Also: WATCH_EVENT.LOG_DETAIL

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST -> IBS RPT
2. IBS -> MESSAGE BOARD -> CREATE/VIEW COMMON/PRIVATE LOG REPORT

Structure: ASCII, direct access, fixed length, 128 bytes/record

Record 1:

```
FORMAT (I6,A)
```

IREC_1+1	I*4	Pointer to next unused record in file
BLANKS	C*122	Space filler

Record 2-n: 128 bytes

```
FORMAT (4A,2I6,A)
```

TSOURCE	C*20	Function or user that made the log entry
TLDATETIME	C*23	Current date and time (VMS format)
TADATETIME	C*23	Actual date and time of activity
TMSGTYPE	C*25	Primary key
FIRST	I*4	Pointer to first record in associated detail file
LAST	I*4	Pointer to last record in associated detail file
TUNUSED	C*25	Space filler

Filename: WATCH_EVENT.LOG_DETAIL

Directory: OFF$DIR:

Purpose: This file maintains a journal of reports sent from the onpost system to the local IBS node. It contains the detail records referenced by WATCH_EVENT.LOG.

See Also: WATCH_EVENT.LOG

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST -> IBS RPT
2. IBS -> MESSAGE BOARD -> CREATE/VIEW COMMON/PRIVATE LOG REPORT

Notes: Applications can write as may detail records for a given log entry as desired.

Structure: ASCII, direct access, fixed length, 80 bytes/record

Record 1-n: 80 bytes
FORMAT (A)

DETAILS(I)	C*80	Free format text

Filename: **WD.DMS**

Directory: MAP$DATA$WD:

Purpose: This file is the graphical representation of the D2 track created from the latest onpost D2 case.

Access: EVENT creates this file after receiving D2 data from onpost. The track can be displayed through IBS -> SHOW LATEST ONPOST TRACK.

Notes: The track itself is created from the DOSAGEnnn.DAT file, and its attributes (if any) are generated from the SLICEnnn.DAT file.

Structure: DMS format

Filename: WEATHER.DAT

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: This file is sent from onpost to offpost and provides weather observations from a number of weather towers. TOWERSEL.DAT is used to pick the appropriate tower readings.

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> WEATHER
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. ONPOSTSIM
4. EVENT

Include:
IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC
IEMIS$INC:WEATHER.INC

Notes: When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, fixed format, 80 bytes/record
This file has up to WS_WO_MAX (100) records of weather data.

Record 1-2: File header and description
```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K (K=J+2;J=1,N;1<=N<=WS_WO_MAX):
```
FORMAT (A8,1X,A8,1X,A1,1X,A1,F6.1,8F6.0,5X)
```

WS_WO_TOWER_ID(J)	C*8	Weather tower ID. In KNOWNPTS, ("TOWER:", 1X, A8), where the A8 is this 8 character ID.
WS_WO_TOWER_CLS(J)	C*8	Cluster ID
WS_WO_STATUS(J)	C*1	Tower status (Y=ONLINE, N=OFFLINE)

WS_WO_STAB(J)	C*1	Stability (A, B, C, D, E, F, U, S, W, X)
WS_WO_WSPD(J)	R*4	Wind speed (m/sec)
WS_WO_WDIR(J)	R*4	Wind direction (degrees East of North)
WS_WO_TEMP(J)	R*4	Temperature (C)
WS_WO_ATMP(J)	R*4	Atmospheric pressure (mm Hg)
WS_WO_CLDH(J)	R*4	Cloud Height (ft)
WS_WO_HMLR(J)	R*4	Height of Mixing Layer (m)
WS_WO_HUM(J)	R*4	Humidity (%)
WS_WO_SIGMA(J)	R*4	Sigma (degrees)
WS_WO_VOLTS(J)	R*4	Battery level (volts)

Filename: WI.DMS

Directory: MAP$DATA$WI:

Purpose: This file is the graphical representation of the location of the weather stations.

See Also: TOWERSEL.DAT, WEATHER.DAT

Access:
1. Accessed by most of the graphics programs in IBS
2. EVENT
3. ONPOSTSIM

Notes: The Event program updates this file when a new WEATHER.DAT file is received from the onpost.

Structure: DMS Format

Filename: WORK_PLAN.DAT

Directory: OFF$DIR:

Purpose: This file contains information on the onpost schedule of activities associated with agent disposal and processing.

See Also: WORKPLAN.DAT

Access:
1. IBS -> STATUS BOARD -> VIEW ONPOST ACTIVITY
2. IBS -> STATUS BOARD -> SELECT ONPOST ACTIVITY

Include: IEMIS$INC:WP.INC

Notes: This file can be generated from the onpost WORKPLAN.DAT report by selecting a single activity. Current weather data can also be used.

Structure: ASCII, sequential, variable record length

Record 1: `FORMAT (3(A23),I6)`

WP_DT	C*23	Date and time info received by IBS (VMS format)
WP_DT_B	C*23	Date and time activity to start
WP_DT_E	C*23	Date and time activity to end
WP_POP	I*4	Team Population

Record 2: `FORMAT (2F12.7,3F5.1,A1)`

WP_LON	R*8	Longitude
WP_LAT	R*8	Latitude
WP_WSPD	R*4	Current wind speed (meters per second)
WP_WDIR	R*4	Current wind direction (degrees clockwise from north)
WP_TEMP	R*4	Current temperature (degrees Celsius)
WP_STAB	C*1	Current stability class (A-F, U, S, or W)

Record 3: `FORMAT (A3,A2,A12,A1,A3,A18)`

WP_MUN	C*3	Munition involved in activity
WP_AGN	C*2	Chemical agent involved in activity
WP_MCE	C*12	MCE local id
WP_MCET	C*1	MCE Type
WP_D2	C*3	D2 case number
WP_AUTH	C*18	Who authorized plan

Record 4: `FORMAT (A80)`

	WP_DESC	C*80	Description
Record 5-9:	FORMAT (A80)		
	WP_PAR(J), J=1,5	C*80	Protective Action Recommendation
Record 10-12:	FORMAT (A80)		
	WP_COM(J), J=1,3	C*80	Comments

Filename: WORKPLAN.DAT

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:

Purpose: Transferred from onpost to offpost. This file provides a schedule of anticipated chemical handling activities at the depot.

See Also: WORK_PLAN.DAT

Access:
1. IBS -> MESSAGE BOARD -> VIEW ONPOST TO IBS REPORT -> WORK PLAN
2. IBS -> MESSAGE BOARD -> VIEW/CREATE COMMON/PRIVATE LOG REPORT
3. IBS -> STATUS BOARD -> SELECT/VIEW ONPOST ACTIVITY
4. ONPOSTSIM
5. EVENT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes: The onpost Work Plan (WORKPLAN.DAT) is used to define the offpost Work Plan file (WORK_PLAN.DAT).

When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

Structure: ASCII, sequential, variable format, <= 80 bytes/record. This file has up to WS_WP_MAX (10) sections describing onpost activities, each beginning with 2 control records with date / location / population / Local ID/ munition / agent information, followed by a description record, followed by WS_WP_PAR_MAX (5) protective action recommendation records. In other words, each activity requires 8 records: 2 control records, a description record, and 5 PAR records.

Record 1-2: File header and description

```
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)
```

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K1 (K1=8*J-5;J=1,N;1<=N<=WS_WP_MAX):

```
FORMAT (2A23,2F12.7,10X)
```

WS_WP_DT_1(J)	C*23	Start time (dd-mmm-yyyy hh:mm:ss.hh)
WS_WP_DT_2(J)	C*23	End time (dd-mmm-yyyy hh:mm:ss.hh)
WS_WP_LON(J)	R*8	Longitude
WS_WP_LAT(J)	R*8	Latitude

Record K2 (K2=8*J-4;J=1,N;1<=N<=WS_WP_MAX):

```
FORMAT (I5,1X,A1,1X,A12,1X,A3,1X,A2,1X,A3,49X)
```

WS_WP_POP(J)	I*4	Team population
WS_WP_MCET(J)	C*1	MCE type: Y=Worst MCE, N=Normal MCE
WS_WP_MCE(J)	C*12	Local ID
WS_WP_MUN(J)	C*3	Munition being handled during activity
WS_WP_AGN(J)	C*2	Agent being handled during activity
WS_WP_D2(J)	C*3	Associated onpost D2 case number

Record K3 (K3=8*J-3;J=1,N;1<=N<=WS_WP_MAX):

```
FORMAT (A80)
```

WS_WP_DESC(J)	C*80	Description of activity

Record K4,K4+4 (K4=8*J-2;J=1,N;I=1,WS_WP_PAR_MAX;1<=N<=WS_WP_MAX):

```
FORMAT (A80)
```

WS_WP_PAR(I,J)	C*80	Protective action recommendation

Filename: XFERLIST.DAT

Directory: ONSITE$DIR:, OFF$FROMDIR:, IEMIS$SIMF:, OFF$SENDDIR:

Purpose: Transferred from onpost to offpost and from offpost to onpost. This file notifies the receiving system that a new data package has arrived from the sending system, and identifies the contents of the package. This file will be a part of every transfer between EMIS and IBS. The types of information that may be transferred are listed in the table below.

Access:
1. IBS -> MESSAGE BOARD -> CREATE IBS TO ONPOST REPORT
2. IBS -> MESSAGE BOARD -> RESEND IBS TO ONPOST REPORT
3. ONPOSTSIM
4. EVENT

Include: IEMIS$INC:HEADER.INC
IEMIS$INC:WATCHSIM.INC

Notes: When the EVENT program finds this file in ONSITE$DIR:, it processes it and moves it to OFF$FROMDIR: for use by other applications.

The following notes apply to the accompanying table:

- An .ATT file always accompanies a .DLG file.
- Report Type is also a field in the file, used to identify the type of report. Report Type determines how the file gets processed, not the filename.
- Off/On tells whether the file is sent from offpost, onpost, or both.
- Text/Graphics/# tells whether the report is a single textual file or if it is accompanied by one or more graphical DLG files.
- Filename tells the name of the file. Lower case files names indicate that the actual filename will be something else. Upper case filenames will be exactly so.
- Some of the offpost files are sent long before an event: EVR, KP, KPL, MCE, PEVR, PIMR, and RCD. PEVR is also sent during an event. The remaining offpost files are sent during an event. Files which get sent both directions can be sent at any time.

Report Type	Off/On	Text/Graphic/#File	
GENERIC TEXT DATA	Y/Y	Y/N	generic.DAT
GENERIC GRAPHIC DATA	Y/Y	N/Y	generic.DLG
KEYED DATA	Y/Y	Y/N	KEY.DAT
MAIL UPDATE	Y/Y	Y/N	MLUPDATE.DAT
XFERLIST	Y/Y	Y/N	XFERLIST.DAT
D2 INPUT	Y/Y	Y/N	D2INPnnn.DAT
D2 LOG	Y/Y	Y/N	D2LOGnnn.DAT
D2 DOSAGE	Y/Y	Y/N	DOSAGnnn.DAT
D2 SLICE	Y/Y	Y/N	SLICEnnn.DAT
BASE POPULATION	N/Y	Y/N	BASE_POP.DAT
CALL LIST	N/Y	Y/N	CALLLIST.DAT
CASUALTY SUMMARY	N/Y	Y/N	CAS_SUM.DAT
ENVIRONMENT	N/Y	Y/N	ENVIRON.DAT
KNOWN POINTS	N/Y	Y/N	KNOWNPTS.DAT
MCE DIRECTORY	N/Y	Y/N	MCE_DIR.DAT
NOTIFICATION	N/Y	Y/N	NOTIFY.DAT
SIGNIFICANT EVENTS	N/Y	Y/N	SIGEVENT.DAT
TOWER SELECTION	N/Y	Y/N	TOWERSEL.DAT
WEATHER	N/Y	Y/N	WEATHER.DAT
WORK PLAN	N/Y	Y/N	WORKPLAN.DAT
OFFPOST CASUALTY REPORT	Y/N	Y/N	OFF_CAS.DAT
OFFPOST EVACUATION ROUTES	Y/N	Y/Y/n	OFF_EVR.DAT, .dlg
OFFPOST KNOWN POINTS	Y/N	Y/Y/1	OFF_KP.DAT, .DLG
OFFPOST KNOWN POLYGONS	Y/N	Y/Y/n	OFF_KPL.DAT, .dlg
OFFPOST MCE DIRECTORY	Y/N	Y/N	OFF_MCE.DAT
PROTECTIVE ACTION REPORT	Y/N	Y/Y/n	OFF_PA.DAT, .dlg
PROPOSED EVACUATION ROUTES		Y/N	Y/Y/nOFF_PEVR.DAT,
.dlg			
PROPOSED MILITARY ROUTES	Y/N	Y/Y/1	OFF_PIMR.DAT, .dlg
RECEPTION CTR DIRECTORY	Y/N	Y/N	OFF_RCD.DAT
RECEPTION CTR REPORT	Y/N	Y/N	OFF_RCR.DAT
OFFPOST SHELTER REPORT	Y/N	Y/N	OFF_SHLT.DAT
OFFPOST STATUS REPORT	Y/N	Y/N	OFF_STAT.DAT
TRAFFIC CONTROL REPORT	Y/N	Y/N	OFF_TC.DAT

Structure: ASCII, sequential, fixed format, 80 bytes/record. For each file being transferred, there will be a single record identifying the type of file, the type of report, and the file- name. Up to WS_XF_MAX (100) files can be transferred at one time.

Record 1-2: File header and description
FORMAT (A4,A4,1X,A4,1X,A12,1X,A12,1X,A23,A12,A5,/,A80)

HEAD_MODE	C*4	Mode (OPER=Operational, EXER=Exercise)
HEAD_FAZE	C*4	Phase (PREP=Planning, EVNT=Response)
HEAD_SITE	C*4	Site name (offpost or onpost)
HEAD_NODE	C*12	Node on which file originated
HEAD_USER	C*12	Name of user who created file
HEAD_DATE	C*23	Date file created
HEAD_FILE	C*12	Filename
HEAD_VERS	C*5	Format version # (V2.00)
HEAD_DESC	C*80	Description of file contents

Record K1 (K1=J+2;J=1,N;1<=N<=WS_XF_MAX):
FORMAT (A1,1X,A32,A40,6X)

WS_XF_FILE_TYPE(J)	C*1	File type (T=TEXT or G=GRAPHIC)
WS_XF_REPORT_TYPE(J)	C*32	Type of report (see table above)
WS_XF_FILE_NAME(J)	C*40	Name of file (see table above)

Appendix B
IBS Icon Dictionary (Icon Control File)

This appendix lists the attribute codes, names, graphic file names, and types of the standard graphic icons used in the IBS and specified in the IBS icon dictionary.

Icon Types. In the following table, the icon type is specified as **F** (facility icon), **R** (resource icon), or no type (general use icon).

Major Attribute Code	Minor Attribute Code	Description	File Name	Type
1	0410	DAM	MAP$ICONS:DAM.DMS	F
3	0120	RECREATION AREA	MAP$ICONS:F_RECAREA.DMS	F
5	1100	TRAF CTRL PT	MAP$ICONS:F_TRAFFIC_A.DMS	F
5	1101	TRAF CTRL PT (INACTIVE)	MAP$ICONS:F_TRAFFIC_I.DMS	F
7	0080	POWER STATION	MAP$ICONS:POWSTA.DMS	F
7	0100	HYDROELECTRIC PLANT	MAP$ICONS:HYDRO.DMS	F
7	0110	CIVILIAN AIRPORT	MAP$ICONS:AIRPORT.DMS	F
7	0111	AIRPORT	MAP$ICONS:AIRPORT.DMS	F
7	0112	CIVILIAN AIR HUB	MAP$ICONS:AIRHUB.DMS	F
7	0115	MILITARY AIRPORT	MAP$ICONS:MAIRPORT.DMS	F
7	0250	TRANSPORTATION FACILITY	MAP$ICONS:F_TRANSPORT.DMS	F
8	0010	CHURCH	MAP$ICONS:CHURCH.DMS	F
8	0020	SCHOOL	MAP$ICONS:SCHOOL.DMS	F
8	0030	HOSPITAL	MAP$ICONS:HOSPITAL_A.DMS	F
8	0031	V.M.A.C. HOSPITAL	MAP$ICONS:VMAC.DMS	F
8	0032	HOSPITAL (INACTIVE)	MAP$ICONS:HOSPITAL_I.DMS	F
8	0181	PUBLIC UTILITY	MAP$ICONS:F_PUB_UTL.DMS	F
8	0190	INDUSTRIAL PARK	MAP$ICONS:INDUST.DMS	F
8	0195	PRIVATE BUSINESS	MAP$ICONS:F_PRIV_BIZ.DMS	F
8	0200	EMERGENCY EQUIPMENT	MAP$ICONS:EQUIP.DMS	
8	0205	NUCLEAR PLANT	MAP$ICONS:NUCPLANT.DMS	F
8	0210	HAZMAT SITE	MAP$ICONS:HAZMAT.DMS	
8	0530	SHELTER	MAP$ICONS:SHELTER_A.DMS	F
8	0536	SHELTER (INACTIVE)	MAP$ICONS:SHELTER_I.DMS	F
8	0560	POLICE STATION	MAP$ICONS:POLICE.DMS	F
8	0570	FIRE STATION	MAP$ICONS:FIRE.DMS	F
8	0585	GOVERNMENT BUILDING	MAP$ICONS:F_GOVT_BLDG.DMS	F
8	0620	E.O.C.	MAP$ICONS:EOC_A.DMS	F
8	0621	E.O.C. (INACTIVE)	MAP$ICONS:EOC_I.DMS	F
8	0690	FUEL TANK FARM	MAP$ICONS:FUEL.DMS	F
8	0710	WATER SUPPLY	MAP$ICONS:WATER.DMS	F
8	0770	COMMUNICATIONS FACILITY	MAP$ICONS:COMMFAC.DMS	F
8	1400	SIREN - ROTATING	MAP$ICONS:SIRENR.DMS	F

Major Attribute Code	Minor Attribute Code	Description	File Name	Type
8	1401	SIREN - NONROTATING	MAP$ICONS:SIRENNR.DMS	F
9	0050	SPECIAL POPULATION	MAP$ICONS:F_SPECPOP.DMS	F
9	0430	OTHER CONTROL POINT	MAP$ICONS:OCONTROL.DMS	
9	0560	EVACUATION SHELTER	MAP$ICONS:EVAC_SHELTER.DMS	
9	0561	RECEPTION CTR	MAP$ICONS:F_RECEPCTR_A.DMS	F
9	0565	RECEPTION CTR (INACTIVE)	MAP$ICONS:F_RECEPCTR_I.DMS	F
9	0566	DECON CENTER	MAP$ICONS:DECON_A.DMS	F
9	0567	DECON CENTER (INACTIVE)	MAP$ICONS:DECON_I.DMS	F
9	0570	MORGUE	MAP$ICONS:MORGUE_A.DMS	F
9	0571	MORGUE (INACTIVE)	MAP$ICONS:MORGUE_I.DMS	F
9	0700	HURRICANE	MAP$ICONS:HURRICANE.DMS	
9	0701	TROPICAL STORM	MAP$ICONS:TROPSTORM.DMS	
9	0702	TSUNAMI	MAP$ICONS:TSUNAMI.DMS	
9	0703	EARTHQUAKE	MAP$ICONS:QUAKE.DMS	
9	0704	FIRE	MAP$ICONS:FIRE3.DMS	
9	0705	BLIZZARD	MAP$ICONS:BLIZZARD.DMS	
9	0706	TORNADO	MAP$ICONS:TORNADO.DMS	
9	0707	FLOOD	MAP$ICONS:FLOOD.DMS	
9	0708	CITY	MAP$ICONS:CITY.DMS	
9	0712	MOBILE MEDICAL UNIT	MAP$ICONS:F_MOBMED.DMS	F
9	0715	PRIVATE RESIDENCE	MAP$ICONS:F_PRIV_RES.DMS	F
9	0812	ENVIRONMENT REPORT	MAP$ICONS:ENVIRONMENT.DMS	
9	0813	WEATHER REPORT	MAP$ICONS:WEATHER.DMS	
9	0900	NAPB TARGET	MAP$ICONS:TARGET.DMS	
9	1006	REFINERY	MAP$ICONS:REFINERY.DMS	F
9	1007	COAL MINE	MAP$ICONS:COAL.DMS	F
9	1008	COKE PRODUCER	MAP$ICONS:COKE.DMS	F
9	1009	NAT GAS PRESSURE PLANT	MAP$ICONS:NGPP.DMS	F
9	1010	NAT GAS UNDERGROUND ST	MAP$ICONS:NGUS.DMS	F
9	1011	PETROL IMPORT FACILITY	MAP$ICONS:PET.DMS	F
9	1012	PORT FACILITY	MAP$ICONS:PORT.DMS	F
9	1013	GRAIN STORAGE	MAP$ICONS:GRAIN.DMS	F
9	1014	FLOUR MILLS	MAP$ICONS:FMILLS.DMS	F
9	1015	FOOD DISTRIBUTION	MAP$ICONS:F_FOOD_DIST.DMS	F
9	1016	POULTRY PRODUCTION	MAP$ICONS:POLT.DMS	F
9	1017	AGRICULTURE E.O.C.	MAP$ICONS:AEOC.DMS	F
9	1018	EGG PRODUCERS	MAP$ICONS:EGGS.DMS	F
9	1019	FOOD PRODUCTION	MAP$ICONS:F_FOOD_PROD.DMS	F
13	0001	1ST AID	MAP$ICONS:R_1ST_AID_.DMS	R
13	0002	BARRICAD	MAP$ICONS:R_BARRICAD.DMS	R
13	0003	CARS	MAP$ICONS:R_CARS____.DMS	R
13	0004	COMM EQ	MAP$ICONS:R_COMM_EQ_.DMS	R
13	0005	CONST EQ	MAP$ICONS:R_CONST_EQ.DMS	R
13	0006	DRY GOOD	MAP$ICONS:R_DRY_GOOD.DMS	R
13	0007	EMERG EQ	MAP$ICONS:R_EMERG_EQ.DMS	R

Major Attribute Code	Minor Attribute Code	Description	File Name	Type
13	0008	FOOD	MAP$ICONS:R_FOOD____.DMS	R
13	0009	GENERATR	MAP$ICONS:R_GENERATR.DMS	R
13	0010	HAND TL	MAP$ICONS:R_HAND_TL_.DMS	R
13	0011	HEAVY EQ	MAP$ICONS:R_HEAVY_EQ.DMS	R
13	0012	LIGHTS	MAP$ICONS:R_LIGHTS__.DMS	R
13	0013	MEDIC EQ	MAP$ICONS:R_MEDIC_EQ.DMS	R
13	0014	OFFICE M	MAP$ICONS:R_OFFICE_M.DMS	R
13	0015	PAVE EQ	MAP$ICONS:R_PAVE_EQ_.DMS	R
13	0016	PEOPLE	MAP$ICONS:R_PEOPLE__.DMS	R
13	0017	POWER TL	MAP$ICONS:R_POWER_TL.DMS	R
13	0018	PREFAB B	MAP$ICONS:R_PREFAB_B.DMS	R
13	0019	RESCUE S	MAP$ICONS:R_RESCUE_S.DMS	R
13	0020	SCALES	MAP$ICONS:R_SCALES__.DMS	R
13	0021	SIREN-NR	MAP$ICONS:R_SIREN_NR.DMS	R
13	0022	SIREN-R	MAP$ICONS:R_SIREN_R_.DMS	R
13	0023	TRANSPRT	MAP$ICONS:R_TRANSPRT.DMS	R
13	0024	TRUCKS	MAP$ICONS:R_TRUCKS__.DMS	R
13	0025	WATER	MAP$ICONS:R_WATER___.DMS	R
13	0026	WEAPONS	MAP$ICONS:R_WEAPONS_.DMS	R

Appendix C
DMS and TXT File Formats

This appendix defines the file structure of DMS and DMS text files. DMS files contain line or irregular point data. Longitude-latitude (long-lat) values are stored for each data item. The file header contains information about the long-lat extents of the file, but does not try to define the type of data. Any data item can be further characterized by associated attributes. Attributes allow data entries to be classified and defined on a general, as well as a specific, basis. For example, through use of attribute codes a stream could be classified as perennial with an average flow rate of 15,500 cubic feet per second and associated with the text name, Lower Salmon River. Also, as some data items may be best represented by predefined icons, this structure supports such representation by allowing attribute codes to be associated with particular figures. For example, roads, streams, hospitals, power lines, political boundaries, and airports may be efficiently represented through these DMS constructs.

C.1 Structure of DMS Files

Line and irregular point data (such as streams, roads, hospitals, and airports) are digitized into DMS format files. This structure supports node, line, and polygon constructs.

The DMS file is a direct access, binary file with a fixed record length of 8192 bytes. Each DMS file has a header that contains the name of an associated text file, if it exists, and information defining the long-lat extents of the data within the file. Following the header are data entries whose structures vary slightly, depending on the type of the data item.

Data values are stored as long-lat offsets from the file reference point of the file. These offset values are scaled to allow the maximum limits of the file to be represented by an Integer*4 (32-bit) offset value. Each file scale factor is contained in the file header as a Real*8 (64-bit) number. A data value may be converted to a long-lat point by multiplying it by the scale factor and then adding the file origin to the result.

The scale factor is calculated by dividing the maximum of longitude and latitude extents by the number of steps to be allowed from the origin. This scheme of storage allows a high degree of precision to be achieved by simply reducing the area contained in a given file. Consistency of data is also maintained as all data is stored in units of long-lat, thus providing a convenient means of displaying files in conjunction with one another.

For most mapping applications, the multiplication factor may be determined from an extent of 360 degrees and a step size based on the maximum Integer*4 word (step size = ± 2.0E9, scale factor = 360/4.0E9 = 9.0E-8). With this scale factor, points could be stored with a precision of ± 0.5 inch for the earth.

C.1.1 DMS File Description Header

A DMS file header consists of the fields shown in Table C.1.

Space for all fields must be reserved; however, actual data need not be supplied for fields marked as optional. A total of 28 longwords (112 bytes) will always be reserved for the file header.

Table C.1. DMS File Header

Field #	Data Type	Data Description
1	C*4	Type of data file (always "DMS ")
2	I*4	Scale of the map from which the data was digitized (optional)
3	R*8	Longitude of the file origin
4	R*8	Latitude of the file origin
5	R*8	Western-most longitude of data contained in the file
6	R*8	Eastern-most longitude of data contained in the file
7	R*8	Southern-most latitude of data contained in the file
8	R*8	Northern-most latitude of data contained in the file
9	R*8	Scale factor used for converting data points to long-lat values
10	C*16	Filename of file containing character data referenced by text attributes within this file (optional)
11	R*8	Date on which data may be considered valid (system date-time format)
12	R*8	Date when file was created
13	R*8	Date when file was last modified
14	C*8	Dummy

C.1.2 DMS File Data Types

Storage of data items begins directly after the file header. Currently, four different types of data are currently supported: line, point, polygon, and complex polygon. Three separate data structures are employed to represent these four data types.

Point Data. Objects that can be represented on maps by a single long-lat location (such as hospitals, airports, schools, and food processing plants) are expressed in DMS files as point data structures. The long-lat pair contained in the point data element is assumed to be the center of the object.

Line Data. Streams, roads, railroads, and power lines are examples of geographic data that is expressed digitally as ordered sets of points describing the associated linear feature on the map. The line data structure is used to represent these ordered sets of points within DMS files. Lines may intersect other lines or themselves with no limitations imposed on the number of intersections. The maximum number of points associated with any single line segment is 32,767.

Polygon Data. An area enclosed by a single geometric boundary that does not cross itself is represented by the polygon data structure. Simple area boundaries, such as state and county lines or lakes without islands, are expressed as polygons. To ensure closure, the first point in a polygon data structure must be the same as the last point. Thus, simple polygons can be plotted by drawing a line from the first point to the second, second to third, and concluding with the last line drawn back to the first point.

Complex Polygon Data. Complex polygons can be used to define areas that are composed of multiple simple regions, either inclusive or exclusive. For example, a lake containing four islands could be expressed as five simple polygons. The boundary of the lake would be an inclusive region with each of the islands represented by an exclusive polygon. The complex polygon data structure allows multiple simple regions to be identified with a particular area. Each simple polygon contains a flag indicating whether the region is inclusive or exclusive to the complex area.

C.1.3 DMS File Data Structures

Data Structure #1

Points or lines and polygons having fewer than 10 points are represented by Data Structure #1, detailed in Table C.2.

Table C.2. Data Structure #1 (Points, Lines and Polygons with <10 Points)

Field #	Data Type	Data Description
1	I*4	Descriptor mask containing pen number, line type, number of attributes, plot flag, and number of data points
2	I*4	Attributes (maximum of 15)
3	I*4	Data points (X values followed by Y values - maximum 32,767 pairs)

Field #1 is a packed Integer*4 (32-bit) word containing five different items. The number of data points in this entry is contained in bits 0-15. Bits 16-18 represent the plot flag. Bits 19-21 hold a value that indicates the type of line to be drawn for line or polygon data. The pen number (color and blinking) used for displaying this entry is contained in bits 22-27. The number of attributes associated with this entry is contained in bits 28-31.

Written in octal (base eight) numeration, this field is divided as follows:

Number of Attributes	Pen Number	Line Type	Plot Flag	Data Points
17	77	7	7	177777

where a 7 represents three bits and a 1 represents one bit.

The plot flag item is used to distinguish between the various data types. Values for plot flag are:

0 = Line
1 = Point
2 = Polygon
3 = Complex Polygon

The line type item indicates whether to draw solid, dotted, wide, or dashed line. Values for the line type are:

0 = Solid Line
1 = Wide Line
2 = Dashed Line
3 = Dotted Line

Field #2 contains attribute codes, one per Integer*4 word. The number of attributes present is specified by bits 28-31 of Field #1.

Written in octal (base eight) numeration, this field is divided as follows:

Major Attribute Code	Minor Attribute Code	Parameter Value
17	7777	177777

where a 7 represents three bits and a 1 represents one bit.

Field #3 contains X and Y data values for each point in the entry. The number of data points for an entry is specified by bits 0-15 of Field #1. For example, if the entry contained 9 data pairs, they would appear as 18 Integer*4 numbers: X1, X2, ..., X9, Y1, Y2, ..., Y9. Longitude and latitude values are calculated from X and Y data values as follows:

Longitude = (X value) (Header Field #9) + (Header Field #3)

Latitude = (Y value) (Header Field #9) + (Header Field #4)

Data Structure #2

Lines and polygons having 10 or more data pairs are represented by Data Structure #2, detailed in Table C.3.

Table C.3. Data Structure #2 (Lines and Polygons with ≥10 Points)

Field #	Data Type	Data Description
1	I*4	Descriptor (same as for Data Structure #1)
2	I*4	Attributes (same as for Data Structure #1)
3	I*4	Minimum and maximum extents, four I*4 entries
4	I*4	Data points (same as for Data Structure #1, Field #3)

Field #3 is the only field in this data structure that is different from those in Data Structure #1. It contains the extents (minimum and maximum values) of an entry. For example, if this structure defines a line with 23 data pairs, this field would contain the minimum X, maximum X, minimum Y, and maximum Y (stored in that order) of the 23 points. These limits are converted to long-lat values as before:

Longitude = (X value) (Header Field #9) + (Header Field #3)
Latitude = (Y value) (Header Field #9) + (Header Field #4)

Data Structure #3

Complex polygons are represented by Data Structure #3, detailed in Table C.4.

Table C.4. Data Structure #3 (Complex Polygons)

Field #	Data Type	Data Description
1	I*4	Descriptor (same as for Data Structure #1)
2	I*4	Attributes (same as for Data Structure #1)
3	I*4	Extents (same as for Data Structure #2)
4	I*2	Number of polygons, starting point index for each polygon
5	I*4	Data points (same as for Data Structure #1, Field #3)

Field #4 describes the makeup of the complex polygon. This field contains the following information: the number of simple polygons composing the complex area, identification of a particular polygon as inclusive or exclusive, and the starting point of each polygon. All values are expressed as Integer*2 data types, and the field is always rounded up to include an even number of entries. For example, if the number of polygons happened to be four, this field would be structured as follows:

> # polygons, starting index of Poly #1, starting index of Poly #2, starting index in Poly #3, starting index in Poly #4, I*2 Dummy

for a total of three Integer*4 words.

The first entry represents the number of simple polygons composing the complex polygon (four, in this example). Entries 2 through 5 indicate the starting point of each polygon within the data arrays and whether the polygon is inclusive or exclusive. Polygons are considered inclusive if the entry has a positive value, and exclusive otherwise.

For example if entry #2 were -1, entry #3 were 6, and entry #4 were 10, the first polygon would be defined by the first five data points, and the interior region of the polygon would be excluded from the complex area. The second polygon would be defined by data points 6 through 9, and the interior region would be included in the complex area.

C.2 Structure of DMS Text Files

Any DMS data item may use attribute codes to reference a text descriptor. This reference is accomplished by including the name of a text data file within the DMS file header and with pointers to particular text strings within the attribute codes.

Attributes referencing text descriptors use all three fields available for attribute codes (major code, minor code, and parameter value. The major code defines the attribute as a pointer to a text string. The parameter value indicates which block contains the beginning of the text descriptor, and the minor code references the location within the block where the descriptor actually starts.

Text files are binary direct access files with a 8192-byte (4096 Integer*2) record size. Text descriptors within these files are required to start on Integer*2 boundaries. Hence, attribute minor codes reference the Integer*2 word that is the start of the text descriptor. The convention is to call the first word of a block word #0 and the last word, #4095.

Attribute parameter values reference 4096 Integer*2 word blocks, with the first block being identified as block #1. A maximum of 32767 blocks may be contained in any one text file.

The first four (4) words of a text file make up the text file header. It defines the record and word for the next available text descriptor. The format of a text file header is shown in Table C.5.

Table C.5. DMS Text File Descriptor Format

Field #	Data Type	Data Description
1	I*2	Next available record to add new text
2	I*2	Next available word to add new text

Text descriptors are packed into these direct access records with the first descriptor beginning in block #1, word #4. Parameters indicating the number of lines of text, character size, rotation, and associated decluttering levels are all contained in the text descriptor. The format of a text descriptor is shown in Table C.6.

Table C.6. DMS Text File Descriptor Format

Field #	Data Type	Data Description
1	I*2	Number of lines of text (MAX = 15)
2	R*4	Character size
3	R*4	Character rotation
4	I*2	Character font
5	R*8	Character scale factor
6	I*2 (N)	Number of bytes for each line
7	C*(*)	Text data

Field #1: The absolute value of this field indicates the number of lines of text information displayed. If the actual value is less than 0, the text is centered.

Field #2: The absolute value of this field indicates the size of characters used when displaying the text string. If the actual value is less than 0, the text is a constant size text.

Field #3 indicates the angle of rotation to which the string is aligned for display.

Field #4 indicates which character font is used.

Field #5 contains the scale factor to be used when displaying the text.

Field #6 contains the actual number of bytes for each line of text. For example, if field #1 had a value of three, an entry in field #5 would be present for each of the three lines of text. If field #5 contained the entries 24, 15, and 19, the first line of text would include 24 characters of text, the second line would include characters 25 through 39, and the third line would include characters 40 through 58.

Field #7 contains the text data. This field will always have an even number of bytes. If the actual number of bytes for the string is odd, one blank character will be appended to this field, so its length is rounded to the nearest even number. The maximum number of characters is 300.

Reader Comment Form

Although PNL staff did their best to ensure the accuracy of this document, it is certain that a document of this size and complexity will have errors. When you wish to report an error or provide a comment, use the attached facsimile form or send E-mail to BM_WOOD@PNL.GOV.

IBS Project
Battelle
Pacific Northwest Laboratory
P.O. Box 999
Richland, WA 99352

Date:

Number of Pages:

From

Name:

Company:

Phone:

Fax:

To

Name: Blanche Wood

Company: Battelle - PNL

Phone: (509) 375-2615

Fax: (509) 375-3641

Message

DISTRIBUTION

No. of
Copies

OFFSITE

2 DOE/Office of Scientific and
Technical Information

Federal Emergancy Management
Agency
Attn: Ed Corvi
500 C Street SW
Washington, DC 20472

Col. Lawrence E. Skelly
United States Army Nuclear and
Chemical Agency
7150 Heller Loop, Suite 101
Springfield, VA 22159

J. Thompson
Operations Support Directorate
HQ FEMA
5321 Riggs Road
Gaithersburg, MD 20882

Madhu Beriwal
Innovative Emergency Management
7423 Picardy Ave, Suite E
Baton Rouge, LA 70808-4362

Rich Kellner
Applied Computing Systems
120 Longview Dr.
Los Alamos, NM 87544

John Bower
Bower Software Services
1114 E. 19th Ave.
Spokane, WA 99203

No. of
Copies

ONSITE

DOE Richland Operations Office

R. B. Goranson

24 Pacific Northwest Laboratory

B. M. Wood (10)
M. J. Burford
T. R. Downing
N. D. Foote
S. M. Johnson
S. W. Matsumoto
M. C. Moise
S. L. Savelli
J. R. Williams
C. Winters (2)
Publishing Coordination
Technical Report Files (3)