List of Appendices

Appendix A
EBS Survey Data Base
Data Dictionary

Environmental Baseline Survey Data Base

Data Dictionary

Prepared For:
Aeronautical Systems Center
Wright-Patterson Air Force Base, Ohio
and
Air Force Center for Environmental Excellence
Brooks Air Force Base, Texas

Prepared By:

Earth Tech, Inc.
1420 King Street, Suite 600
Alexandria, Virginia 22314

November 1999

Table of Contents

List of Tables

Section 1.0 Introduction

The Environmental Baseline Survey (EBS) Data Base was developed for U.S. Air Force, Aeronautical Systems Center (ASC) from requirements identified by the Acquisition Environmental Management office. It contains a package of attribute data bases—hazwi0.tab, asbesti0.tab, leadi0.tab, pcbi0.tab, radoni0.tab, medi0.tab, ordi0.tab, bldgi0.tab, and facnoi0.tab—that can be used with MapInfo and MapInfo Data Processing System, Version 2.0 to produce condition of property maps, ad hoc graphics, and reports.

The EBS Data Base Data Dictionary contains guidelines on the data entry into the EBS Data Base. The mapping process requires complete and accurate data bases to ensure that environmental factors are depicted correctly on the maps generated for the EBS reports.

This document contains enough information for the users to update and accurately rebaseline any EBS data base that has been provided ASC. Section 2.0 provides a data base overview stating the general nature of the data base, a history of its development, use, and maintenance, and general guidelines for completing the data entry process. Section 3.0 provides data entry guidelines for creating the data base for environmental factors affecting property categorization. Section 4.0 provides guidelines on creating the data bases for disclosure factors.

Section 2.0 Overview

2.1 Data Base Overview

The EBS Data Base was developed in 1995 for ASC so that EBSs for airplane and missile plants could be automated, manipulated, and ultimately uploaded into the existing Technical Information System where data are stored. This version of the data dictionary modifies definitions previously in use, based on amendments to the Comprehensive Environmental Restoration, Compensation, and Liability Act (CERCLA) Section 120(h)(4)(A) and changes in DOD policy.

The EBS Data Base provides attribute data that describe, in tabular form, the environmental condition of property at the 11 airplane and missile plants owned by the Air Force. The nine tables created for each plant are called hazwi0.tab, asbesti0.tab, leadi0.tab, pcbi0.tab, radoni0.tab, medi0.tab, ordi0.tab, bldgi0.tab, and facnoi0.tab. Only the attribute data in hazwi0 are linked with spatial data using MapInfo and the EBS Data Processing System Version 2.0 to create a condition of property map. All of the data tables can be used to produce ad hoc graphics and reports. Existing data tables do not need to be modified, as a result of the changes in this document. However, the user should run the MapInfo EBS Processing System Version 2.0 to ensure that property category is correct.

The EBS Data Base design is identical for each of the 11 Air Force plants. All the data were entered using this data dictionary. Since the completion of the EBSs, additional data have become available, the "shelf-life" of the EBSs has expired, or regulatory requirements have changed indicating that the original surveys require updating. The EBS Data Base updates should use the guidelines in this dictionary to ensure valid entries. There are no auto-fill fields in the Data Base, and no tracking when changes are made. Therefore, it is important that the project manager for each plant be notified when access is given to the data, and when changes are made.

2.2 EBS Process Overview

The colors and symbols that appear on the oversized maps (Plates 3-1 and 3-2) in the EBS reports are generated by data that have been entered into the EBS Data Base. The data in the data bases can be grouped and sorted to create the colors and symbols on the maps as well as the summary reports in each section of the EBS report. These guidelines were written to assist the data base task managers who are responsible for making sure that complete and adequate data are entered into the data bases to create accurate maps and reports.

There are 15 environmental factors that are evaluated in the EBS reports. Data must be entered for each of the factors that are pertinent to the plant's environmental history. These 15 factors include property categorization factors and disclosure factors, which are explained in the following subsections.

2.2.1 Property Categorization Factors

Property categorization factors determine how a piece of property is categorized according to the Department of Defense (DOD) seven-parcel categorization. There are nine environmental factors that contribute to property categorization. Information within these factors is combined into one attribute data base named "hazwi0", commonly referred to as the hazwaste table. The nine property categorization factors include:

- Hazardous and petroleum material management,
- Hazardous and petroleum waste management,
- Radioactive materials and mixed waste,
- Aboveground/underground storage tanks and pipelines,
- Oil/water separators (considered POL waste storage),
- Wastewater treatment and disposal (exceedances or spills only),
- IRP sites,
- Pesticides, and
- Solid waste (on-site disposal only).

Each building, IRP site, or other location included in hazwi0.tab will be categorized into one or more of the seven categories depending on the specific codes entered in the data base. Each of the categories will be a different color on the Property Categorization Map (Plate 3-1) in the EBS report. The seven categories of property that will be colorized on the map include:

- **Category 1**: Areas where no release or disposal of hazardous substances or petroleum products has occurred (including no migration of these substances from adjacent areas).
- **Category 2**: Areas where petroleum products have been released (or areas that required additional evaluation and are known to contain only petroleum products).
- **Category 3**: Areas where hazardous substances have been released but do not require a removal or remedial action.
- **Category 4**: Areas where hazardous substances have been released and all remedial actions necessary to protect human health and the environment have been taken.
- **Category 5**: Areas where hazardous substances have been released and/or remedial actions are under way, but all required remedial actions have not yet been completed.
- **Category 6**: Areas where hazardous substances have been released but all required remedial actions have not yet been implemented.
- **Category 7**: Areas that are unevaluated or require additional evaluation.

Symbols representing the specific locations of the property categorization factors will appear on the Location of Environmental Factors Map (Plate 3-2) in the EBS report. (Plates 3-1 and 3-2 are usually E-sized drawings.)

2.2.2 Disclosure Factors

1. Instead, they are represented by a sysmbol on the facility on Plate 3-2 where a particular disclosure factor occurs. d in Six of the environmental factors will not affect property categorization if present in a properly managed condition (i.e., no release into the environment). These are the disclosure factors; they will not be represented by a color on Plate 3-1. Instead, a symbol will appear on the facility where a particular disclosure factor occurs. The data for each of these factors are not combined into one data base; they are entered into individual data bases. The six disclosure factors and their corresponding file names:

- Asbestos (asbesti0.tab),
- Lead-based paint (leadi0.tab),
- PCBs (pcbi0.tab),
- Radon (radoni0.tab),
- Medical and biohazardous waste (medi0.tab), and
- Ordnance (ordi0.tab).

Not all plants have all of these disclosure factors associated with them. A data base does not have to be created if a disclosure factor does not apply.

2.2.3 Related Data Bases

In addition to the data bases mentioned above, two other data bases must be created to complete the GIS mapping process. These data bases and their corresponding file names are:

- Building Inventory - A list of all buildings at the plant (bldgi0.tab).
- FACNO List - A list of all unique FACNOs and their names, including buildings, IRP sites, and other locations. In the EBS report, this list is called the index of unique identifiers (facnoi0.tab).

2.3 Getting Started

Prior to data entry, a reference list that includes all of the resources and documentation associated with a plant must be generated (AFPxx-Ref.doc where "xx" is the plant number). It is essential to have a reference number assigned to each document so that the number may be entered into the data base along with its associated data. The documents may have to be referred to several times during the preparation of the EBS report; therefore, it is highly desirable to have an accurate reference list and an organized file drawer to complete the data entry process.

Remember that the EBS is intended to include both historical and current information on each of the 15 environmental factors. Sources for historical information include comprehensive documents such as record searches, preliminary assessments, and management action plans; old lists of tanks, accumulation points, and PCB equipment locations; aerial photographs of the plant; and interviews with long-time plant personnel.

More recent information can be obtained from SPCC plans, annual hazardous waste generation reports, ECAMP reports, asbestos, lead-based paint and radon surveys, IRP documents, and interviews with plant personnel.

Sections 3.0 and 4.0 of this document provide guidelines on entering data associated with property categorization factors and disclosure factors, respectively.

2.4 General Data Entry Guidelines

General data entry guidelines include the following:

- Do not make entries in all capital letters.
- Do not capitalize every word when describing a substance or general location: Domestic hot water piping, not Domestic Hot Water Piping.
- Limit use of abbreviations. Make entries easy to read without having to consult a key.
- Do not enter "Unknown" if the information is not known. Simply leave the field blank.
- All fields should be created as character and not numeric fields.
- Capitalize each word of an IRP site name.
- Be as consistent as possible when naming FACNOs, describing locations, or summarizing activities at a site.

Specific guidelines for each data base are provided in Sections 3.0 and 4.0.

2.5 When Data Entry Is Complete

When the data bases are complete, they can be sent to the TIS. There are several checks that can be done to ensure that the data are accurate and complete before transferring to the TIS:

- Check for duplicate records. Any duplicate should be deleted from the data base.
- Proofread all entries for spelling, punctuation, and capitalization.
- Check that FACNO names make sense and adequately describe the FACNO.
- Run the MapInfo EBS Processing System Version 2.0 to ensure:
  - Those fields required for the property categorization analysis are completed.
  - Each unique FACNO in the hazwi0.tab has a correspondence FACNO in one of the map tables. i.e., the attribute data and the maps reflect the same information.

Review the data base as often as necessary to identify any inconsistencies.

Section 3.0 Property Categorization Factors Data Entry Guidelines

This section provides guidelines for creating the Hazwaste Data Base for the nine property categorization factors.

3.1 Property Categorization Methodology

The seven DOD Environmental Conditions Categories that describe plant property are derived from data entered into specific fields in hazwi0.tab. These fields are HISTORY, STATUS, RASTATUS, and TYPE. During the property recategorization process, the program will assign category numbers to each record in the hazwi0 based on the data contained in these fields. It is imperative that the user know the relationship between the fields used during the recategorization process. Appendix Table 3-1 shows the relationship between the fields and provides a listing of the valid field entries for each of the four fields used to determine property categorization. The user must use Appendix Table 3-1 to ensure the data are entered accurately and completely.

Appendix Table 3-1 provides a summary of the property categorization methodology used to arrive at the correct colors on Condition of Property map (Plate 3-1) in the EBS report. If after the map is generated a particular area on the map appears to be colored incorrectly, review hazwi0 and check that the proper entries have been made as per Appendix Table 3-1.

In many cases, there may be different categories of data associated with the same piece of property. When this occurs, the category with the highest number is assigned to that property (for mapping purposes). For example, if Landfill 1 had soil contamination that has been cleaned up (RASTATUS = C, Category 4) but has other remedial action underway (RASTATUS = U, Category 5), then Landfill 1 would be colored yellow (Category 5). The user can always use the hazwi0.tab file to determine all property categorizations for a specific FACNO.

3.2 Setting Up the Hazwaste Data Base

Use the guidelines in Appendix Table 3-2 when entering data into hazwi0. . The individual data elements in the database design are shown on the table. The "Order" is the sequence of the fields in the data base. The "Field Name" is the field or column heading. The "Field Width" is the number of characters that can be entered into a particular field. **Information entered must not exceed the field width**. Some fields have a limited list of entries that are valid for analysis. Other fields have various entries that are valid and may differ from plant to plant. The "Definition" of the field names or valid entries, and "Comments" on data entry are also provided in Appendix Table 3-2.

Table 3-1
Summary of Property Categorization Methodology

Category	Brief Definition	Color on EBS Plate 3-1	Valid Entries in Hazwaste Data Base			
			STATUS	HISTORY	RASTATUS	TYPE
1	No release; POL, POLW, HAZ, HAZW, PES, RAD, RADW	White	Y	S	No data entered for "clean" properties	POL, POLW, HAZ, HAZW, PES, RAD, RADW
2	Release of POL or POLW	Blue	Y	R or D	N, C, U or F	POL, POLW
2	Unevaluated or further investigation required	Blue	A	S,R or D	No data entered	POL, POLW
3	Release HAZ, HAZW, PES, RAD, RADW; below action level	Light Green	Y	R or D	N	HAZ, HAZW, PES, RAD, RADW
4	Release HAZ, HAZW, PES, RAD, RADW; cleanup complete	Dark Green	Y	R or D	C	HAZ, HAZW, PES, RAD, RADW
5	Release HAZ, HAZW, PES, RAD, RADW; cleanup underway	Yellow	Y	R or D	U	HAZ, HAZW, PES, RAD, RADW
6	Release HAZ, HAZW, PES, RAD, RADW; cleanup not begun	Red	Y	R or D	F	HAZ, HAZW, PES, RAD, RADW
7	Unevaluated or requires further investigation	Gray	A	S, R or D	No Data Entered	HAZ, HAZW, PES, RAD, RADW

Key:

A	=	**Additional investigation required, or unevaluated**
C	=	Remedial action complete and/or no further remedial action required
D	=	Disposal
F	=	Remedial action not yet implemented, will occur in the future
N	=	No remedial action required
R	=	Release
S	=	Storage
U	=	Remedial action underway
Y	=	Entered for all records except those under condition "A"
POL	=	Petroleum product
POLW	=	Petroleum waste product
HAZ	=	Hazardous material
HAZW	=	Hazardous waste material
PES	=	Pesticide
RAD	=	Radioactive material
RADW	=	Radioactive waste material

3.2.1.1.1.1 File Name: HAZWI0

Table 3-2
Hazwaste Data Base Dictionary

Order	Field Name	Field Width	Valid Entry	Definition	Comment
1	FACNO	10	Various	Facility number associated with the nearest building number, IRP site number, or unique ID where substance is stored, release or disposed	• Use numbers and names that facility uses • USTs, ASTs, and OWSs (refer to SOURCE) are assigned unique FACNOs. Combine FACNO and TANKNO with a dash (not to exceed a field width of 10)
2	TANKNO	10	Various (for tanks and OWSs)	Tank ID number assigned to USTs, ASTs, and OWSs	• Use numbers that facility uses • If tank does not have a number, assign it a unique number • Choose a TANKNO that, when combined with the FACNO, will not exceed a field width of 10 • If tank has a replacement, use the same TANKNO followed by an "X" to identify the replacement tank.
3	STATUS	1	Refer to Table 3-1 A = Additional investigation is required Y = All other records	A code indicating whether the property needs further investigation or not.	• An "A" usually refers to an IRP site that is still under investigation (remedial action has not been determined) • If an "A" is entered there is no need to enter anything in RASTATUS • Do not enter an "A" simply because ALL the information could not be obtained (e.g., tank capacity) • Do enter an "A" if vital information is missing or questionable (e.g., the presence of a tank)
4	CON	10	BOEING = Boeing North America; CA = Cabaco, Inc.; FAA = Federal Aviation Administration; Hughes = Hughes Missile Systems Company; LA = Los Angeles County; **LASC = ???????;** LK = Lockheed; LM = Lockheed Martin; MD = McDonnell Douglas; MM = Martin Marietta; NO = Northrop; **PA = ??????**; RI = Rockwell International Corporation; EGG = EG&G	Name of the contractor that occupies the above FACNO	• List at left is not exclusive. • Entering the contractor has the cross-purpose of tracking changes made to the data base.

3.2.1.1.1.2 FILE NAME: HAZWI0

Table 3-2
Hazwaste Data Base Dictionary

Order	Field Name	Field Width	Valid Entry	Definition	Comment
5	LOCATION	40	Various	Description of location of storage, release, or disposal; name of IRP site; name of department; room number; SWMU number is not used for the FACNO	• Do not restate the building number; it is already entered in the FACNO
6	HISTORY	1	Refer to Table 3-1 S = Storage R = Release D = Disposal	A code indicating if the substance was stored, released, or disposed.	• An "S" refers to storage only. If a release has occurred at the same location, create a duplicate record with an "R" in this field. • If an "R" or "D" entered here, enter a code in RASTATUS
7	TYPE	4	Refer to Table 3-1 HAZ = Hazardous material; HAZW = Hazardous waste; POL = Petroleum product; POLW = Petroleum waste; PES = Pesticide; RAD = Radioactive material; RADW = Radioactive waste	A code indicating the type of substance stored, released or disposed.,	• Do not add "W" to HAZ or POL if the substance that was released/disposed was a material and not a waste • Enter disclosure factors as HAZW if of asbestos, lead-based paint, PCB, radon, medical/biohazard, or ordnancee. • Enter POL for diesel and heating oil only. Gasoline is HAZ, not POL. Waste oil is HAZW.
8	SUBSTANCE	40	Various	Hazardous substance as defined by CERCLA and petroleum produces and their derivatives	• Generally enter one substance per record (line of data) unless source is an accumulation point, 90-day storage area, TSDF, or IRP site with many released substances in one area • Refer to 40 CFR 302.4 for list of CERCLA hazardous substances. Use these nomenclature • Generally list all substances being stored regardless of whether it is a CERCLA hazardous substance. Often the hazardous constituent of a product is unknown.

3.2.1.1.1.3 FILE NAME: HAZWI0

Table 3-2
Hazwaste Data Base Dictionary

Order	Field Name	Field Width	Valid Entry	Definition	Comment
9	SOURCE	20	UST = Underground storage tank; AST = Aboveground storage tank; OWS = Oil water separator; Accumulation point = RCRA satellite accumulation area; 90-day storage = Less than 90 day storage area; TSDF = Permitted treatment, storage and disposal facility unit; Drum = 55-,30-,5-gallon drum Process tank = tank used in process and not for waste Landfill = landfill Machine pit = sump Outfall = NPDES discharge point Degreaser = Vapor degreaser	Unit where storage, release or disposal occurred	• List at left includes most common sources; it is not exclusive. Use the "source" field to group together various substances • Do not enter type of activity/process that generated the substance
10	QUANTITY	10	Numbers	Amount in appropriate units of measure of storage, release or disposal	• Units are typically volume (for example capacity of tank), but can be area, distance, velocity, etc. • Do not enter commas
11	UNITS	10	gallons, pounds, kilograms, gal/year, lbs/year, cu. ft.	Units of amount stored, released or disposed	• List at left includes most common units; it is not exclusive • Do not capitalize • Enter quantity in English units if both English and metric are available • Enter consistent units of measurement
12	DATESTART	8	04/15/91, 04/91, 1991	The date when storage, release or disposal began	• Enter 2-digit year for all updated entries • If a release event begins and ends on the same day, enter the same date in DATESTART and DATEEND
13	DATEEND	8	04/15/91, 04/91, 1991, Present	The date when the storage, release or disposal ended	• Enter 2-digit year for all updated entries • If storage, release or disposal is occurring at the time of the EBS, enter "Present"

3.2.1.1.1.4 FILE NAME: HAZWI0

Table 3-2
Hazwaste Data Base Dictionary

Order	Field Name	Field Width	Valid Entry	Definition	Comment
14	REFERENCE	20	Refer to Appendix A of plant-specific EBS for appropriate number reference	Reference number(s) from source(s) of information	• Numbers should correspond to numbers in EBS references listed in Appendix A of the EBS • Enter numbers in numerical order • Separate reference numbers by commas, no spaces
15	TANKSTATUS	8	Active = In place and in use Inactive = In place and not in use Removed = Tank not in place	Identifies the actions taken on the tank or oil/water separator	• Use this field for tanks and oil/water separators only
16	ACTIVITY	40	Various	Description of activity, best management practice, remedial action, stage of IRP process, preventive measure, or whatever was done to clean up or prevent release	None
17	RASTATUS	1	Refer to Table 3-1 N = No remedial action required C = Remedial action complete and/or no <u>further</u> remedial action required U = Remedial action underway F= Remedial action not yet implemented, will occur in the future	A code identifying the stage of remediation	• Enter only if a release or disposal has occurred • Remember that this is the status of remedial action, not the status of investigation • The letter entered here will determine the property categorization • Blank **DOES NOT** = Unknown

3.2.1.1.1.5 FILE NAME: HAZWI0

Table 3-2
Hazwaste Data Base Dictionary

Order	Field Name	Field Width	Valid Entry	Definition	Comment
18	CATEGORY	1	Refer to Table 3-1 1 = No release; POL, POLW, HAZ, HAZW, PES, RAD, RADW 2= Release of POL/POLW or POL/POLW unevaluated or further investigation required 3 = Release HAZ, HAZW, PES, RAD, RADW; below action level 4,= Release HAZ, HAZW, PES, RAD, RADW; cleanup complete 5 = Release HAZ, HAZW, PES, RAD, RADW; cleanup underway 6 = Release HAZ, HAZW, PES, RAD, RADW; cleanup not begun, 7 = Unevaluated or requires further investigation (HAZ, HAZW, PES, RAD, RADW)	A code indicating the property category value for the record	• This field is generated using the algorithm of property categorization established by the Air Force and presented in Table 2-1.

Section 4.0 Disclosure Factors Data Entry Guidelines

This section provides guidelines for creating the six disclosure factor data bases and other related data bases required for the mapping process.

4.1 Setting Up the Disclosure Data Bases

The guidelines in the following tables are to be used when creating the disclosure factor data bases. These tables are set up the same way as Appendix Table 3-2. Many of the fields in the disclosure data bases are similar to those in the Hazwaste Data Base. However, the fields for STATUS and RASTATUS are not included because they are necessary only for property categorization. Information in the disclosure data bases does not determine categories of property, but provides additional information regarding disclosure factors at the plant.

Please note that if a spill or release has occurred that is associated with any of the disclosure factors, it must be entered as HAZW into the data base of property categorization factors (hazwi0). The property where that spill or release occurred will then receive a color on the map. For example, if there were a PCB oil leak or spill at a site, this information would be entered in the Hazwaste Data Base, not the PCB data base.

4.1.1 Medical/Biohazardous Waste

The medical/biohazardous waste data base is to include only storage of medical/biohazardous waste; any release or disposal of this waste on plant property must be entered into the Hazwaste Data Base. Medical/biohazardous waste data base guidelines are provided in Appendix Table 4-1.

4.1.2 Ordnance

The ordnance data base is to include only storage of ordnance; any release or disposal of ordnance (UXO) on the plant property must be entered into the Hazwaste Data Base. Ordnance data base guidelines are provided in Appendix Table 4-2.

4.1.3 Asbestos

The asbestos data base is to include only positive results data from asbestos surveys conducted at the plant; any release or disposal of asbestos on plant property must be entered into the Hazwaste Data Base. Asbestos data base guidelines are provided in Appendix Table 4-3.

4.1.4 PCBs

The PCB data base is to include only storage of PCB equipment; any release or disposal of PCBs on the plant property must be entered into the Hazwaste Data Base. PCB data base guidelines are provided in Appendix Table 4-4.

4.1.4.1.1.1 FILE NAME: MEDI0

Table 4-1
Medical/Biohazardous Waste Data Base Dictionary

Order	Field Name	Width	Valid Entry	Definition	Comment
1	FACNO	10	Various	Building number or unique ID	Use numbers and names that facility uses
2	CON	10	Various	Initials of contractor that occupies the above FACNO	None
3	LOCATION	40	Various	Description of storage location; name of department	None
4	SUBSTANCE	40	Various	Substance stored	None
5	QUANTITY	10	Various	Amount of storage	None
6	UNITS	10	gallons, pounds, kilograms, gal/year, lbs/year	Units of amount stored	List at left includes most common units; it is not exclusive
7	DATESTART	8	04/15/91, 04/91, 1991	Date storage began	None
8	DATEEND	8	04/15/91, 04/91, 1991, Present	Date storage ended	If storage is occurring at the time of the EBS, enter "Present"
9	REFERENCE	20	XX,YY,ZZZ	Reference number(s) from source(s) of information	• Numbers should correspond to numbers in EBS reference list • Enter numbers in numerical order • Separate reference numbers by commas, no spaces
10	ACTIVITY	40	Various	Description of activity, best management practice, preventive measure, or whatever was done to prevent release	None

4.1.4.1.1.2 FILE NAME: ORDI0

Table 4-2
Ordnance Data Base Dictionary

Order	Field Name	Width	Valid Entry	Definition	Comment
1	FACNO	10	Various	Building number or unique ID	Use numbers and names that facility uses
2	CON	10	Various	Initials of contractor that occupies the above FACNO	None
3	LOCATION	40	Various	Description of storage location	None
4	SUBSTANCE	40	Various	Substance stored	None
5	DATESTART	8	04/15/91, 04/91, 1991	Date storage began	None
6	DATEEND	8	04/15/91, 04/91, 1991, Present	Date storage ended	If storage is occurring at the time of the EBS, enter "Present"
7	REFERENCE	20	XX,YY,ZZZ	Reference numbers from source(s) of information	• Numbers should correspond to numbers in EBS reference list • Enter numbers in numerical order • Separate reference numbers by commas, no spaces
8	ACTIVITY	40	Various	Description of activity, best management practice, preventive measure, or whatever was done to prevent release	None

4.1.4.1.1.3 FILE NAME: ASBESTI0

Table 4-3
Asbestos Data Base Dictionary

Order	Field Name	Width	Valid Entry	Definition	Comment
1	FACNO	10	Various	Building number or unique ID	Use numbers and names that facility uses
2	CON	10	Various	Initials of contractor that occupies the above FACNO	None
3	LOCATION	40	Various	Description of ACM location	This may describe the sample location or area of a building where ACM is located
4	QUANTITY	10	Various	Amount of ACM	None
5	UNITS	10	sq. feet, lin. feet	Units of amount of ACM	List at left includes most common units; it is not exclusive
6	REFERENCE	20	XX,YY,ZZZ	Reference number(s) from source(s) of information	• Numbers should correspond to numbers in EBS reference list • Enter numbers in numerical order • Separate reference numbers by commas, no spaces
7	ACTIVITY	40	Various	Description of removal activity, best management practice, preventive measure, or whatever was done to cleanup or prevent release	None

4.1.4.1.1.4 FILE NAME: PCBI0

Table 4-4
PCB Data Base Dictionary

Order	Field Name	Width	Valid Entry	Definition	Comment
1	FACNO	10	Various	Building number or unique ID	Use numbers and names that facility uses
2	CON	10	Various	Initials of contractor that occupies the above FACNO	None
3	SERIALNO	10	Various	Serial number of equipment	None
4	LOCATION	40	Various	Description of storage location; name of department	None
5	EQUIPTYPE	20	Transformer, light ballast, capacitor	Equipment that contains PCBs	List at left includes most common equipment; it is not exclusive
6	SIZE	10	Various	Size of equipment	None
7	SUNITS	20	kVA	Units of size	List at left includes most common units; it is not exclusive
8	QUANTITY	10	Various	Equipment capacity	None
9	QUNITS	10	Gallons, kilograms	Units of equipment capacity	List at left includes most common units; it is not exclusive
10	CONC	10	Various	PCB concentration	Enter equipment in the data base only if the PCB concentration is greater than 50 ppm
11	CUNITS	5	ppm	Units of PCB concentration	None
12	DATESTART	8	04/15/91, 04/91, 1991	Date storage began	None
13	DATEEND	8	04/15/91, 04/91, 1991, Present	Date storage ended	If storage is occurring at the time of the EBS, enter "Present"
14	REFERENCE	20	XX,YY,ZZZ	Reference number(s) from source(s) of information	• Numbers should correspond to numbers in EBS reference list • Enter numbers in numerical order • Separate reference numbers by commas, no spaces
15	ACTIVITY	40	Various	Description of activity, best management practice, preventive measure, or whatever was done to prevent release	None

4.1.5 Radon

The radon data base is to include only positive results data from radon surveys conducted at the plant. Radon data base guidelines are provided in Appendix Table 4-5.

4.1.6 Lead-Based Paint

The lead-based paint data base is to include only positive results data from lead-based paint surveys conducted at the plant. If no surveys have been conducted, the building inventory data base shall be used to list all buildings constructed prior to or during 1978. Lead-based paint data base guidelines are provided in Appendix Table 4-6.

4.2 Setting Up Related Data Bases

The building inventory and FACNO list data bases are additional data bases required for the EBS mapping process.

4.2.1 Building Inventory

The building inventory data base is to include all buildings and structures located on the property. Building inventory data base guidelines are provided in Appendix Table 4-7.

4.2.2 FACNO List

The FACNO list data base is to include a list of all unique FACNOs that are entered into each data base. This data base should be generated after all other data bases are complete. The NAME field must be edited so that each FACNO is adequately described. FACNO list data base guidelines are provided in Appendix Table 4-8.

4.2.2.1.1.1 FILE NAME: RADONI0

Table 4-5
Radon Data Base Dictionary

Order	Field Name	Width	Valid Entry	Definition	Comment
1	FACNO	10	Various	Building number or unique ID	Use numbers and names that facility uses
2	CON	10	Various	Initials of contractor that occupies the above FACNO	None
3	LOCATION	40	Various	Description of testing location	None
4	CONC	10	Various	Concentration of radon	None
5	UNITS	5	pCi/L	Units of concentration	None
6	REFERENCE	20	XX,YY,ZZZ	Reference number(s) from source(s) of information	• Numbers should correspond to numbers in EBS reference list • Enter numbers in numerical order • Separate reference numbers by commas, no spaces
7	ACTIVITY	40	Various	Description of activity, best management practice, or whatever was done to prevent release	None

4.2.2.1.1.2 FILE NAME: LEADI0

Table 4-6
Lead-Based Paint Data Base Dictionary

Order	Field Name	Width	Valid Entry	Definition	Comment
1	FACNO	10	Various	Building number or unique ID	Use numbers and names that facility uses
2	CON	10	Various	Initials of contractor that occupies the above FACNO	None
3	LOCATION	40	Various	Description of lead location	This may describe the sample location or the area of the building where the lead-based paint is located
4	CONC	10	Various	Sampling analysis results	If no survey has been conducted, this data base is not needed; use the building inventory to show all buildings constructed before 1978
5	UNITS	10	Various	Concentration units	None
6	LEADSOURCE	20	Various	Description of sample location	This may describe the unit that contains lead-based paint
7	REFERENCE	20	XX,YY,ZZZ	Reference number(s) from source(s) of information	• Numbers should correspond to numbers in EBS reference list • Enter numbers in numerical order • Separate reference numbers by commas, no spaces
8	ACTIVITY	40	Various	Description of activity, best management practice, preventive measure, or whatever was done to prevent release	None

4.2.2.1.1.3 FILE NAME: BLDGI0

Table 4-7
Building Inventory Data Base Dictionary

Order	Field Name	Width	Valid Entry	Definition	Comment
1	FACNO	10	Various	Building number or unique ID	Use numbers and names that facility uses
2	CON	10	Various	Initials of contractor that occupies the above FACNO	None
3	NAME	40	Various	Name of building	Use names that facility uses
4	AREA	10	Various	Area of building	None
5	UNITS	10	square feet	Units of building area	None
6	YEARBUILT	4	19XX	Year of building construction	This field cannot be blank; estimate if unsure and enter "*" in EST field
7	EST	1	*	Indicates YEARBUILT is estimated	Use when YEARBUILT is estimated
8	REFERENCE	20	XX,YY,ZZZ	Reference number(s) from source(s) of information	• Numbers should correspond to numbers in EBS reference list • Enter numbers in numerical order • Separate reference numbers by commas, no spaces

4.2.2.1.1.4 FILE NAME: FACNOI0

Table 4-8
FACNO List Data Base Dictionary

Order	Field Name	Width	Valid Entry	Definition	Comment
1	FACNO	10	Various	Building number, IRP site number, or unique ID	Use numbers and names that facility uses
2	NAME	40	Various	Name of building, IRP site, or other location	Use names that facility uses

Note: There should be no blanks in this data base.

Appendix B
Queries Most Often Used to Display EBS Data

4.2.2.1.1.4.1 Appendix B

Queries Most Often Used to Display EBS Data

The following queries can be used by the MapInfo user to view the attribute data in the EBS processing system.

Building Descriptions
Applicable File: bldgi0.tab
Query: SELECT ALL

Hazardous Waste Storage Areas
Applicable File: hazwi0.tab
Query: SOURCE = “Accumulation Point” or
SOURCE = “90-day storage” or
SOURCE = “TSDF”, and
HISTORY = “S”

Aboveground Storage Tanks
Applicable File: hazwi0.tab
Query: SOURCE = “AST” and
HISTORY = “S”

Underground Storage Tanks
Applicable File: hazwi0.tab
Query: SOURCE = “UST” and
HISTORY = “S”

Buildings Containing Asbestos
Applicable File: asbesti0.tab
Query: SELECT ALL

Buildings Assumed to Contain Lead-Based Paint
Applicable File: bldgi0.tab
Query: YEARBUILT = < 1978

Location of PCB-Containing Equipment
Applicable File: pcbi0.tab
Query: SELECT ALL

Facilities Surveyed for Lead-Based Paint
Applicable File: leadi0.tab

Query: SELECT ALL

Hazardous Materials and Petroleum Products Stored by Facility

Applicable File: hazwi0.tab

Query: TYPE = “POL” or
TYPE = “HAZ” or
TYPE = “RAD” or
TYPE = “PES” and
TYPE = “S”

Hazardous and Petroleum Wastes Stored by Facility

Applicable File: hazwi0.tab

Query: TYPE = “POLW” or
TYPE = “HAZW” or
TYPE = “RADW” and
TYPE = “S”

Hazardous Substances and Petroleum Products/Wastes Releases

Applicable File: hazwi0.tab

Query: TYPE = “POLW” or
TYPE = “HAZ” or
TYPE = “POL” or
TYPE = “RADW” or
TYPE = “RAD” or
TYPE = “PES” or
TYPE = “HAZW” and
HISTORY = “R” or
HISTORY = “D”

Appendix C
Geographic Projections for EBS Plants
Wright-Patterson AFB Technical Information System

Plant	Plant Geographic Projection	Datum	Zone	Units	Spheroid
AFP3	Stateplane	NAD27	5026 Oklahoma North	Feet	Clarke 1866
AFP4	Stateplane	NAD27	5351 Texas North Central	Feet	Clarke 1866
AFP6	Stateplane	NAD27	3651 Georgia East	Feet	Clarke 1866
AFP42	Stateplane	NAD27	3426 California Zone VII	Feet	Clarke 1866
AFP44	Stateplane	NAD27	3176 Arizona Central	Feet	Clarke 1866
AFP85	Stateplane	NAD27	5001 Ohio South	Feet	Clarke 1866
AFP PJKS	Stateplane	NAD27	3476 Colorado Central	Feet	Clarke 1866

Appendix D
Glossary

Glossary

Archive: The process of compressing and storing the current EBS data for a given plant.

ARC/INFO: A geographic information system (GIS) software product designed to capture, store, update, manipulate, analyze, and display geographically referenced information.

ARC/INFO Macro Language (AML): A high-level, algorithmic language that provides full programming capabilities and toolset for building menus to tailor user interfaces for specific applications within ARC/INFO.

ArcLink: A MapBasic program that allows the MapInfo user to import and use ARC/INFO export file formats (.e00 files).

Attribute data: A tabular file containing characteristics of a geographic feature linked to the feature by a unique identifier (FACNO). Attribute data are synonymous with an INFO file (.dat extension) in ARC/INFO or a table (.tab extension) in MapInfo.

Baseline data: Coverage and .dat files for an EBS plant currently residing in the ARC/INFO component of the EBS system.

Coverage: A file containing a set of thematically associated data considered as a unit in ARC/INFO and MapInfo. A coverage represents a single theme, or layer in a map, such as structures, soils, roads or land use.

.dat files: Files containing attribute data stored in ARC/INFO format.

.dbf files: Files containing attribute data stored in dBASE format.

.e00 files: Files containing any coverage that has been converted to ARC/INFO export format.

Export: The process within the Technical Information System (TIS) that selects all the coverage and .dat files for one EBS plant, creates ARC/INFO export format (.e00) files, and inserts the files into a directory ($TlSROOT/afpxx/exports) for access by MapInfo.

FACNO (Unique Identifier): An identifier, up to 10 characters in length, assigned for each plant building, site, or other area not associated with a building or site (e.g., outdoor spill, PCB storage area, UST).

gzip: A command unique to UNIX that allows for further compression beyond the export process. The resulting file will be a gzipped exported file specified as <filename>.e00.gz.

Hazwaste Table: The name commonly used to refer to either the hazwi0.tab in MapInfo or hazw.dat in ARC/INFO.

Import: The process within the TIS that selects .e00 and .dbf files from the import directory ($TlSROOT/afpxx/import_e00) and converts them into coverage and .dat files for the ARC/INFO administrator.

MapBasic: A software language that lets the user customize and automate the MapInfo desktop mapping software.

MapInfo: A software product designed to view, modify, and import/export spatial and tabular data such as EBS plant data.

New data: Coverage and .dat files for an EBS plant that have been altered in MapInfo and imported to the TIS ARC/INFO platform. The data are considered new because they will replace the current EBS plant data set.

Spatial data: Graphic files containing the location, shape of, and relationships among geographic features, usually stored as coordinates and topology. Spatial data are synonymous with a coverage in ARC/INFO and tables in MapInfo.

.tab files: Files containing attribute or spatial data stored in MapInfo.

Technical Information System (TIS): The TIS is the computer system used by the Aeronautical Systems Center (ASC) that serves as the central collection point for environmental sampling and analysis data, and maps relating to the environmental cleanup of Air Force missile and aircraft plants.

wilbur: The Sun SPARCstation on the TIS housed at ASC (Building 8) used to store the EBS plant map and attribute data. The file names and directories used on **wilbur** include:

$TISROOT - parent directory of the TIS system

afpxx - Air Force plant identifier, e.g., afp44

$TlSROOT/afpxx - working directory for the TIS EBS component

$TlSROOT/afpxx/import_e00 - the latest edits to the EBS data prior to insertion into baseline directory

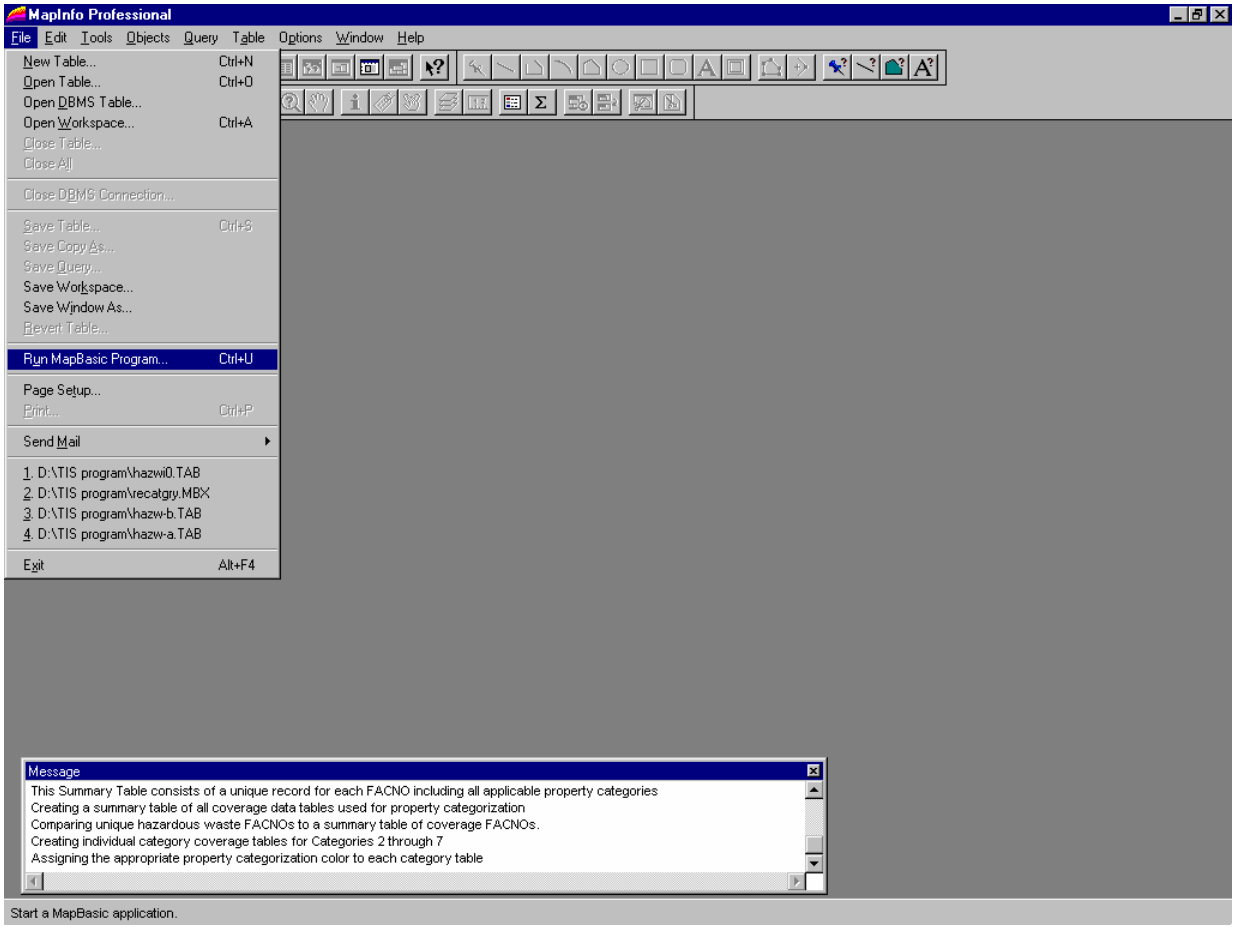
$TlSROOT/afpxx/parent - directory containing EBS files one version old

$TlSROOT/afpxx/exports - baseline version of the EBS ready to import into MapInfo

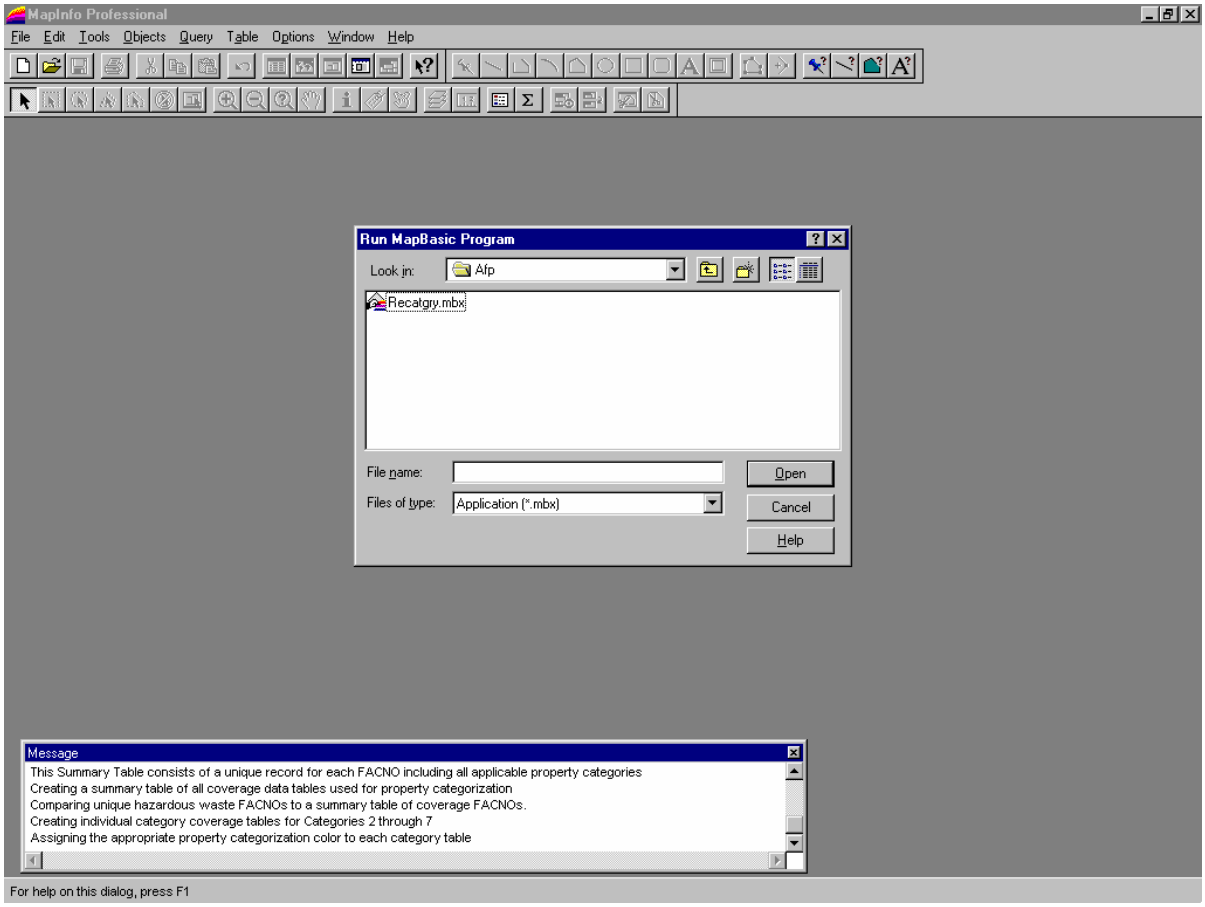
$TlSROOT/afpxx/archive - directory containing gzipped EBS data two versions old.

Appendix E
Project Manager's Guide
MapInfo EBS Process Screen Captures

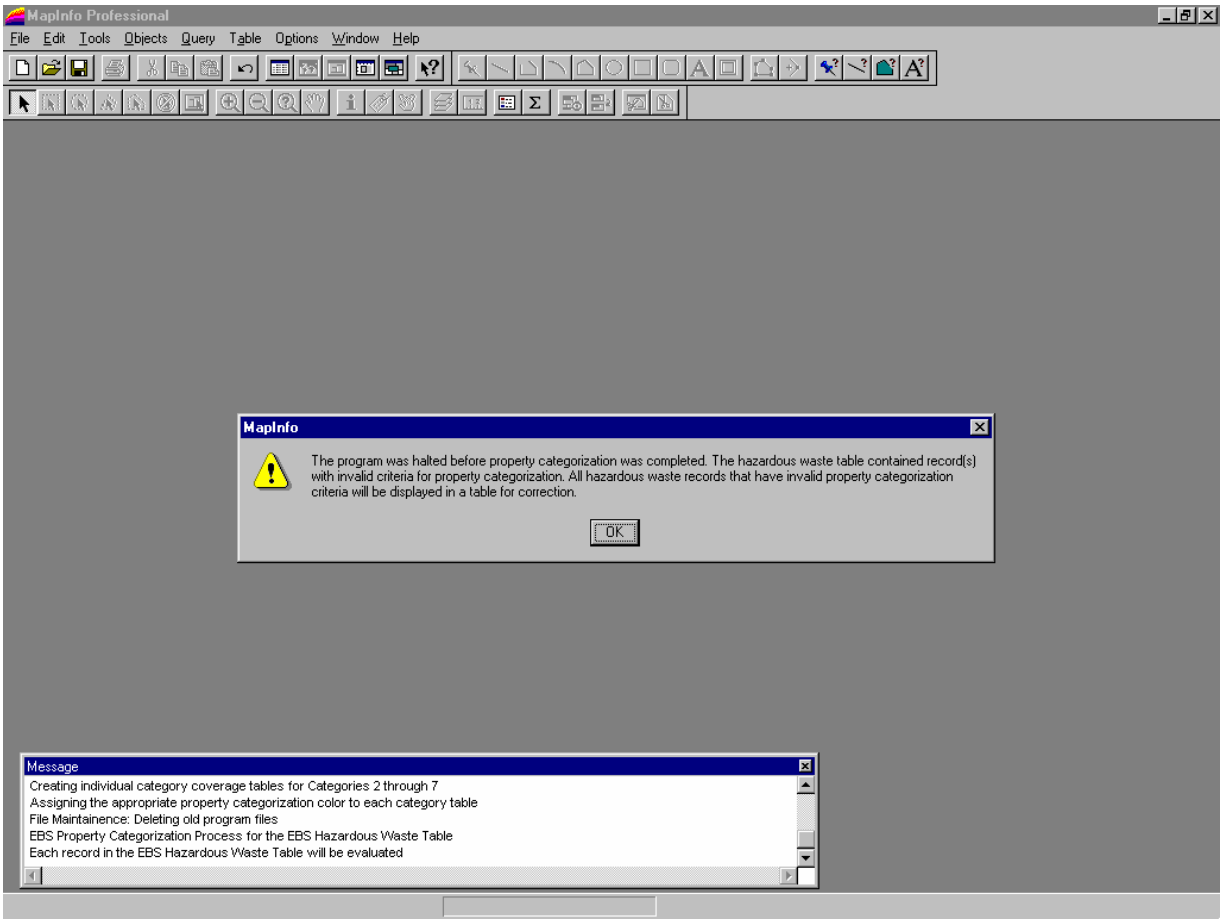
Screen 1 of 2 for starting the recatgry.mbx program



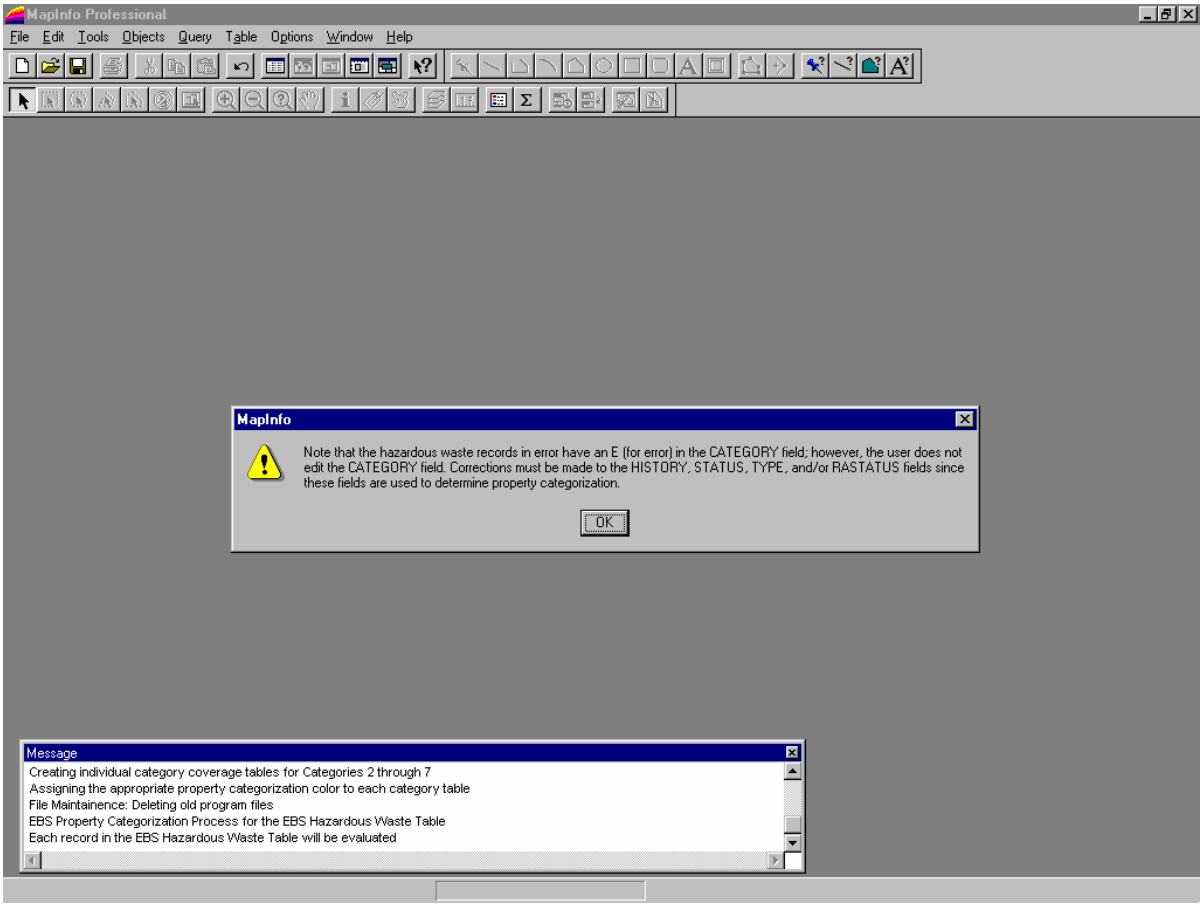
Screen 2 of 2 for starting the recatgry.mbx program



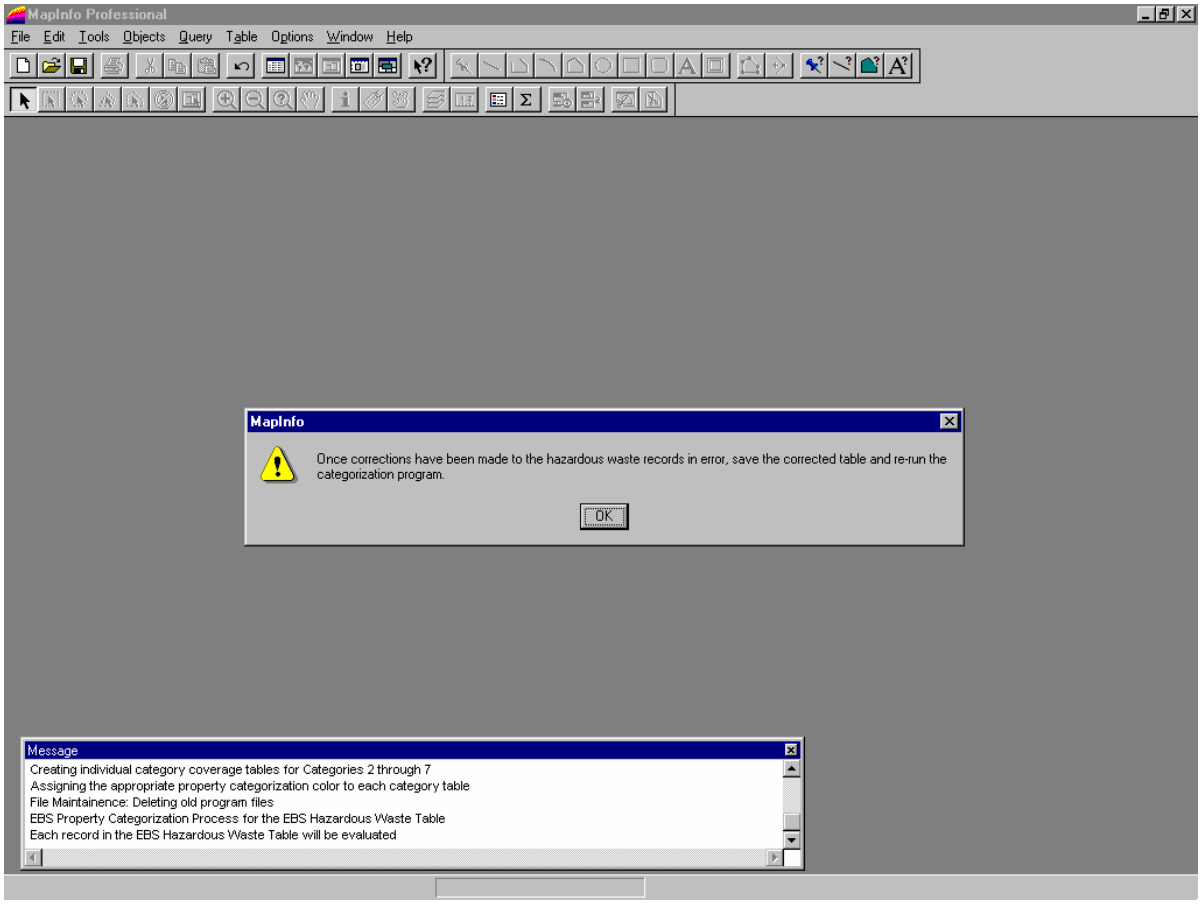
Screen 1 of 7 when an invalid entry in the HISTORY, STATUS, RASTATUS, or TYPE field is encountered. (For valid field entries see Table 1)



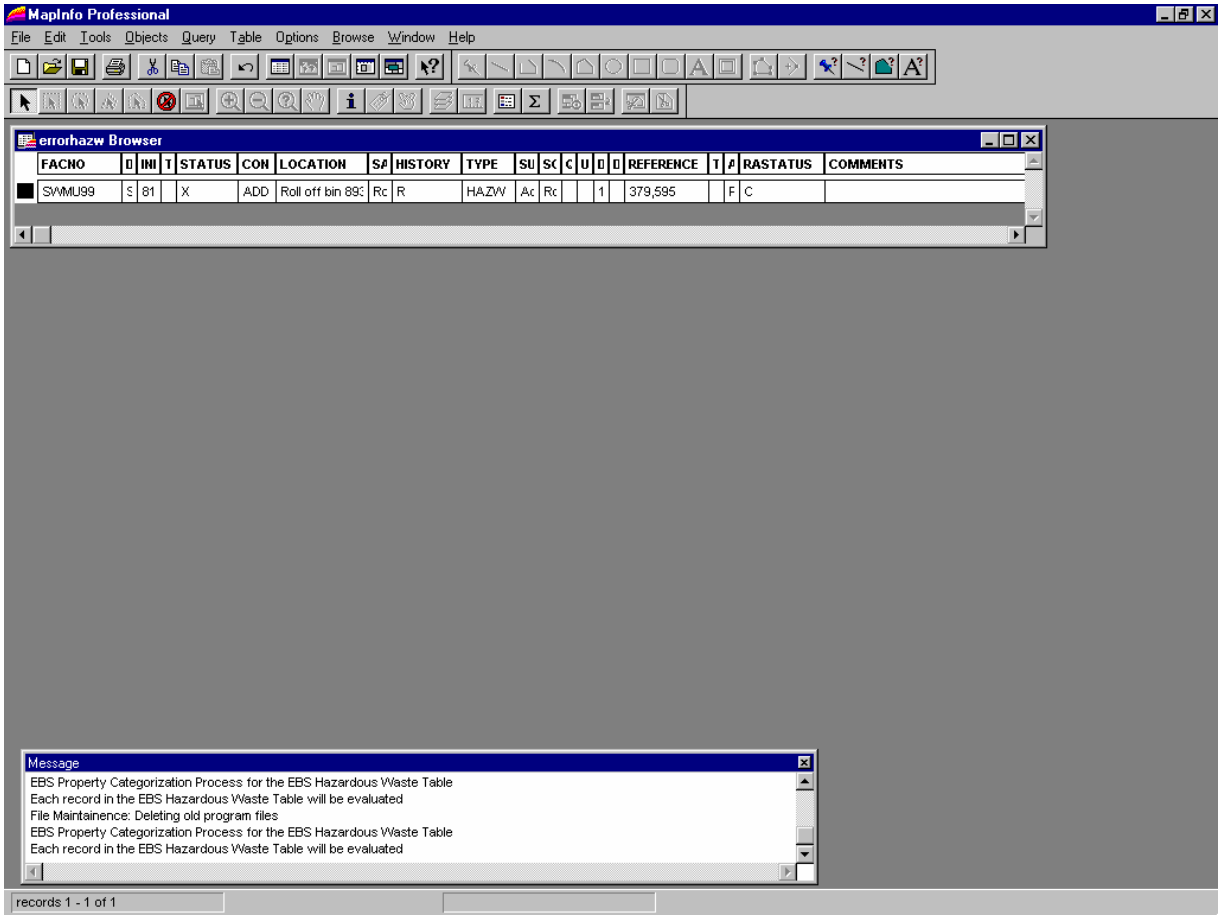
Screen 2 of 7 when an invalid entry in the HISTORY, STATUS, RASTATUS, or TYPE field is encountered.



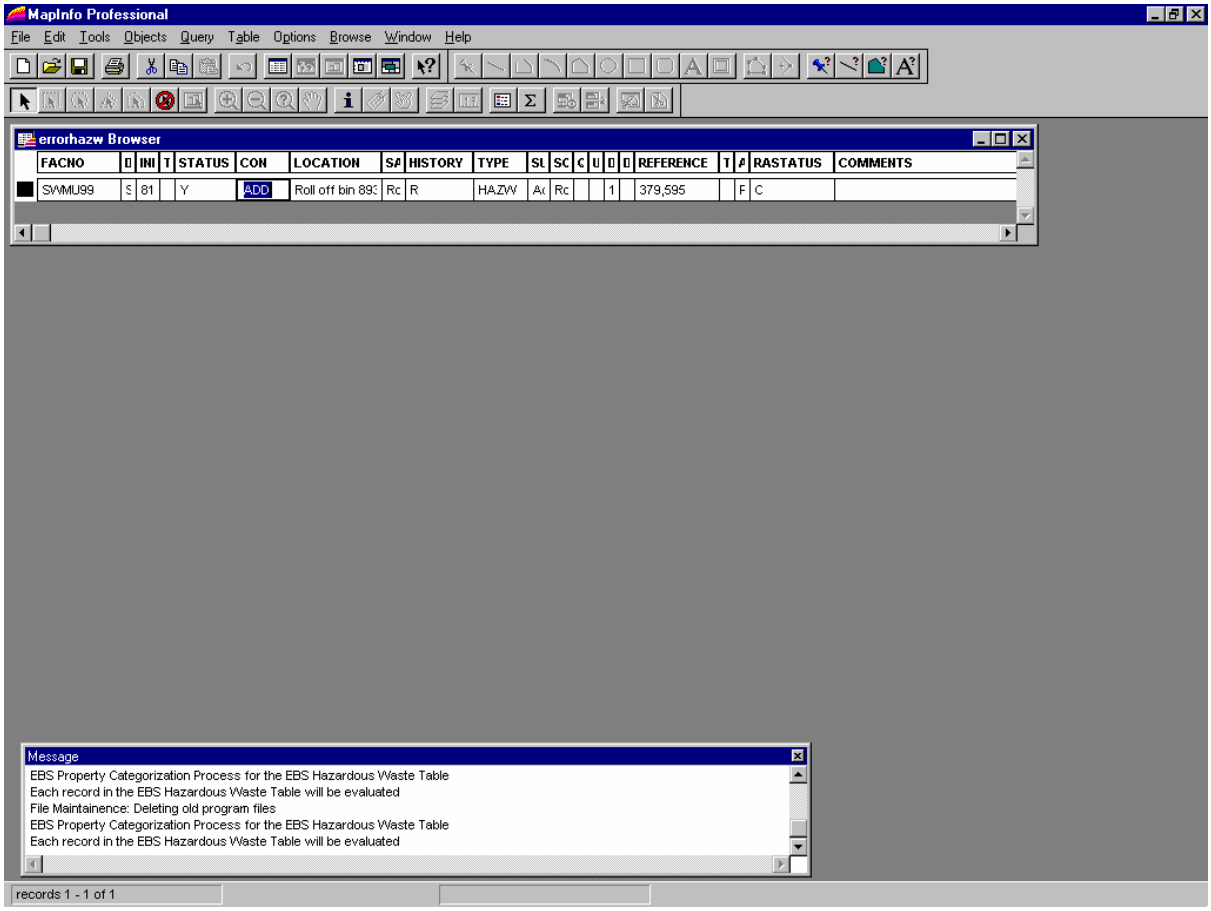
Screen 3 of 7 when an invalid entry in the HISTORY, STATUS, RASTATUS, or TYPE field is encountered.



Screen 4 of 7 when an invalid entry in the HISTORY, STATUS, RASTATUS, or TYPE field is encountered.

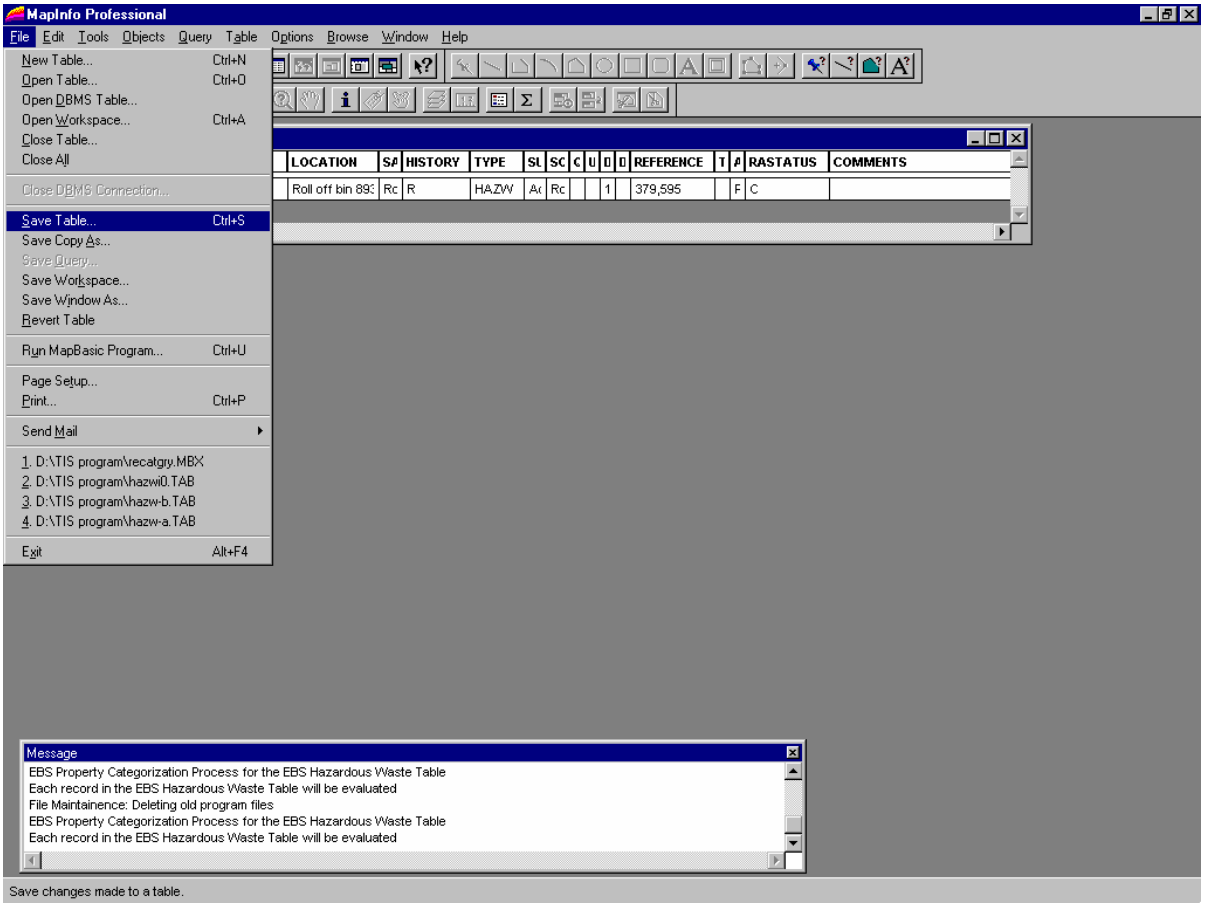


Screen 5 of 7 when an invalid entry in the HISTORY, STATUS, RASTATUS, or TYPE field is encountered.

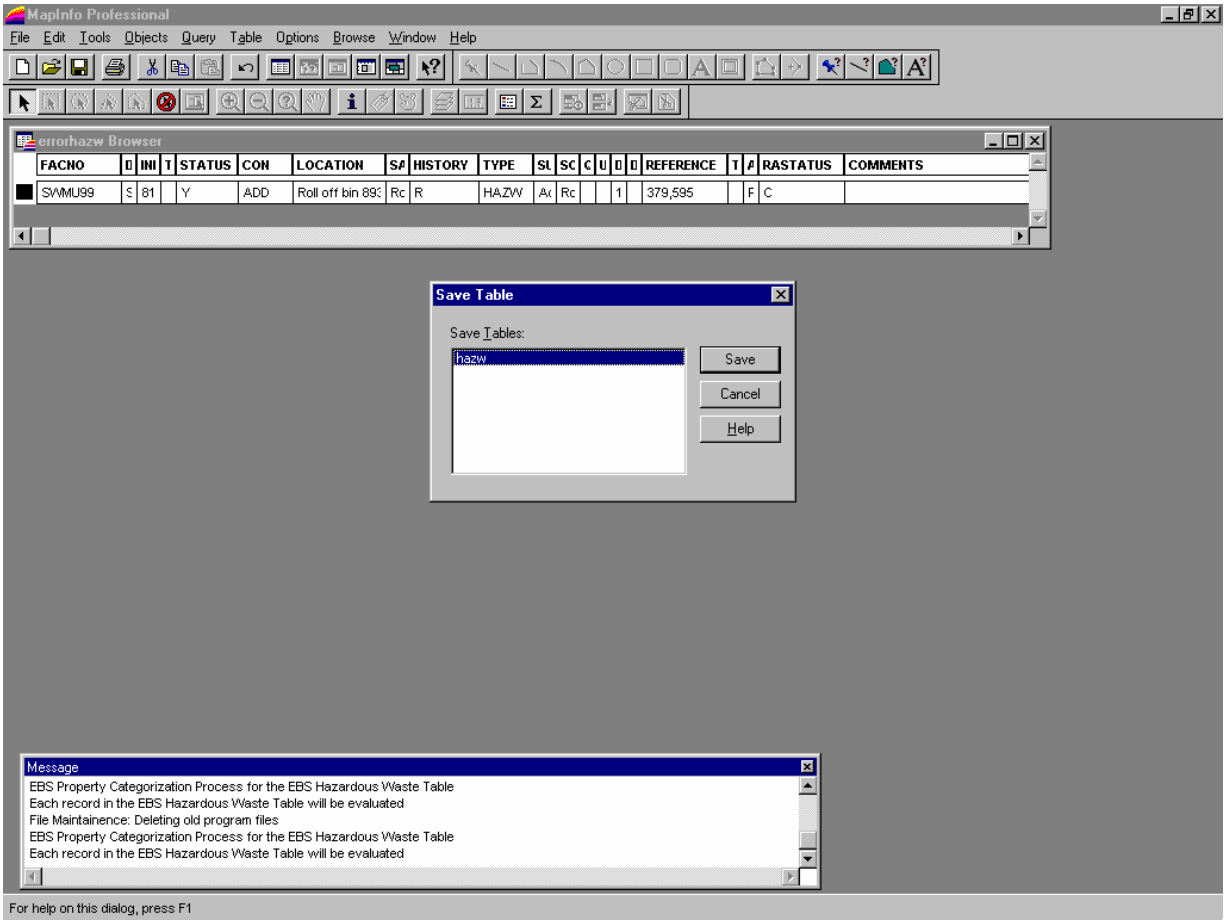


After making changes/fixes in the errorhazw Browser window, save the table. In order to do this 'TAB' out of the field being edited. After hitting the "TAB" key, the "save copy" option under the file pull-down menu becomes active.

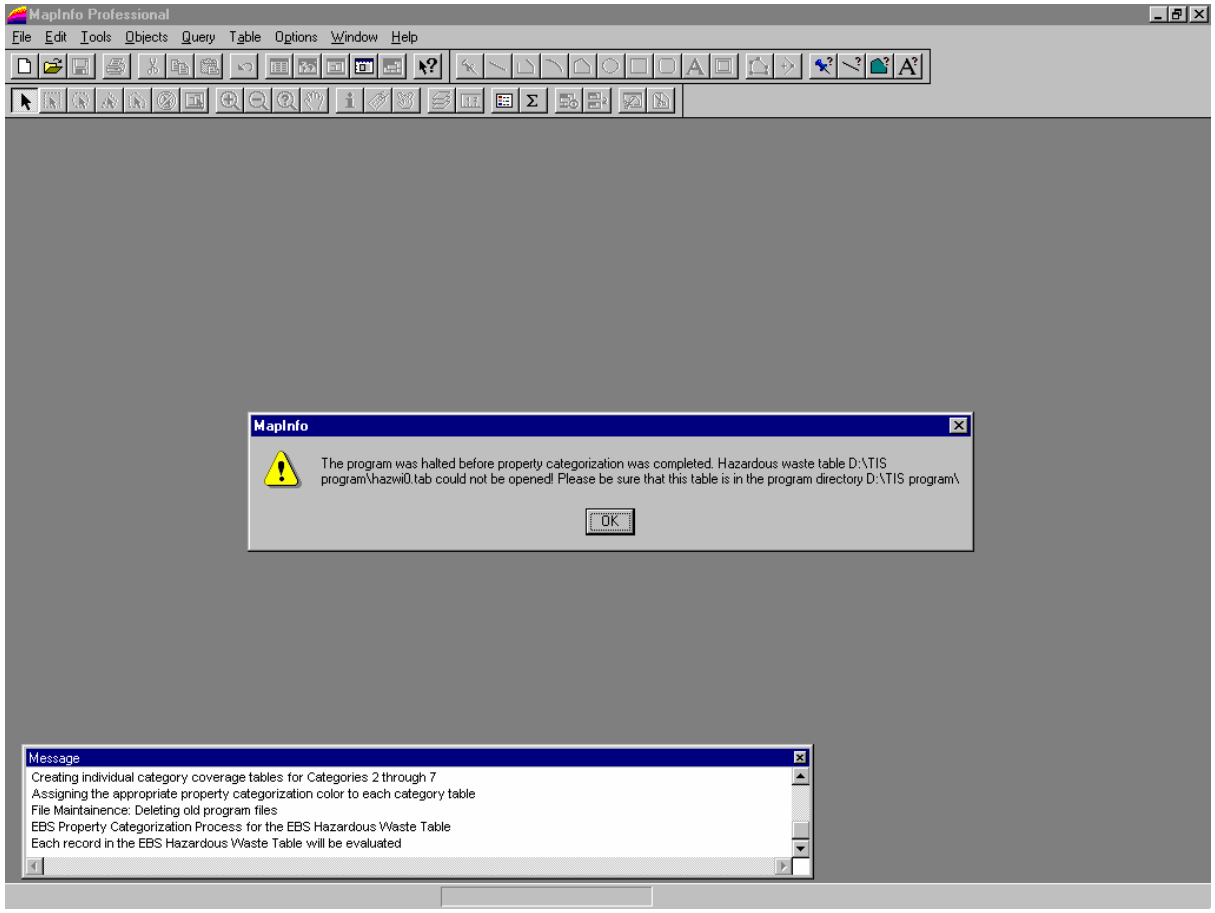
Screen 6 of 7 when an invalid entry in the HISTORY, STATUS, RASTATUS, or TYPE field is encountered.



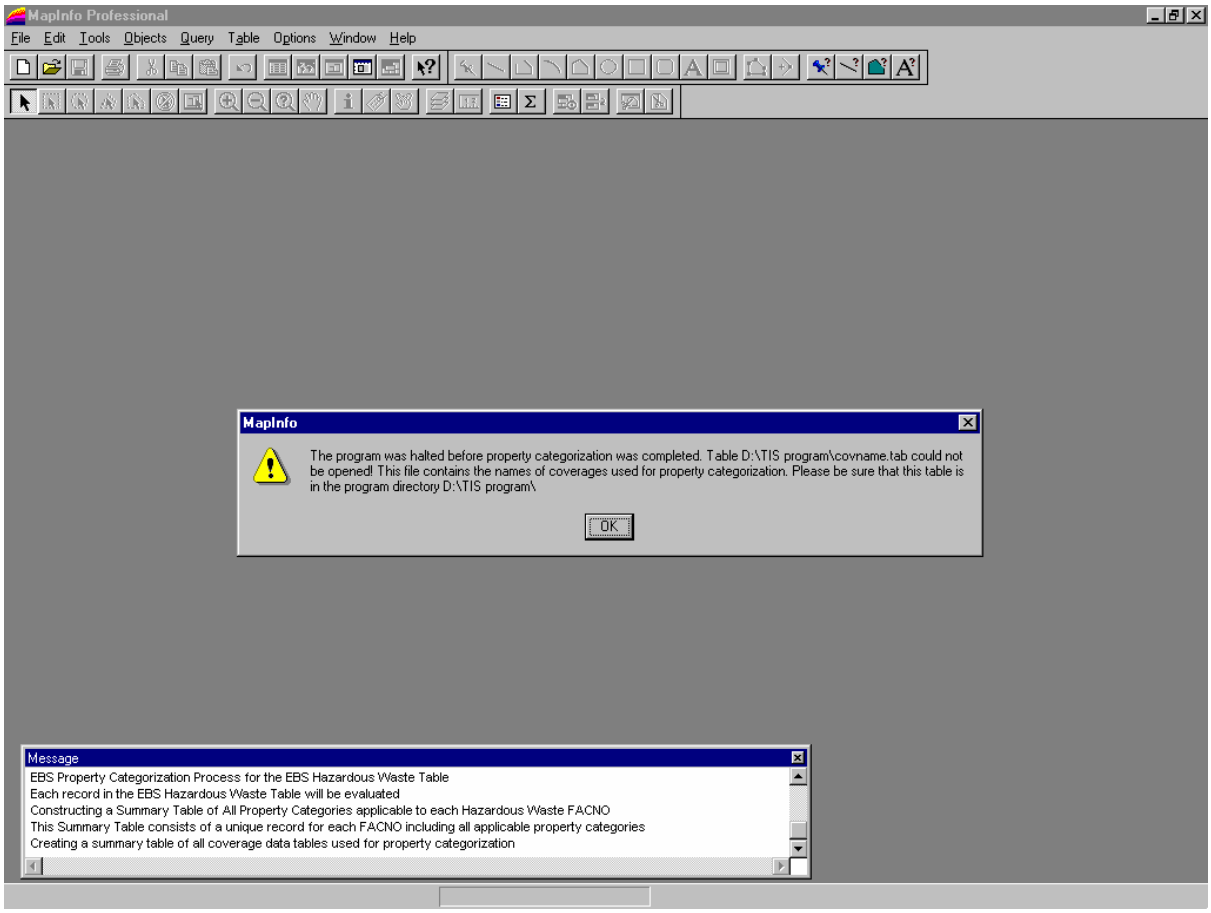
Screen 7 of 7 when an invalid entry in the HISTORY, STATUS, RASTATUS, or TYPE field is encountered.



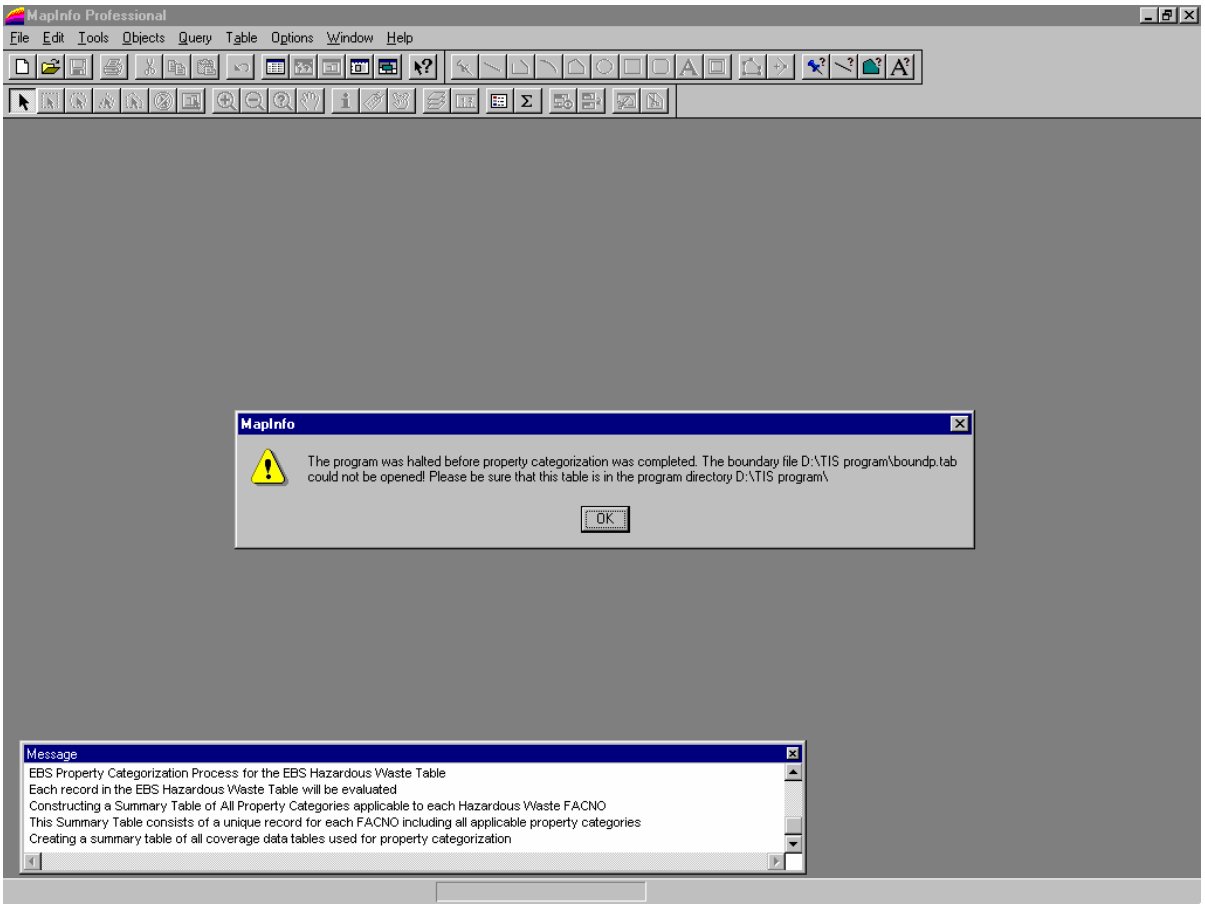
Screen 1 of 1 telling the user that the Hazwi0.tab could not be found.



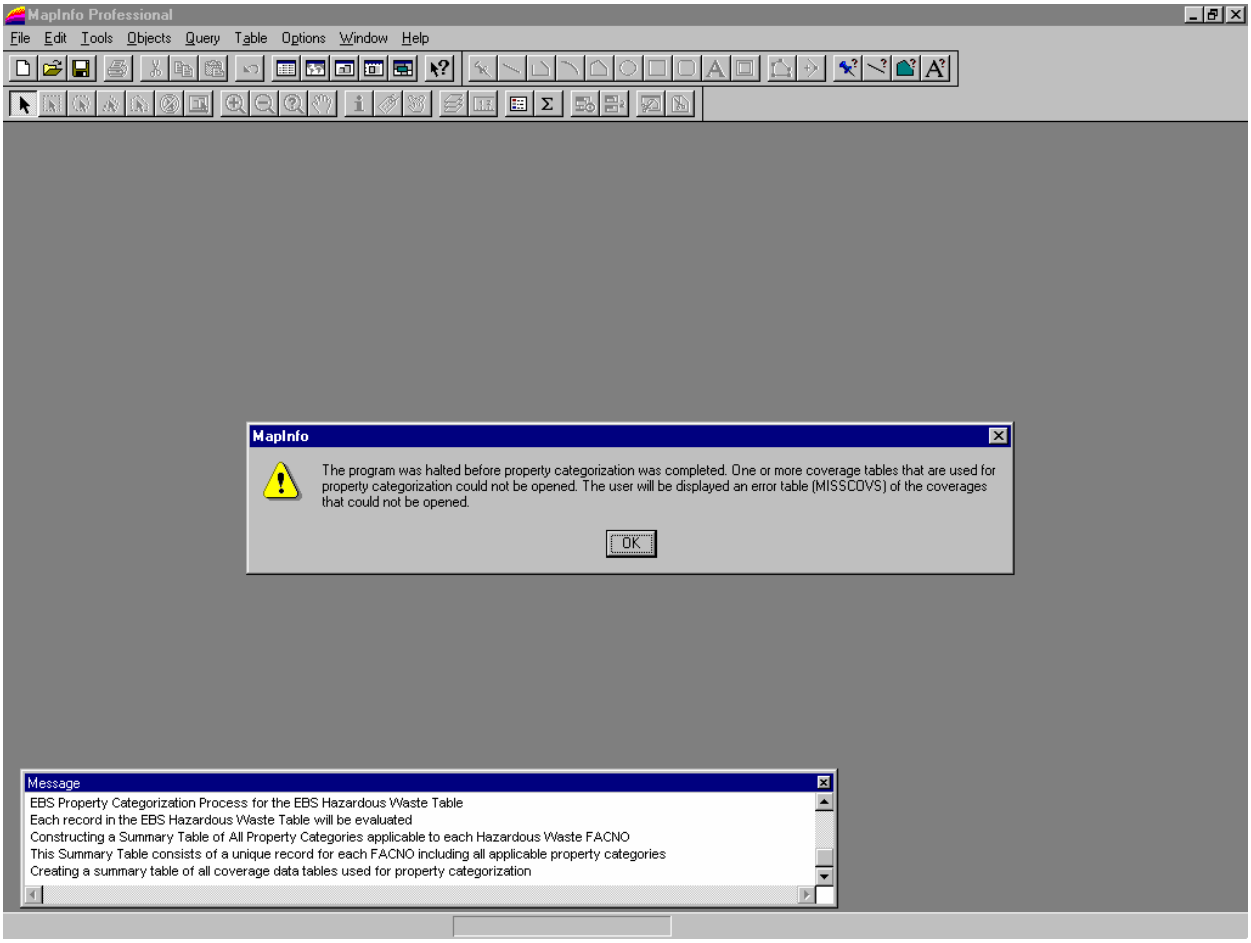
Screen 1 of 1 telling the user that the covname.tab file could not be found.



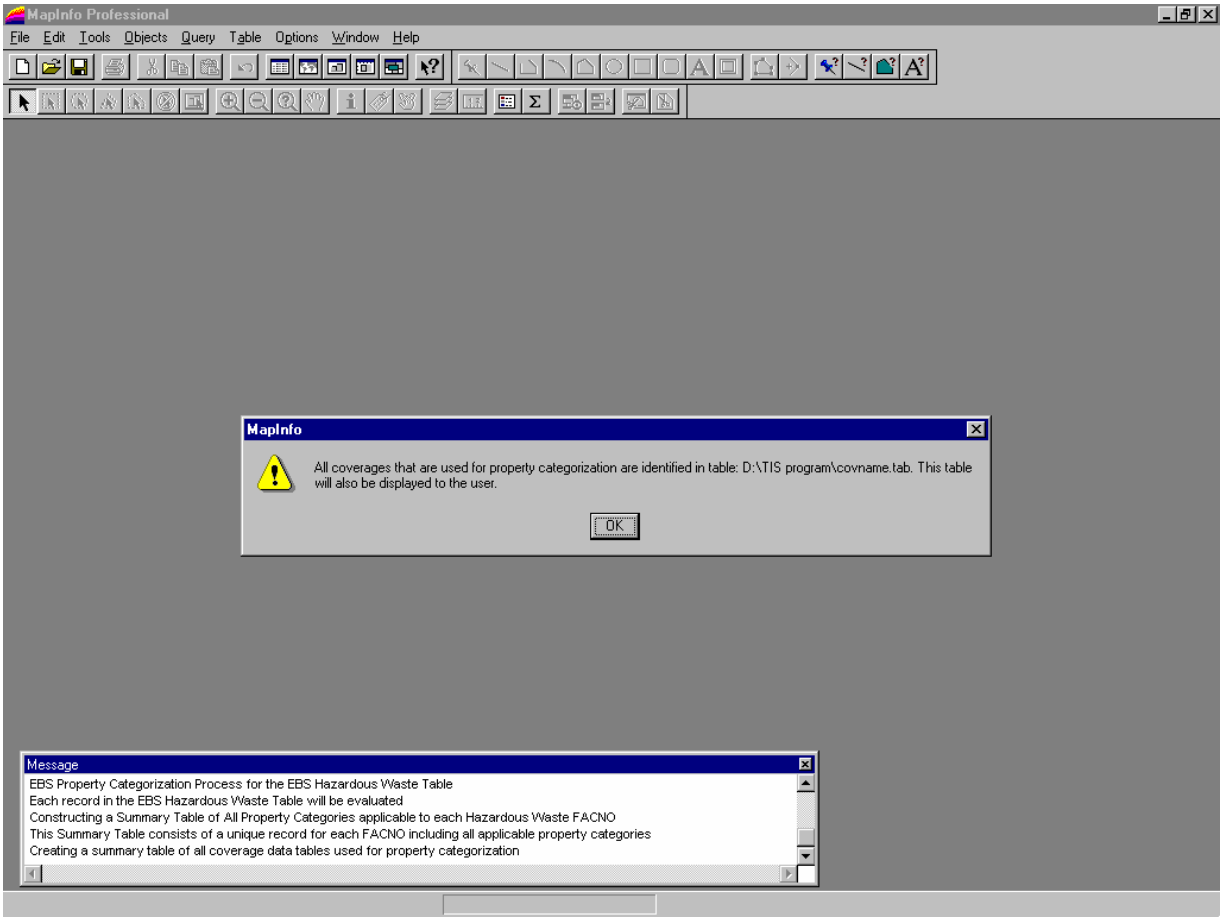
Screen 1 of 1 telling the user that the boundp.tab file could not be found.



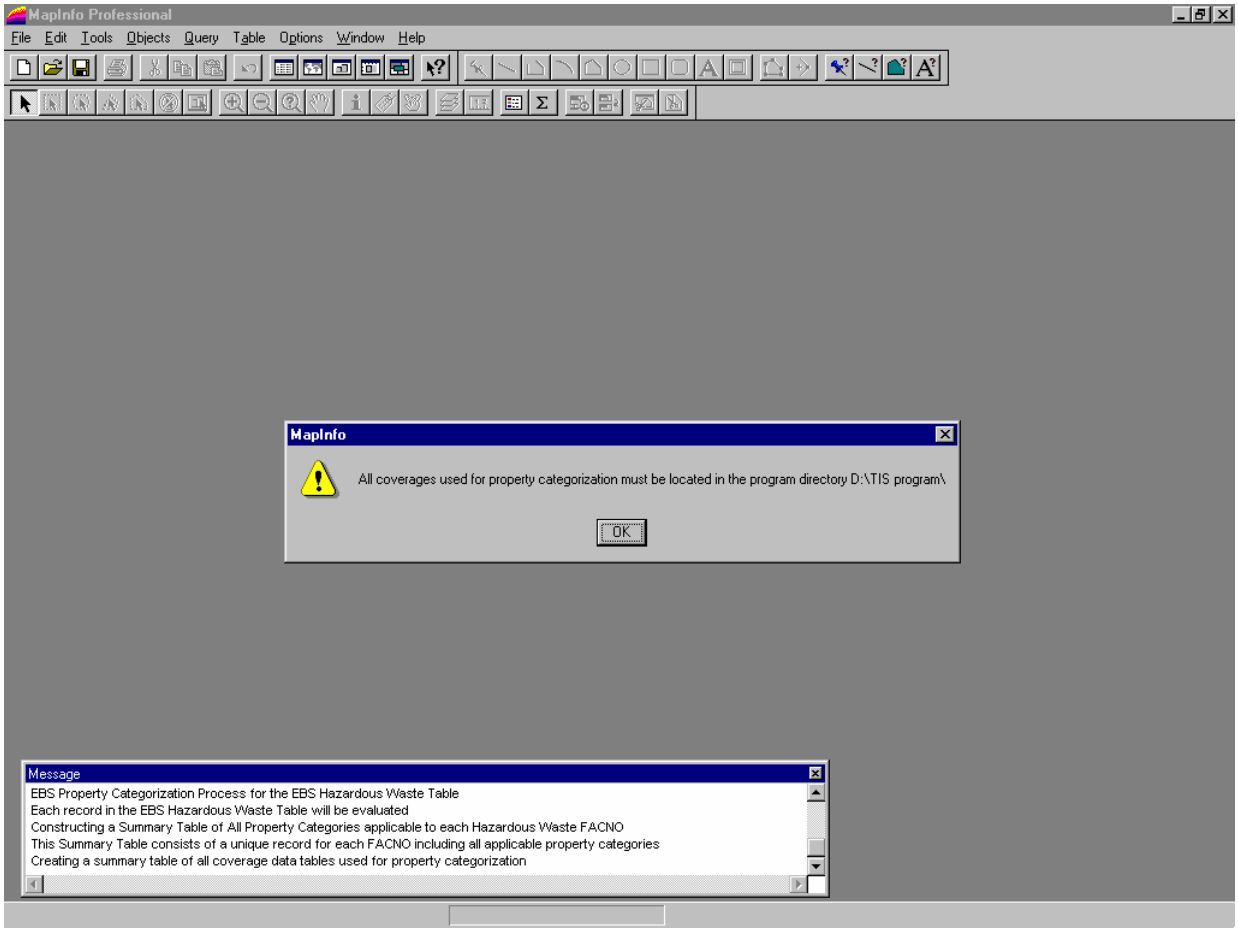
Screen 1 of 4 telling the user that one or more of the coverages needed
for categorization could not be found.



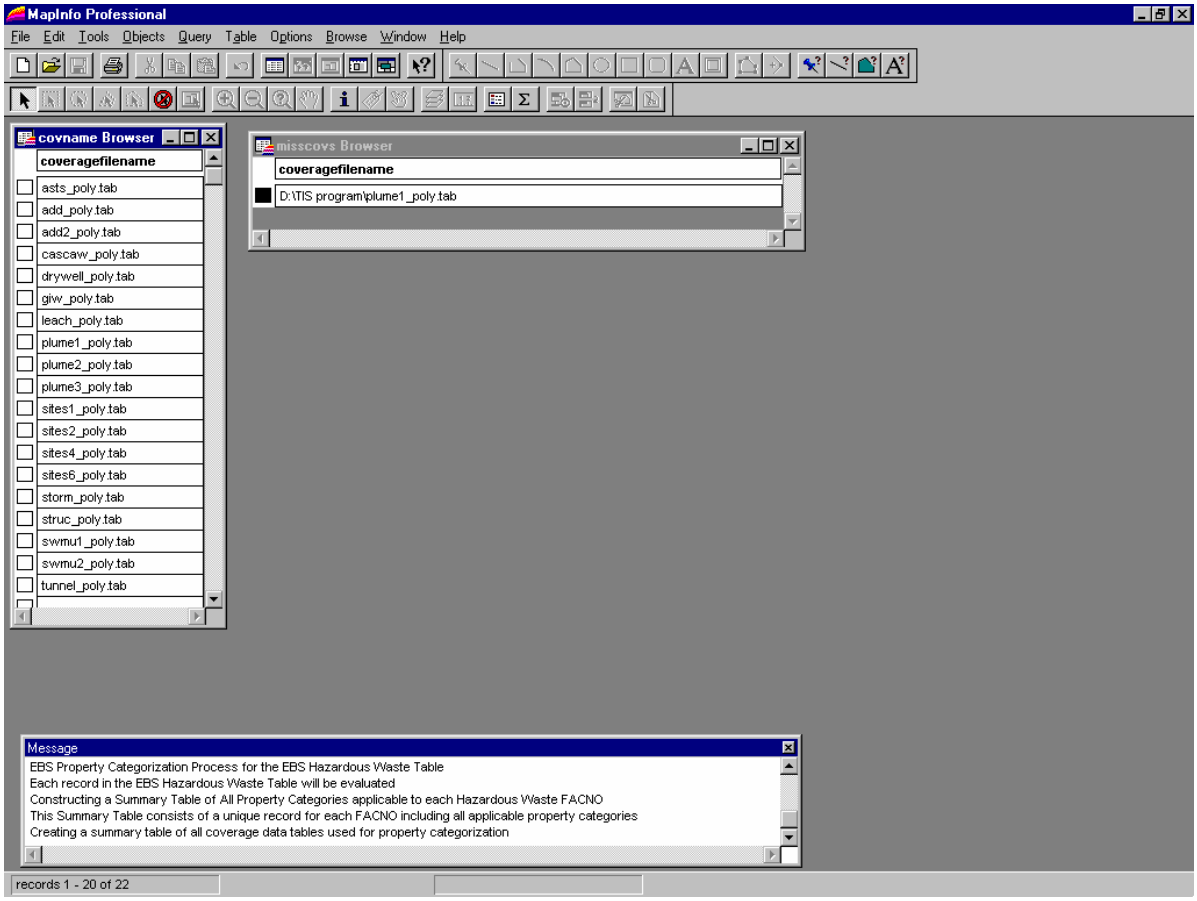
Screen 2 of 4 telling the user that one or more of the coverages needed
for categorization could not be found.



Screen 3 of 4 telling the user that one or more of the coverages needed
for categorization could not be found.

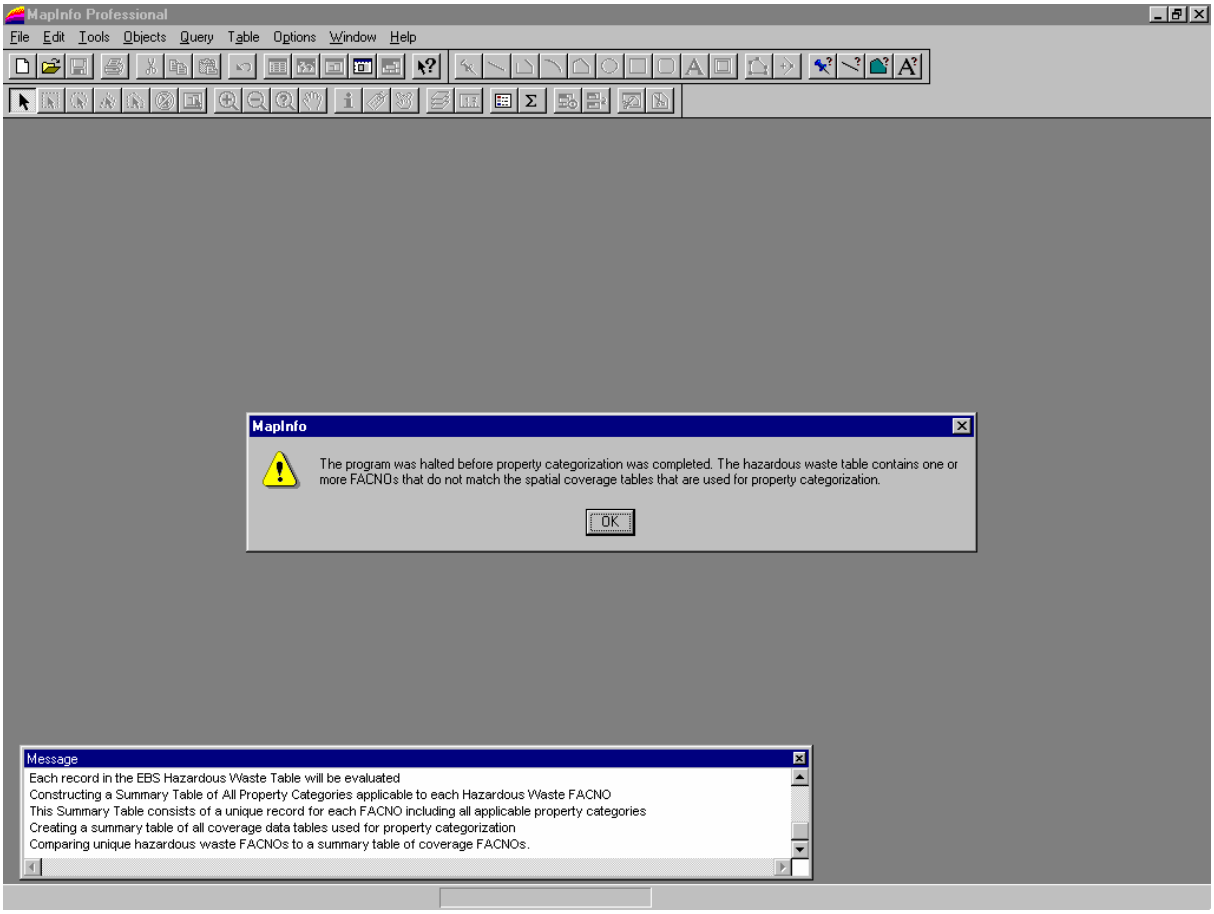


Screen 4 of 4 telling the user that one or more of the coverages needed for categorization could not be found.

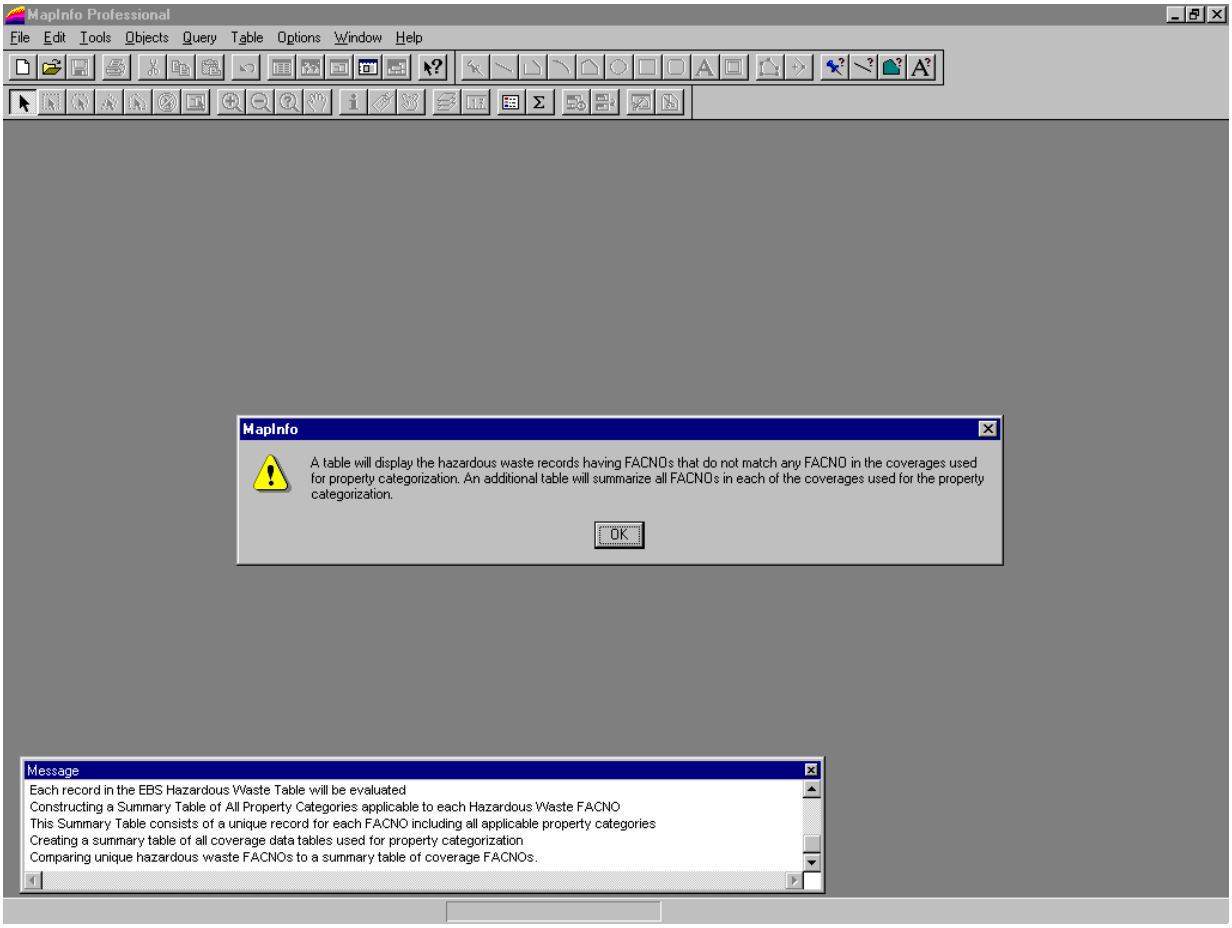


This screen contains two different browser windows. The browser labeled “covname Browser” contains a list of the valid table names for the particular plant. The second window labeled “misscovs Browser” shows the tables that are located in the plants working directory but that do not match any table names in the covname.tab file.

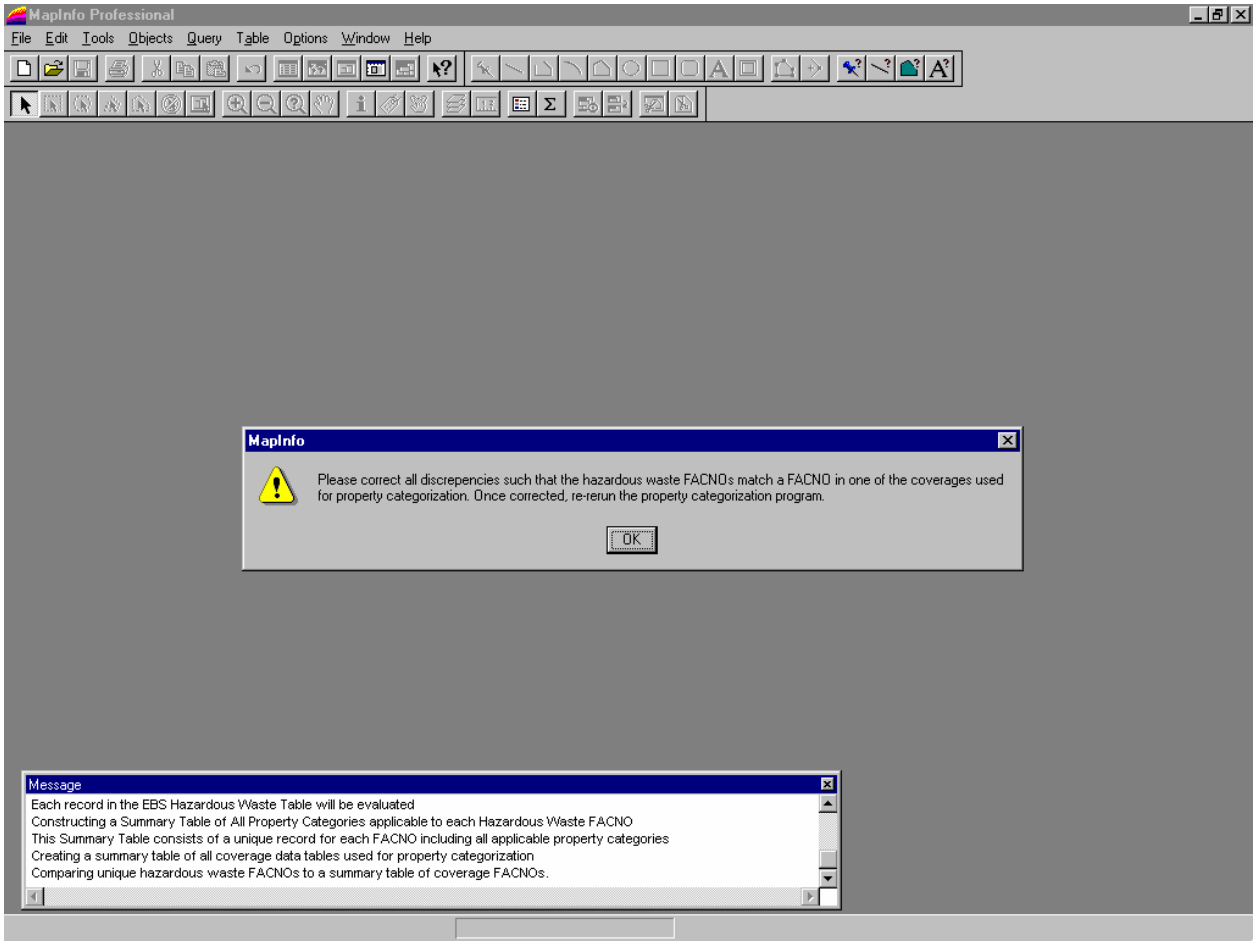
Screen 1 of 4 telling the user that one or more of the facnos contained in the hazwi0.tab file do not have corresponding facnos in any of the spatial coverages.



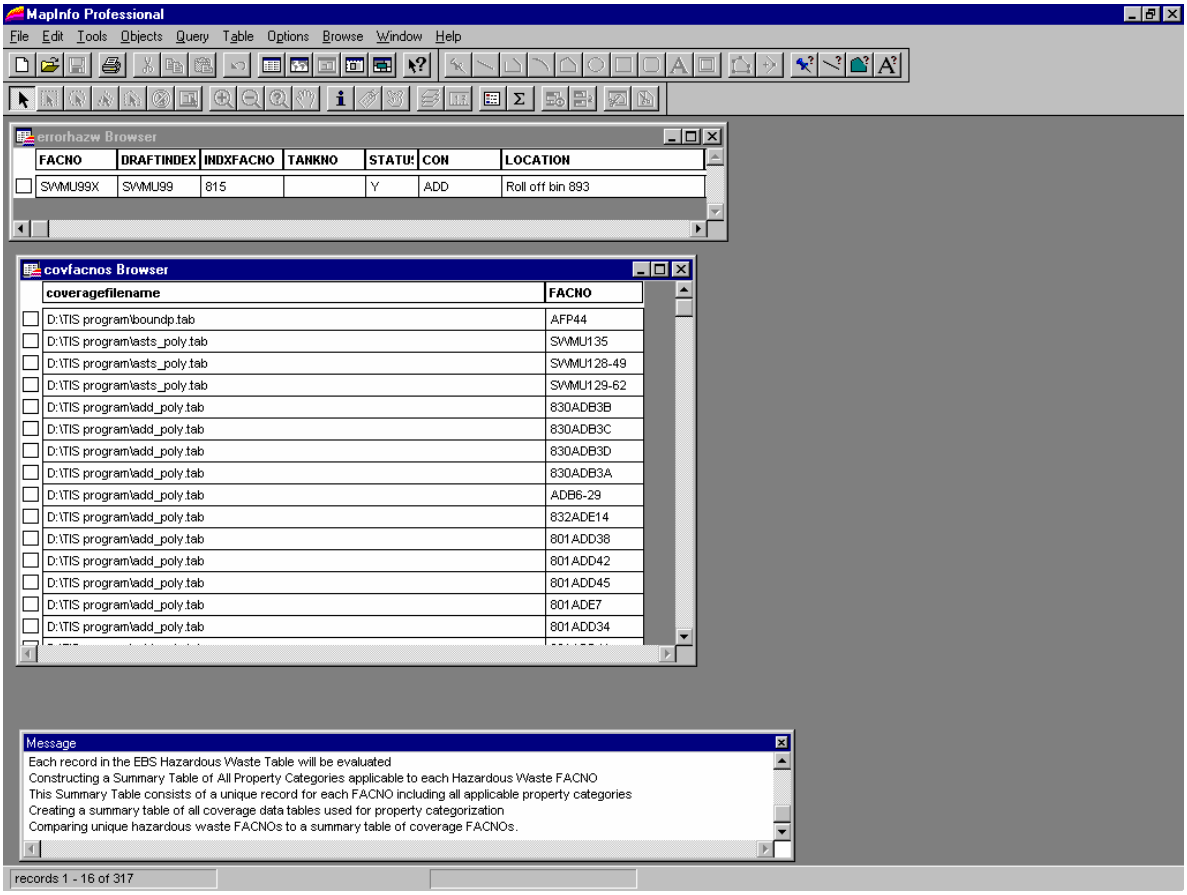
Screen 2 of 4 telling the user that one or more of the facnos contained in the hazwi0.tab file do not have corresponding facnos in any of the spatial coverages.



Screen 3 of 4 telling the user that one or more of the facnos contained in the hazwi0.tab file do not have corresponding facnos in any of the spatial coverages.

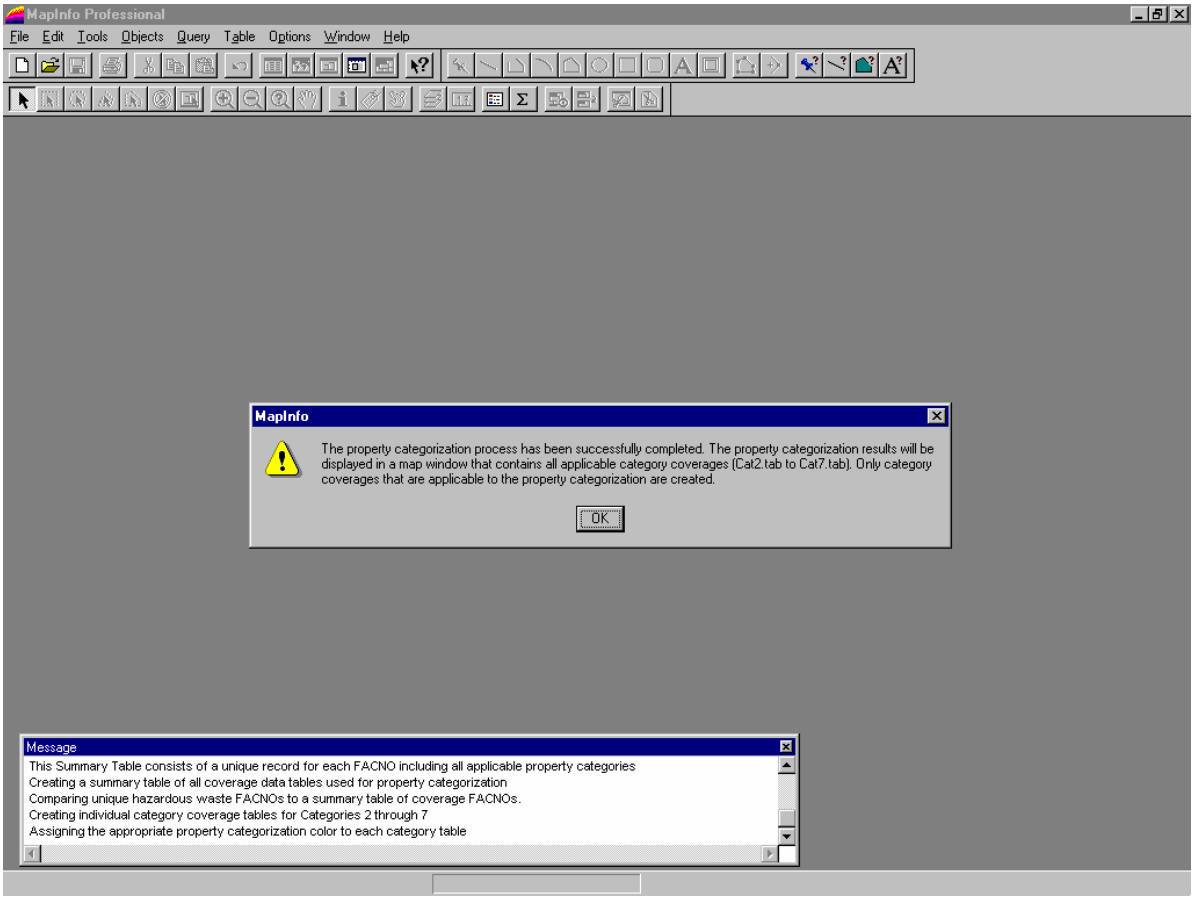


Screen 4 of 4 telling the user that one or more of the facnos contained in the hazwi0.tab file do not have corresponding facnos in any of the spatial coverages.

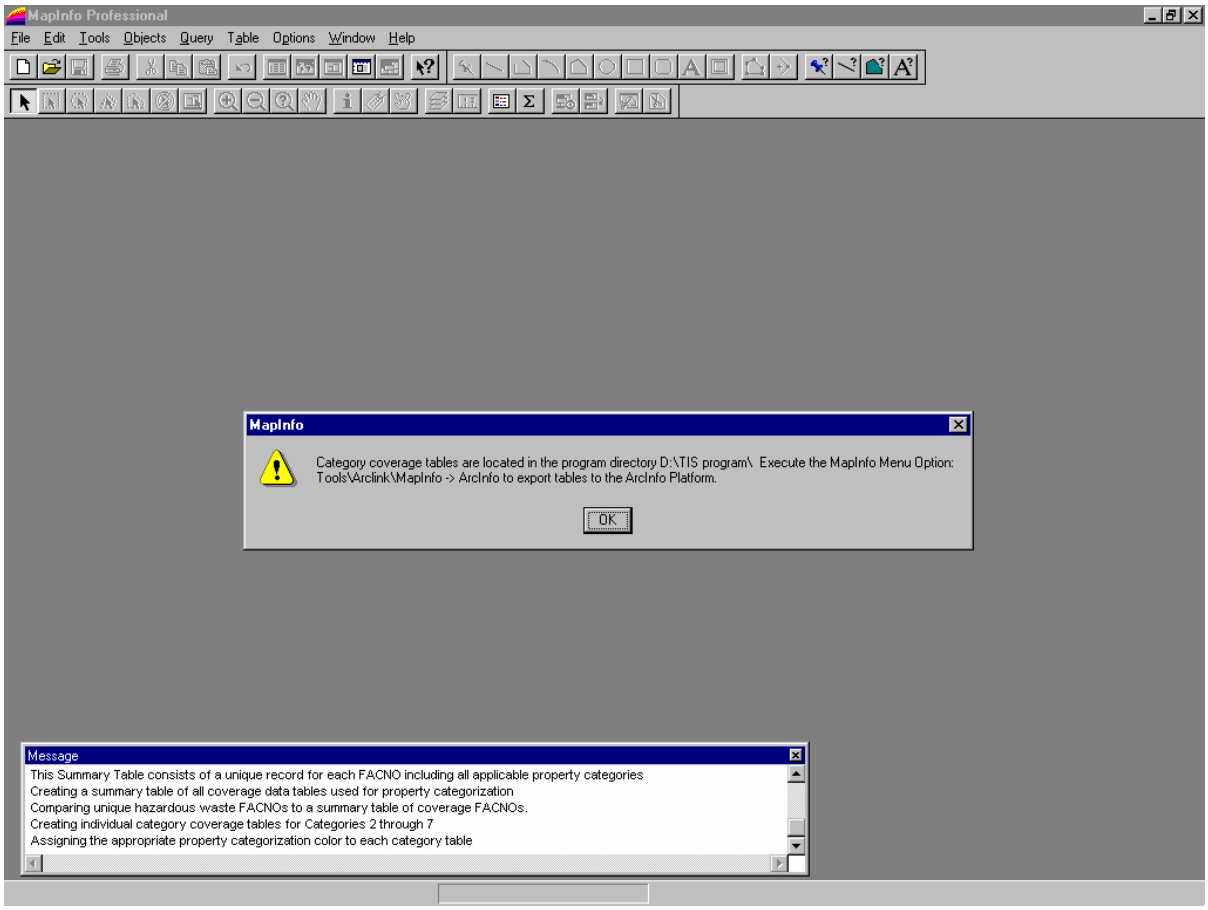


This screen contains two different browser windows. The browser labeled "covfacnos Browser" contains a list of all of the facnos and the tables they are in. The second window labeled "errorhazw Browser" shows the facnos that are located in the hazwi0.tab file but that do not have a corresponding facno in any of the valid tables. The user can use the facno listing to help identify if the facno is really not included in any of the valid tables or if the facno might have been mistyped originally in the hazwi0.tab.

Screen 1 of 2 telling the user that the recategorization has been completed successfully.



Screen 2 of 2 telling the user that the recategorization has been completed successfully.



APPENDIX F
PROJECT MANAGER'S GUIDE
MAPINFO EBS PROCESS NAMING CONVENTIONS FOR ALL COVERAGE/ ATTRIBUTE FILES USED IN THE EBS PROCESS

Table F-1 Air Force Plant 44	
Attribute/Coverage Files AFP44 (Tucson, Arizona)	**Files Used in Property Categorization**
ATTRIBUTE NAMES	
HAZWI0	●
COVERAGE NAMES	
Addp	●
Add2p	●
ASTSp	●
Boundp	●
Cascawp	●
Drywellp	●
Giwp	●
Leachp	●
Plume1p	●
Plume2p	●
Plume3p	●
Sites1p	●
Sites2p	●
Sites4p	●
Sites6p	●
Stormp	●
Strucp	●
Swmu1p	●
Swmu2p	●
Tunnelp	●
Ustsp	●
Attribute/Coverage Files AFP44 (Tucson, Arizona)	**Files Used in Property Categorization**
OTHER NAMES USED FOR EBS PROCESS	
COVNAME.TAB	●
CONDPROP	

APPENDIX G
MAPINFO EBS PROCESS
RECATGRY.MBX SOURCE CODE

```
'   Program written by Earth Tech, Inc.
'                      1420 King Street, Suite 600
'                      Alexandria, VA
'                           (703)549-8728
'   Earth Tech Point of Contact:  Len Fried, John Cooley
'   Program last modified by Earth Tech on 9/20/99
'
'   Program prepared for the United States Air Force
'                                 Aeronautical Systems Center
'                             Acquisition Environmental Management (ASC/EM)
'                                 Wright-Patterson Air Force Base, Ohio 45433
'
'   Program submitted to ASC/EM on 9/21/99
'   Program last modified on 9/20/99


'   This MapBasic program implements the Property Categorization Methodology established by:
Memorandum, Department of
'   Defense, 1996. Clarification of Uncontaminated Environmental Condition of Property at Base
Realignment and
'   Closure Installations, 21 October.


'   The program categorizes property on MapInfo coverages based on information stored in an
associated hazardous waste
'   table (HAZWI0.TAB). The program implements error trapping to halt program execution if errors
occur that would
'   prevent the completion of property categorization. In these cases, the user is provided
detailed error messages
'   to aid in correcting the error.


Include "Mapbasic.def"


' Declaration of procedures used by this program


Declare sub main
Declare sub initial_variables
Declare sub delete_old_program_files
Declare sub hazwaste_categorization
Declare sub summarytable_of_facno_propertycategories
Declare sub table_of_unique_facnos_from_propertycoverages
Declare sub compare_hazfacnos_to_propertycoverage_facnos
Declare sub construct_individual_category_coverages
Declare sub update_coverage_categories
Declare sub color_category_coverage
Declare sub create_summarytable_of_facno_propertycategories
Declare sub end_program_message
Declare sub apply_specific_color_to_coverage_facnos(ByVal file_path as string,ByVal catcolor as
integer)
Declare sub error_message(ByVal errorcode as smallint)
Declare sub setmaxcat(maxcat as string, currentcat as string)
Declare sub update_coverage_facno_categories(ByVal file_name as string)
Declare sub delete_file(ByVal file_pathname as string,ByVal filename as string)
Declare sub verifyColumn(ByVal ColumnName As String, ByVal TabName As String, VerifyFlag As
String)
Declare sub sql_select_invalid_hazcats
Declare sub sql_select_inv_facnos
Declare sub sql_select_missing_coverages
Declare sub close_all_open_tables
Declare sub cleanup_system_files
Declare sub map_category_coverages
Declare sub map_category_coverage(ByVal file_pathname as string, ByVal tablename as
string,makenewmapwindow as string)
Declare sub create_category_table(fullpath_tablename as string)


' Declaration of Public or Global variables


global file_path as string
global prog_directorypath as string
global hazw_file as string
global hazwsmry_file as string
global facnocats_file as string
global covname_file as string
global boundp_file as string
```

```
global bndry_file as string
global misscovs_file as string
global missfacno_file as string
global facnocovcats_file as string
global covfacnos_file as string
global tempcov_file as string
global cat2_file as string
global cat3_file as string
global cat4_file as string
global cat5_file as string
global cat6_file as string
global cat7_file as string

' Main program procedure

sub main
    call close_all_open_tables
    call initial_variables
    call delete_old_program_files
    call hazwaste_categorization
    call summarytable_of_facno_propertycategories
    call table_of_unique_facnos_from_propertycoverages
    call compare_hazfacnos_to_propertycoverage_facnos
    call construct_individual_category_coverages
    call color_category_coverage
    call update_coverage_categories
    call end_program_message
    call cleanup_system_files
    call map_category_coverages
end sub
```

4.3 ' The following procedure closes all open tables

```
sub close_all_open_tables
        close all
end sub
```

4.4 ' The following procedure initializes global program variables

```
sub initial_variables

    prog_directorypath = ApplicationDirectory$()
    hazw_file = prog_directorypath + "hazwi0.tab"
    hazwsmry_file = prog_directorypath + "hazwsmry.tab"

    facnocats_file = prog_directorypath + "facnocats.tab"

    covname_file = prog_directorypath + "covname.tab"
    boundp_file = prog_directorypath + "boundp.tab"
    bndry_file = prog_directorypath + "bndry.tab"
    misscovs_file = prog_directorypath + "misscovs.tab"
    missfacno_file = prog_directorypath + "missfac.tab"
    facnocovcats_file = prog_directorypath + "facnocovcats.tab"
    covfacnos_file = prog_directorypath + "covfacnos.tab"
```

```
    tempcov_file = prog_directorypath + “tempcov.tab”

    cat2_file = prog_directorypath + “cat2.tab”
    cat3_file = prog_directorypath + “cat3.tab”
    cat4_file = prog_directorypath + “cat4.tab”
    cat5_file = prog_directorypath + “cat5.tab”
    cat6_file = prog_directorypath + “cat6.tab”
    cat7_file = prog_directorypath + “cat7.tab”
end sub

‘ The following procedure deletes program tables that are created each time this program is executed.

‘ Essentially, this is a house-cleaning procedure that guarantees old program tables are deleted before

‘ the program starts a new EBS property categorization. The following tables are deleted: cat2.tab, cat3.tab,

‘ cat4.tab, cat5.tab, cat6.tab, cat7.tab, bndry.tab, hazwsmry.tab, facnocats.tab, misscovs.tab, missfac.tab,

‘ facnocovcats.tab, covfacnos.tab, tempcov.tab.

sub delete_old_program_files

    Print “File Maintainence: Deleting old program files”
    Dim file_cntr as smallint
    Dim file_spec as string
```

```
Dim file_drop as string
    Dim totalcategories as smallint
    file_cntr = 2
    totalcategories = 7
    Do while file_cntr <= totalcategories
    file_drop = "cat" + file_cntr
        file_spec = prog_directorypath + file_drop + ".tab"
        Call delete_file(file_spec,file_drop)
        file_cntr = file_cntr + 1
    Loop
    Call delete_file(bndry_file,"bndry")
    Call delete_file(hazwsmry_file,"hazwsmry")
    Call delete_file(facnocats_file,"facnocats")
    Call delete_file(misscovs_file,"misscovs")
    Call delete_file(missfacno_file,"missfac")
    Call delete_file(facnocovcats_file,"facnocovcats")
    Call delete_file(covfacnos_file,"covfacnos")
        Call delete_file(tempcov_file,"tempcov")
end sub

' This procedure receives the pathname and filename for a file. If the file exists,
' the procedure deletes the file.

sub delete_file(ByVal file_pathname as string,ByVal filename as string)
```

4.5 If FileExists(file_pathname) then

```
    Open table file_pathname
    drop table filename
    End if
End sub

' This procedure updates the EBS hazardous waste table according to the criteria for property
categorization.
' This procedure determines if the CATEGORY field exists in the hazardous waste table. If the
CATEGORY field
' exists, the procedure drops (deletes) the CATEGORY field and creates a new blank
CATEGORY field; if the
' CATEGORY field does not exists, the procedure creates a new blank CATEGORY field. Each
hazardous waste table
' record is evaluated according to the values stored in its HISTORY, STATUS, TYPE, and
```

RASTATUS fields. Based

‘ on the combination of values in these four fields, a specific property category (1 to 7) is assigned to the

‘ records CATEGORY field. If an invalid combination of values is stored in the HISTORY, STATUS, TYPE,

‘ and RASTATUS fields, an E (for error) is stored in the records CATEGORY field. Once all records in the

‘ hazardous waste table are evaluated for property categorization, if any errors occurred, the program halts and

‘ a Mapinfo table is displayed containing all hazardous waste table records in error (i.e., having an E in the

‘ CATEGORY field).

```
sub hazwaste_categorization

    Print “EBS Property Categorization Process for the EBS Hazardous Waste Table”
    Print “Each record in the EBS Hazardous Waste Table will be evaluated”
    Dim rowcntr as smallint

    Dim errorflag as string
    Dim verifycolumnflag as string


    OnError Goto Bad_open
    open table hazw_file as hazw
    OnError Goto 0
    call verifycolumn(“category”,hazw,verifycolumnflag)
    if verifycolumnflag = “No” then
    Alter table hazw(Add category Char (1))

    else

    Alter table hazw(Drop category)
    Alter table hazw(Add category Char (1))
    end if
```

errorflag = “No”

rowcntr = 1
fetch first from hazw

do while not EOT(hazw)

if (hazw.type = “HAZ” or hazw.type = “HAZW” or hazw.type = “POL” or hazw.type = “POLW” or hazw.type = “PES”

or hazw.type = “RAD” or hazw.type = “RADW”) and hazw.status = “Y” and hazw.history = “S” and ltrim$(hazw.rastatus) = “”

then update hazw set Category = “1” where rowid = rowcntr

elseif (hazw.type = “POL” or hazw.type = “POLW”) and hazw.status = “Y” and (hazw.history = “R” or hazw.history = “D”) and

(hazw.rastatus = “N” or hazw.rastatus = “C” or hazw.rastatus = “U” or hazw.rastatus = “F”)

then update hazw set Category = “2” where rowid = rowcntr

elseif (hazw.type = “HAZ” or hazw.type = “HAZW” or hazw.type = “PES” or hazw.type = “RAD” or hazw.type = “RADW”) and

hazw.status = “Y” and (hazw.history = “R” or hazw.history = “D”) and hazw.rastatus = “N”

then update hazw set Category = “3” where rowid = rowcntr

elseif (hazw.type = “HAZ” or hazw.type = “HAZW” or hazw.type = “PES” or hazw.type = “RAD” or hazw.type = “RADW”) and

hazw.status = “Y” and (hazw.history = “R” or hazw.history = “D”) and hazw.rastatus = “C”

then update hazw set Category = “4” where rowid = rowcntr

elseif (hazw.type = “HAZ” or hazw.type = “HAZW” or hazw.type = “PES” or hazw.type = “RAD” or hazw.type = “RADW”) and

hazw.status = “Y” and (hazw.history = “R” or hazw.history = “D”) and hazw.rastatus = “U”

then update hazw set Category = “5” where rowid = rowcntr

elseif (hazw.type = “HAZ” or hazw.type = “HAZW” or hazw.type = “PES” or hazw.type = “RAD” or hazw.type = “RADW”) and

hazw.status = “Y” and (hazw.history = “R” or hazw.history = “D”) and hazw.rastatus = “F”

then update hazw set Category = “6” where rowid = rowcntr

elseif (hazw.type = “HAZ” or hazw.type = “HAZW” or hazw.type = “PES” or hazw.type = “RAD” or hazw.type = “RADW”) and

hazw.status = “A” and (hazw.history = “S” or hazw.history = “R” or hazw.history = “D”) and ltrim$(hazw.rastatus) = “”

then update hazw set Category = “7” where rowid = rowcntr
else update hazw set Category = “E” where rowid = rowcntr

errorflag = “Yes”

end if
fetch next from hazw
rowcntr= rowcntr+ 1
loop

4.5.1 If errorflag = “Yes” then

Call error_message(1)
Call error_message(10)
Call error_message(11)
Commit table hazw
close all
Call sql_select_invalid_hazcats
end program

```
    Else    commit table hazw
            close all
```

End If

```
    Last_exit:     exit sub
‘               “This routine prevents the program from unintentionally executing the error
handlers”
    Bad_open:      Call error_message(2)
            end program
```

end sub

‘ This procedure is executed if errors occurred while updating the hazardous waste table for property

‘ categorization. This procedure displays a Mapinfo table containing all hazardous waste table records in

‘ error (i.e., having an E in the CATEGORY field).

sub sql_select_invalid_hazcats

4.6 OnError Goto Bad_open

```
open table hazw_file as hazw
    OnError Goto 0
    Select * from hazw where category = "E" into errorhazw
    Browse * from errorhazw
      Last_exit:       exit sub
'             "This routine prevents the program from unintentionally executing the error
handlers"
       Bad_open:      Call error_message(2)
              end program
end sub

' This procedure creates a summary table (HAZWSMRY.TAB) based on an evaluation of all the
hazardous waste
' table records. For each unique FACNO in the hazardous waste table, a unique record is created
in the summary table
' for each FACNO and all applicable categories (i.e., 1 through 7)
sub summarytable_of_facno_propertycategories
Print "Constructing a Summary Table of All Property Categories applicable to each Hazardous
Waste FACNO"
Print "This Summary Table consists of a unique record for each FACNO including all applicable
property categories"
    Dim facnohold as string
        Dim rowcntr as smallint
        Dim errorflag as string
        Dim fieldname as string
        Dim casecategory as string
        Dim maxcat as string
        Dim totalcategories as smallint

        OnError Goto Bad_open
        open table hazw_file as temphazw
        select * from temphazw into Hazw order by facno
        OnError Goto 0

    Call create_summarytable_of_facno_propertycategories
    errorflag = "No"
    totalcategories = 7
    rowcntr = 1
    fetch first from Hazw
    Do while not EOT(Hazw)
                                        If Hazw.facno = "" then
Fetch next from Hazw
Else
                                        facnohold = Hazw.facno
```

```
        maxcat = "0"
        Insert into hazwsmry(facno) values(facnohold)
    Do while not EOT(Hazw) and facnohold = Hazw.facno
    casecategory = Hazw.category
If val(casecategory) >= 1 and val(casecategory) <= totalcategories then
    fieldname = "cat"+ casecategory
Update hazwsmry set fieldname = casecategory where rowid = rowcntr
            Call setmaxcat(maxcat,casecategory)
                        Else     errorflag = "Yes"
                       End If
    Fetch next from Hazw
    Loop
        Update hazwsmry set highestcat = maxcat where rowid = rowcntr
        rowcntr = rowcntr + 1
    End if
    Loop
    If errorflag = "Yes" then
        close all
        Call error_message(3)
        End Program
    Else
        Commit table hazwsmry
        Close All
    End If
        Last_exit:     exit sub
'               This routine prevents the program from unintentionally executing the error
handlers"
        Bad_open:      Call error_message(2)
                       end program
end sub

' This procedure compares two string integers and returns the maximum value as a string
' in the first procedure parameter maxcat
sub setmaxcat(maxcat as string, currentcat as string)
        if val(maxcat) < val(currentcat)then
maxcat = currentcat
        End if
End sub
```

4.7 ' This procedure opens the COVNAME.TAB table that identifies all the coverages that should

' be used for the property categorization. Each record in COVNAME.TAB should contain the filename for

‘ a single coverage in the table’s COVERAGEFILENAME field. This procedure reads each coverage name

‘ from COVNAME.TAB, it opens the coverage table and inserts specific coverage data for each coverage FACNO

‘ into a summary table (BNDRY.TAB). Specifically, the summary table (BNDRY.TAB) contains a unique record

‘ for each FACNO with its AREA, PERIMETER, and the filename for its coverage. If a coverage cannot be opened,

‘ the coverage name is inserted into a error file (MISSCOVS.TAB). At the end of this procedure, if any coverages

‘ could not be opened, the user is informed, the program is halted, and the user is presented a error table

‘ displaying the coverages that could not be opened. The COVNAME.TAB table is also displayed to the user.

sub table_of_unique_facnos_from_propertycoverages

Print “Creating a summary table of all coverage data tables used for property categorization”

```
Dim file_spec as string
Dim file_name as string
Dim error_count as smallint
Dim verifycolumnflag as string

    OnError Goto Bad_open1
Open table covname_file as covname
OnError Goto Bad_open2
Open table boundp_file as boundary
OnError Goto 0

select * from boundary where facno <> “” into tempcovfile
Commit table tempcovfile as bndry_file
Close table tempcovfile

Close table boundary
Open table bndry_file as allcoverages
call verifycolumn(“coveragefilename”,allcoverages,verifycolumnflag)
if verifycolumnflag = “No” then
Alter table allcoverages(Add coveragefilename Char (120))

else

Alter table allcoverages(Drop coveragefilename)
Alter table allcoverages(Add coveragefilename Char (120))
end if
update allcoverages set coveragefilename = boundp_file
```

```
    create table misscovs (coveragefilename char(120)) file misscovs_file
    error_count = 0
    fetch first from covname

    Do while not EOT(covname)

    file_name = ltrim$(covname.coveragefilename)
    If file_name <> "" then
        file_spec = prog_directorypath + file_name
        OnError goto bad_open3
        open table file_spec as validcoveragefile
        OnError goto 0
        select area,perimeter,facno from validcoveragefile where facno <> "" into tempcovfile
        commit table tempcovfile as tempcov_file
                                            close table tempcovfile
close table validcoveragefile
                                            open table tempcov_file as tempcovfile
alter table tempcovfile(Add coveragefilename Char(120))
    update tempcovfile set coveragefilename = file_spec
    insert into allcoverages(area,perimeter,facno,coveragefilename) select
        area,perimeter,facno,coveragefilename from tempcovfile
    close table tempcovfile

    end if
    Get_Next_Coverage: fetch next from covname
    Loop
    If error_count > 0 then
    commit table misscovs

        call error_message(16)
        close all
        Call sql_select_missing_coverages
        end program
    else

    commit table allcoverages
    close all
    end if
        Last_exit:      exit sub
        Bad_open1:      Call error_message(4)
                        end program
        Bad_open2:      Call error_message(5)
    close all

    end program
```

4.8 Bad_open3: error_count = error_count + 1

```
                                            insert into misscovs(coveragefilename) values
                                            (file_spec)
Resume Get_Next_Coverage
```

end sub

‘ This procedure is executed if one or more coverages identified in COVNAME.TAB could not be opened. This procedure

‘ displays a Mapinfo table (MISSCOVS) containing all coverages that could not be opened.

sub sql_select_missing_coverages

OnError Goto Bad_open1
open table misscovs_file as misscovs
OnError Goto Bad_open2
Open table covname_file as covname
OnError Goto 0

*4.8.1 Select * from covname*

Browse * from covname

Select * from misscovs
Browse * from misscovs

```
   Last_exit:      exit sub
‘            “This routine prevents the program from unintentionally executing the error
handlers”
       Bad_open1:     Call error_message(15)
              end program
```

Bad_open2: Call error_message(4)

close all

end program

end sub

```
‘       This procedure creates the facnocats (FACNOCATS.TAB) table that contains all the records
in the
‘       hazardous waste summary table (HAZWSMRY.TAB) that have a category greater than 1. The
FACNOs in the
‘       facnocats table are compared against the summary table of coverage data (BNDRY.TAB) that
‘       consists of all FACNOs in all coverages specified in COVNAME.TAB (i.e. the file that
specifies
‘       all the coverages that are to be used for property categorization). If a FACNO in the
‘       facnocats table is not contained in the bndry table, then the FACNO is not defined on
‘       any coverage specified in COVNAME.TAB. This is an error since each FACNO in the
‘       hazardous waste table with a category greater than 1 (i.e., the facnocats table) must be
‘       in a single coverage. If this error occurs, the program is halted. The user is provided
with
‘       a table containing the hazardous waste table records that have FACNOs that are not in any
‘       coverage. Also, the program displays a master list of the coverages identified in
COVNAME.TAB
‘       and the FACNOs that occur in each coverage. The user must correct the hazardous waste
records
‘       or the coverages such that each FACNO in the hazardous waste table having a property
category
‘       greater than 1 matches a FACNO in one of the coverages identified in COVNAME.TAB
```

sub compare_hazfacnos_to_propertycoverage_facnos

Print “Comparing unique hazardous waste FACNOs to a summary table of coverage FACNOs.”

```
Dim facnohold as string
Dim missflag as string

    OnError Goto Bad_open1
open table hazwsmry_file as hazwsmry
Onerror goto 0

Select * from hazwsmry where highestcat <> "1" into facnocats
commit table facnocats as facnocats_file
close all
open table facnocats_file as facnocats
OnError Goto Bad_open2
open table bndry_file as allcoverages
OnError Goto 0
create table miss_fac (missfacno char(10)) file missfacno_file

missflag = "No"
fetch first from facnocats
Do while not EOT(facnocats)
    facnohold = facnocats.facno
    select facno from allcoverages where allcoverages.facno = facnohold
    If selectionInfo(sel_info_nrows) = 0 then
        insert into miss_fac(missfacno) values (facnohold)
        missflag = "Yes"
    end if
    fetch next from facnocats
loop
Commit table miss_fac
close all
```

4.8.2 If missflag = "Yes" then

```
Call error_message(7)

Call error_message(12)

Call error_message(13)
close all
Call sql_select_inv_facnos
end program
end if
    Last_exit:     exit sub
'             This routine prevents the program from unintentionally executing the error
handlers"
    Bad_open1:     Call error_message(8)
                   end program
    Bad_open2:     close all
Call error_message(6)
                                        end program
end sub
```

```
'       This procedure is executed if a hazardous waste FACNO having a property category greater
than 1 is not
'       in any coverage identified in COVNAME.TAB. The user will be provided a table containing
the
'       hazardous waste table records with FACNOs having a property category greater than 1 that
are not in any
'       coverage identified in COVNAME.TAB. Also, this procedure displays a master list of the
coverages identified
'       in COVNAME.TAB and the FACNOs in each coverage. The user must correct the hazardous waste
records
'       or the coverages such that each FACNO in the hazardous waste table having a property
category greater
'       than 1 matches a FACNO on one of the coverages identified in COVNAME.TAB
```

```
sub sql_select_inv_facnos
    OnError Goto Bad_open1
    open table missfacno_file as missfacnos
    OnError Goto Bad_open2
    open table hazw_file as Hazw
        OnError Goto 0
    Select * from hazw,missfacnos where hazw.facno = missfacnos.missfacno into errorhazw
    Browse * from errorhazw
    OnError Goto Bad_open3
    open table bndry_file as allcoverages
    OnError Goto 0
    Select coveragefilename, facno from allcoverages into covfacnos
    commit table covfacnos as covfacnos_file
        close table covfacnos
    close table allcoverages
    open table covfacnos_file as covfacnos
    Browse * from covfacnos
```

4.8.3 Last_exit: exit sub

```
'               "This routine prevents the program from unintentionally executing the error
handlers"
    Bad_open1:
    Call error_message(14)
    close all
    end program
    Bad_open2:
    Call error_message(2)
    close all
    end program
    Bad_open3:
Call error_message(6)
```

close all

end program
end sub

‘ This procedure creates the facnocovcats table that is the intersection of the

‘ summary table (facnocats.tab) of FACNOs with categories greater than 1 with the summary

‘ table (bndry.tab) of FACNOs in the coverages identified in COVNAME.TAB. Based on previous

‘ error checking, each FACNO in the facnocats table will match a FACNO in the bndry table.

‘ Once the facnocovcats table is created, it is used to create individual coverage

‘ tables (Cat2.tab to Cat7.tab) for property categories 2 through 7. Please note that only

‘ applicable property category coverages are created; a particular category coverage is not

‘ created if there are no hazardous waste records that evaluate to that property category.

4.9 Sub construct_individual_category_coverages

Print “Creating individual category coverage tables for Categories 2 through 7”

Dim sel_cnt as smallint

OnError Goto Bad_open1
open table facnocats_file as facnocats
OnError Goto Bad_open2
open table bndry_file as allcoverages
OnError Goto 0

Select * from allcoverages,facnocats where allcoverages.facno = facnocats.facno into facnocovcats

commit table facnocovcats as facnocovcats_file

Select * from facnocovcats where cat2 = “2” into tempselectiontable order by facno

call create_category_table(cat2_file)

Select * from facnocovcats where cat3 = “3” into tempselectiontable order by facno

call create_category_table(cat3_file)

Select * from facnocovcats where cat4 = “4” into tempselectiontable order by facno

call create_category_table(cat4_file)

4.10 Select * from facnocovcats where cat5 = “5” into tempselectiontable order by facno

call create_category_table(cat5_file)

4.11 Select * from facnocovcats where cat6 = “6” into tempselectiontable order by facno

call create_category_table(cat6_file)

4.12 Select * from facnocovcats where cat7 = “7” into tempselectiontable order by facno

call create_category_table(cat7_file)
close all
‘

4.12.1 Last_exit: exit sub

‘ This routine prevents the program from unintentionally executing the error handlers”

Bad_open1:

Call error_message(9)

end program

Bad_open2:

close all
Call error_message(6)
end program
end sub
sub create_category_table(fullpath_tablename as string)
if selectionInfo(sel_info_nrows) > 0 then
commit table tempselectiontable as fullpath_tablename
end if
close table tempselectiontable
end sub

```
‘       This procedure modifies the property category coverage tables such that the mappable
objects
‘       in each category table are assigned the color appropriate for that property category
‘     (i.e., category 2 = blue, category 3 = lightgreen, category 4 = dark green, category 5 =
yellow,
‘     category 6 = red, and category 7 = grey).
```

sub color_category_coverage

4.13 Print “Assigning the appropriate property categorization color to each category table”

Dim catblue as integer
Dim catlightgreen as integer
Dim catdarkgreen as integer
Dim catyellow as integer
Dim catred as integer

```
    Dim catgrey as integer

    catblue = rgb(176,255,255)
    catlightgreen = rgb(160,255,160)
    catdarkgreen =  rgb(0,244,0)
    catyellow = rgb(255,255,144)
    catred = rgb(255,128,128)
    catgrey = rgb(191,192,192)

    call apply_specific_color_to_coverage_facnos(cat2_file,catblue)
    call apply_specific_color_to_coverage_facnos(cat3_file,catlightgreen)
    call apply_specific_color_to_coverage_facnos(cat4_file,catdarkgreen)
    call apply_specific_color_to_coverage_facnos(cat5_file,catyellow)
    call apply_specific_color_to_coverage_facnos(cat6_file,catred)
    call apply_specific_color_to_coverage_facnos(cat7_file,catgrey)
end sub


‘ This procedure receives the filename for a specific category coverage table and the EBS property

‘ categorization color that is applicable to all the mappable objects in the category table.

‘ This procedure determines if the specific category table exists (a specific property category table will not

‘ exist if there are no hazardous waste records that evaluate to the property category). If the category

‘ table exists, it assigns the categorization color to each mappable object in the category table.

sub apply_specific_color_to_coverage_facnos(ByVal file_path as string, ByVal catcolor as integer)

    dim new_brush as brush
    dim cat_object as object
    dim rowcntr as smallint
```

4.13.1 If FileExists(file_path) then

```
    Onerror Goto bad_open

    open table file_path as categorytable
    Onerror Goto 0

    fetch first from categorytable
    rowcntr = 1
    new_brush = makebrush(2,catcolor,catcolor)
    Do while not EOT(categorytable)
        cat_object = categorytable.obj
```

```
        alter object cat_object info obj_info_brush,new_brush
        update categorytable set obj = cat_object where rowid = rowcntr
        fetch next from categorytable
        rowcntr = rowcntr + 1
    loop
    commit table categorytable
    close all
    end if
        Last_exit:      exit sub
        bad_open:       Note "Program was halted before property categorization was completed.
Could not open category table: " + file_path
                        end program
end sub

' This procedure updates the CATEGORY field in the individual coverages used for property
categorization

' that are identified in COVNAME.TAB. This procedure reads each coverage name from COVNAME.TAB,

' it opens the coverage table and updates the CATEGORY field in the coverage table based on
the property

' categorization. Specifically, the CATEGORY field for each FACNO in the coverage table is
assigned the

' highest property category identified in the summary table (HAZWSMRY.TAB) of all property
categories

' applicable to each FACNO. The HAZWSMRY.TAB was constructed based on the property
categorization for the

' hazardous waste table.

sub update_coverage_categories

    Dim filename as string

    OnError Goto Bad_open1
        Open table covname_file as covname
        OnError Goto 0


        fetch first from covname
        Do while not EOT(covname)
    filename = ltrim$(covname.coveragefilename)
    if filename <> "" then
                                        Call update_coverage_facno_categories(filename)
end if
                                        fetch next from covname
        Loop
        close all
```

```
        Last_exit:      exit sub
    Bad_open1:
    Call error_message(4)
    end program
end sub
```

4.15 ' This procedure is used to update the CATEGORY field of coverages that are used for property

```
' categorization (i.e., a coverage identified in COVNAME.TAB). The procedure receives the filename
' of a specific coverage, opens the coverage, and tests whether the coverage has a CATEGORY field.
' If the coverage does not have a CATEGORY field, this field is created. If the coverage does have a
' CATEGORY field, the CATEGORY field is dropped (deleted), and a new empty CATEGORY field is created.
' The CATEGORY field for each FACNO in the coverage table is assigned the highest property category
' identified in the summary table (HAZWSMRY.TAB) of all property categories applicable to each FACNO.
' The HAZWSMRY.TAB was constructed based on the property categorization for the hazardous waste table.
sub update_coverage_facno_categories(ByVal file_name as string)
    Dim file_spec as string
    Dim facnohold as string
    Dim fieldname as string
    Dim highestcategory as string
    Dim verifycolumnflag as string
    Dim rowcntr as smallint

    file_spec = prog_directorypath + file_name
    OnError goto bad_open1
    open table file_spec as validcoverage
    OnError goto bad_open2
    open table hazwsmry_file as hazwsmry
    OnError goto 0

    Call verifyColumn("category", validcoverage, verifycolumnflag)
    if verifycolumnflag = "No" then
    alter table validcoverage(Add category Char(1))
```

```
    else
    alter table validcoverage(Drop category)
    alter table validcoverage(Add category Char(1))
    end if
    rowcntr = 1
    fetch first from validcoverage
    Do while not EOT(validcoverage)
        if validcoverage.facno <> "" then
        facnohold = validcoverage.facno
    select highestcat from hazwsmry where hazwsmry.facno = facnohold into temphighcat
    If selectionInfo(sel_info_nrows) > 0 then
        highestcategory = temphighcat.highestcat
        update validcoverage set category = highestcategory where rowid = rowcntr
                                                    else
update validcoverage set category = "1" where rowid = rowcntr
end if
                                                    close table temphighcat
        end if
        fetch next from validcoverage
        rowcntr = rowcntr + 1
    loop
    Last_exit:  commit table validcoverage
    close table validcoverage
    close table hazwsmry
    exit sub
    Bad_open1:  close all
```

Note "The program was halted before property categorization was completed. Could not open coverage table:"+ validcoverage + ". Coverage tables need to be located in program directory " + prog_directorypath

```
    end program
```

4.16 Bad_open2: close all

```
    Call error_message(8)
    end program
end sub
```

' This procedure is executed if the categorization process was successfully completed using the hazardous waste table

' and the coverages identified in COVNAME.TAB. It informs the user that the property categorization process has been

‘ completed and informs the user that a map window displaying the categorization results will be displayed.

sub end_program_message

Note “The property categorization process has been successfully completed. The property categorization results will be displayed in a map window that contains all applicable category coverages (Cat2.tab to Cat7.tab). Only category coverages that are applicable to the property categorization are created.”

Note “Category coverage tables are located in the program directory “+ prog_directorypath + “ Execute the MapInfo Menu Option: Tools\Arclink\MapInfo -> ArcInfo to export tables to the ArcInfo Platform.”

end sub

‘ This procedure performs house cleaning by deleting temporary tables created
‘ by the program during the property categorization.

sub cleanup_system_files
Call delete_file(facnocats_file,”facnocats”)
Call delete_file(tempcov_file,”tempcov”)
end sub

‘ This procedure opens a map window and displays the results of the property categorization; specifically,

‘ the map window displays only the category coverages (i.e., Cat2.tab to Cat7.tab) that are applicable to the

‘ property categorization. A particular property category may not be applicable if there are no hazardous waste

‘ records that evaluate to that property category.

sub map_category_coverages

4.17 Dim MakeNewMapWindow as string

MakeNewMapWindow = “Yes”

call map_category_coverage(cat2_file,”cat2”,MakeNewMapWindow)
call map_category_coverage(cat3_file,”cat3”,MakeNewMapWindow)
call map_category_coverage(cat4_file,”cat4”,MakeNewMapWindow)
call map_category_coverage(cat5_file,”cat5”,MakeNewMapWindow)
call map_category_coverage(cat6_file,”cat6”,MakeNewMapWindow)
call map_category_coverage(cat7_file,”cat7”,MakeNewMapWindow)
set map zoom entire
end program
end sub

‘ This procedure receives the pathname and tablename for a mappable property category table.

‘ The procedure determines whether the specified table exists; a particular property category table

‘ may not be applicable (i.e. will not exist) if there were no hazardous waste records that evaluated

‘ to that property category. If a new map window has not been created to display the property categorization results,

‘ the procedure creates the map window and maps the table. If a new map window has been created to display the

‘ property categorization results, the procedure maps the table as a new layer in the existing map window.

sub map_category_coverage(ByVal file_pathname as string, ByVal tablename as string,MakeNewMapWindow as string)

4.18 If FileExists(file_pathname)and MakeNewMapWindow = "Yes" then

Open table file_pathname as tablename
    map from tablename Position(0,0) Max
MakeNewMapWindow = “No”
    elseif FileExists(file_pathname) and MakeNewMapWindow = “No” then
Open table file_pathname as tablename
    Add map layer tablename
    end if
end sub

‘ This procedure receives a column name and table name. It searches the table to verify

‘ that the column is in the table. The procedure returns either a “Yes” or “No” by its

‘ verifyflag parameter depending on whether the column is or is not in the table, respectively.

sub verifyColumn(ByVal ColumnName As String, ByVal TabName As String,VerifyFlag As String)

```
    Dim col_name As String
    Dim I as smallint

        VerifyFlag = “No”
    For I = 1 to NumCols(TabName)
        col_name = ColumnInfo(TabName,”col” + I, COL_INFO_NAME)
        If col_name = ColumnName then
                                            VerifyFlag = “Yes”
end if
        Next
end sub
```

‘ This procedure creates a summary table of each FACNO and all its applicable property categories

‘ based on the property categorization for the hazardous waste table

sub create_summarytable_of_facno_propertycategories

create table hazwsmry

4.19 (facno char(10),

Cat1 char(1),
Cat2 char(1),
Cat3 char(1),
Cat4 char(1),
Cat5 char(1),
Cat6 char(1),
Cat7 char(1),
HighestCat char(1)) file hazwsmry_file
end sub

‘ This procedure displays program error messages. It receives an integer that identifies a specific

‘ error and uses the MapInfo Note function to display the error to the user

sub error_message(ByVal errorcode as smallint)

Do Case errorcode
Case 1
Note “The program was halted before property categorization was completed. The hazardous waste table contained record(s) with invalid criteria for property categorization.” + “ All hazardous waste records that have invalid property categorization criteria will be displayed in a table for correction. “

4.20 Case 2

Note “The program was halted before property categorization was completed. Hazardous waste table “ + hazw_file + “ could not be opened! “ + “Please be sure that this table is in the program directory “ + prog_directorypath

4.21 Case 3

Note “A category value was encountered outside the valid range(1-7). Please review hazardous waste table “ + hazw_file + “ for invalid categories.”

4.22 Case 4

Note “The program was halted before property categorization was completed. Table “ + covname_file + “ could not be opened! This file contains the names of coverages used for property categorization. “ + “Please be sure that this table is in the program directory “ + prog_directorypath

4.23 Case 5

Note “The program was halted before property categorization was completed. The boundary file “ + boundp_file + “ could not be opened!” + “ Please be sure that this table is in the program directory “ + prog_directorypath

4.24 Case 6

Note “The program was halted before property categorization was completed. The spatial summary table “ + bndry_file + “ could not be opened” + “Please be sure that this table is in the program directory “ + prog_directorypath

4.25 Case 7

Note “The program was halted before property categorization was completed. The hazardous waste table contains one or more FACNOs that do not match the spatial coverage tables that are used for property categorization.”

4.26 Case 8

Note “The program was halted before property categorization was completed. The category summary table “ + hazwsmry_file + “ could not be opened” + “Please be sure that this table is in the program directory “ + prog_directorypath

4.27 Case 9

Note “The program was halted before property categorization was completed. The category summary table “ + facnocats_file + “ could not be opened” + “Please be sure that this table is in the program directory “ + prog_directorypath

4.28 Case 10

Note “Note that the hazardous waste records in error have an E (for error) in the CATEGORY field; however, the user does not edit the CATEGORY field. Corrections must be made to the HISTORY, STATUS, TYPE, and/or RASTATUS fields since these fields are used to determine property categorization. “

4.29 Case 11

Note “Once corrections have been made to the hazardous waste records in error, save the corrected table and re-run the categorization program.”

4.30 Case 12

Note “A table will display the hazardous waste records having FACNOs that do not match any FACNO in the coverages used for property categorization. An additional table will summarize all FACNOs in each of the coverages used for the property categorization.”

4.31 Case 13

Note “Please correct all discrepencies such that the hazardous waste FACNOs match a FACNO in one of the coverages used for property categorization. Once corrected, re-rerun the property categorization program.”

4.32 Case 14

Note “The program was halted before property categorization was completed. The missing facnos table “ + missfacno_file + “ could not be opened” + “Please be sure that this table is in the program directory “ + prog_directorypath

4.33 Case 15

Note “The program was halted before property categorization was completed. The missing coverages table “ + misscovs_file + “ could not be opened” + “Please be sure that this table is in the program directory “ + prog_directorypath

4.34 Case 16

Note “The program was halted before property categorization was completed. One or more coverage tables that are used for property categorization could not be opened. The user will be displayed an error table (MISSCOVS) of the coverages that could not be opened.”

Note “All coverages that are used for property categorization are identified in table: “ + covname_file + “. This table will also be displayed to the user.”

Note “All coverages used for property categorization must be located in the program directory “ + prog_directorypath

4.35 End Case

end sub

Appendix H
Wilbur's System Administrator Guide
ARC/INFO EBS Data Processing System Screen Captures

Figure 1. Screen 1 of 1 showing the first menu list after running the "runtis" command

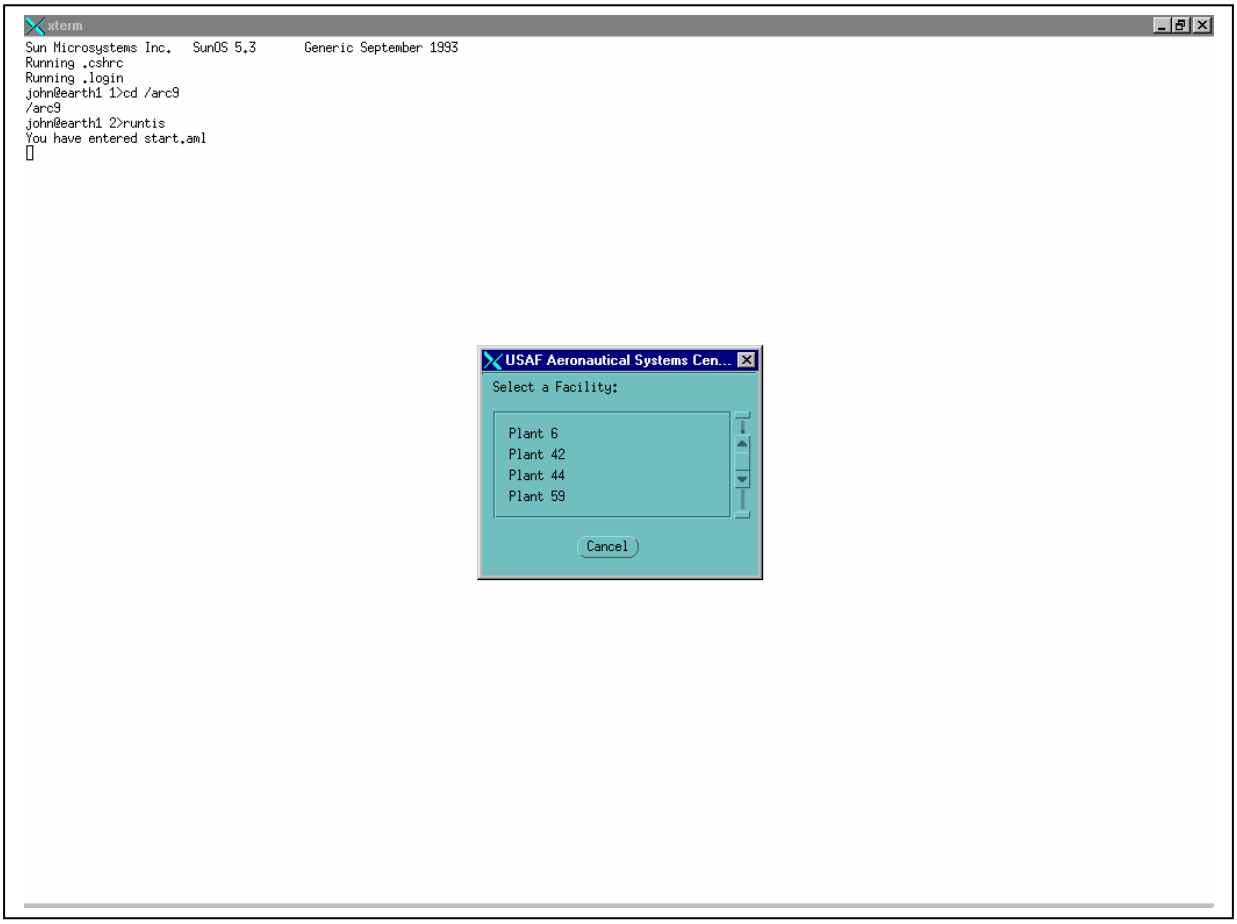


Figure 2. Screen 1 of 3 showing the TIS GUI and menu options

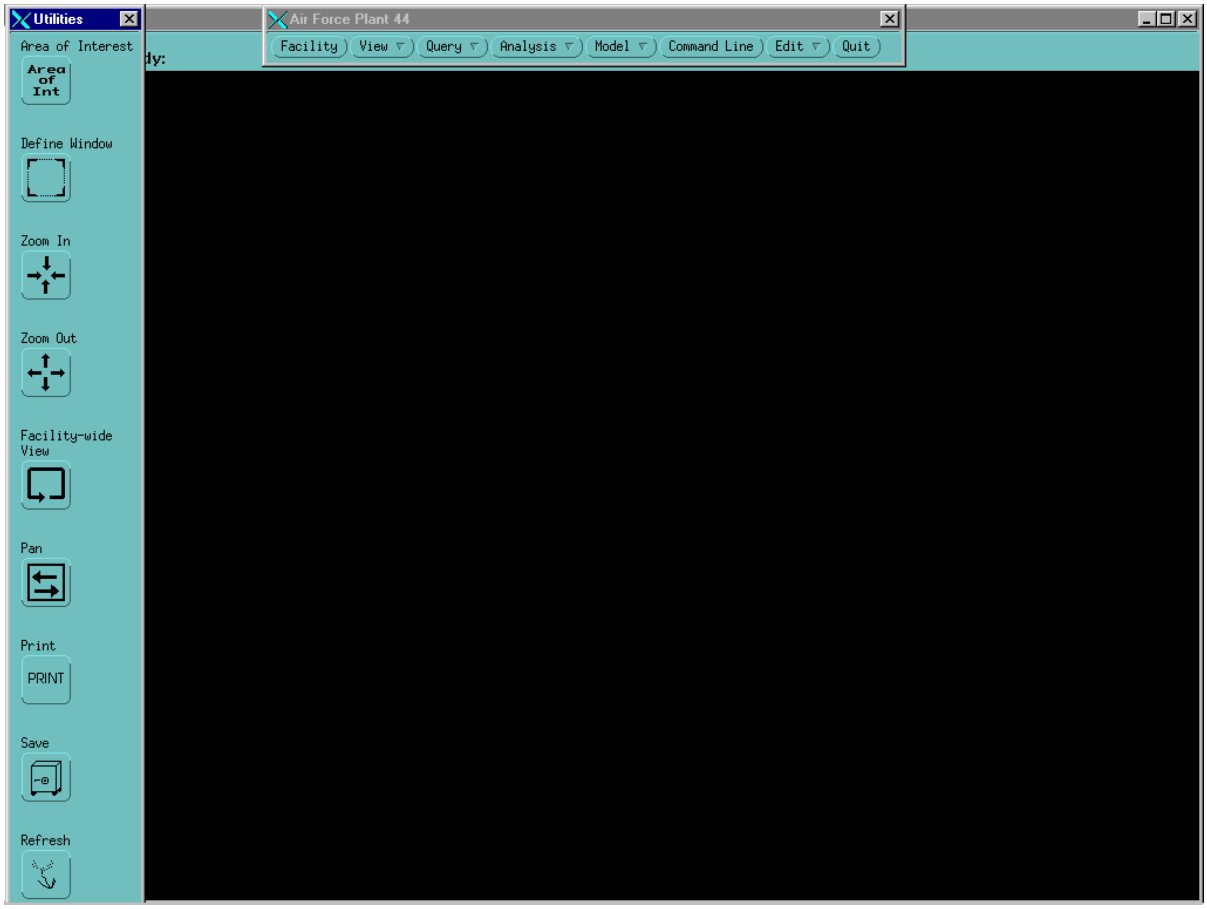


Figure 3. Screen 2 of 3 showing the TIS GUI and menu options including how to get to the EBS Processing menu

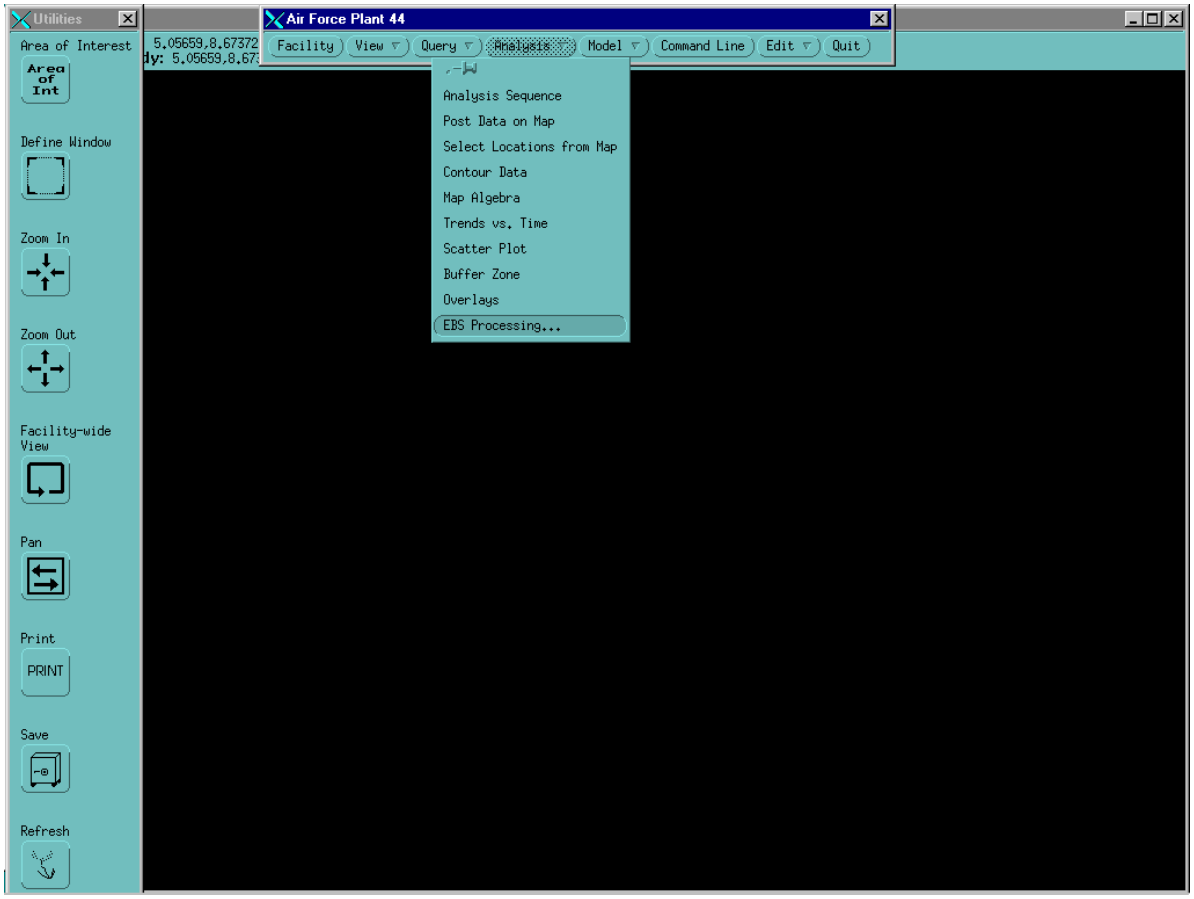


Figure 4. Screen 3 of 3 showing the TIS GUI and the EBS Operations menu options

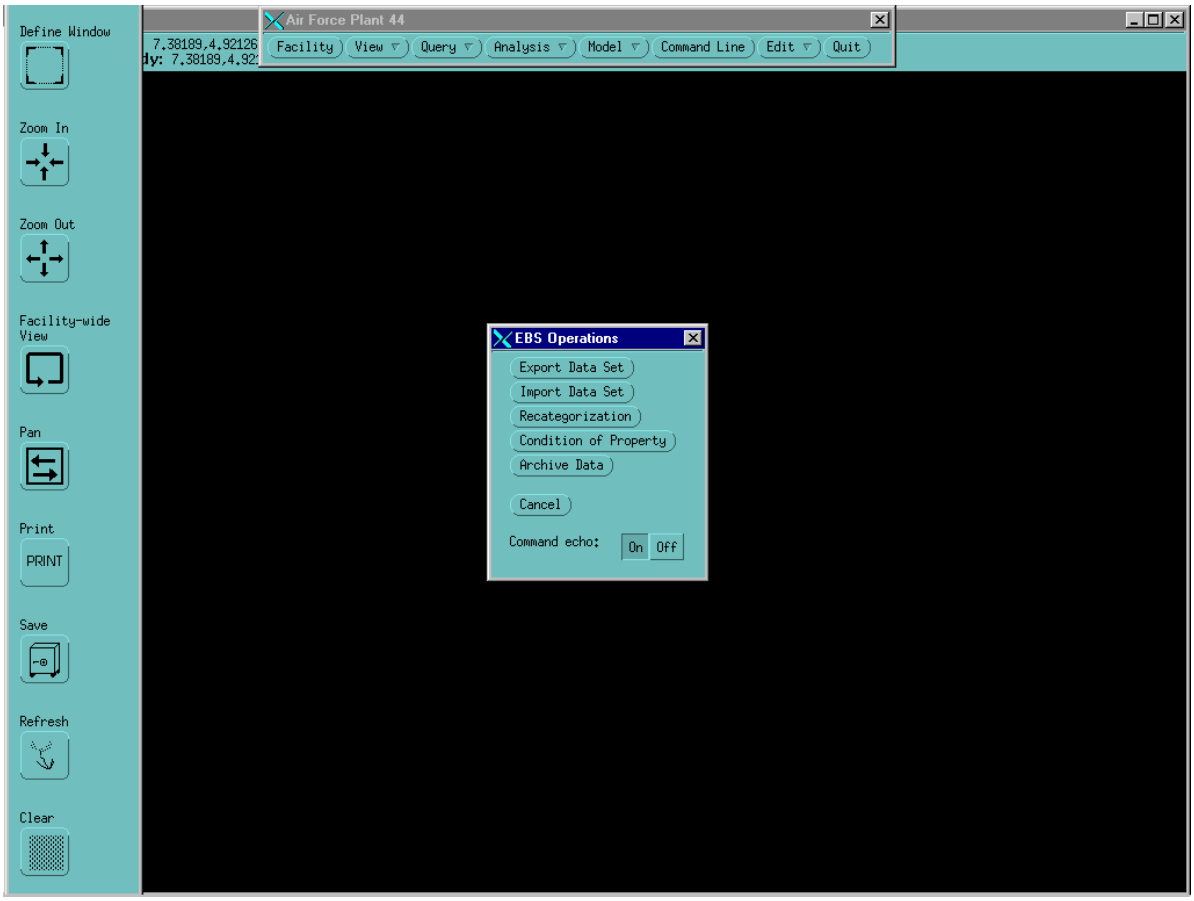


Figure 5. Screen 1 of 3 showing the process for a successful recategorization

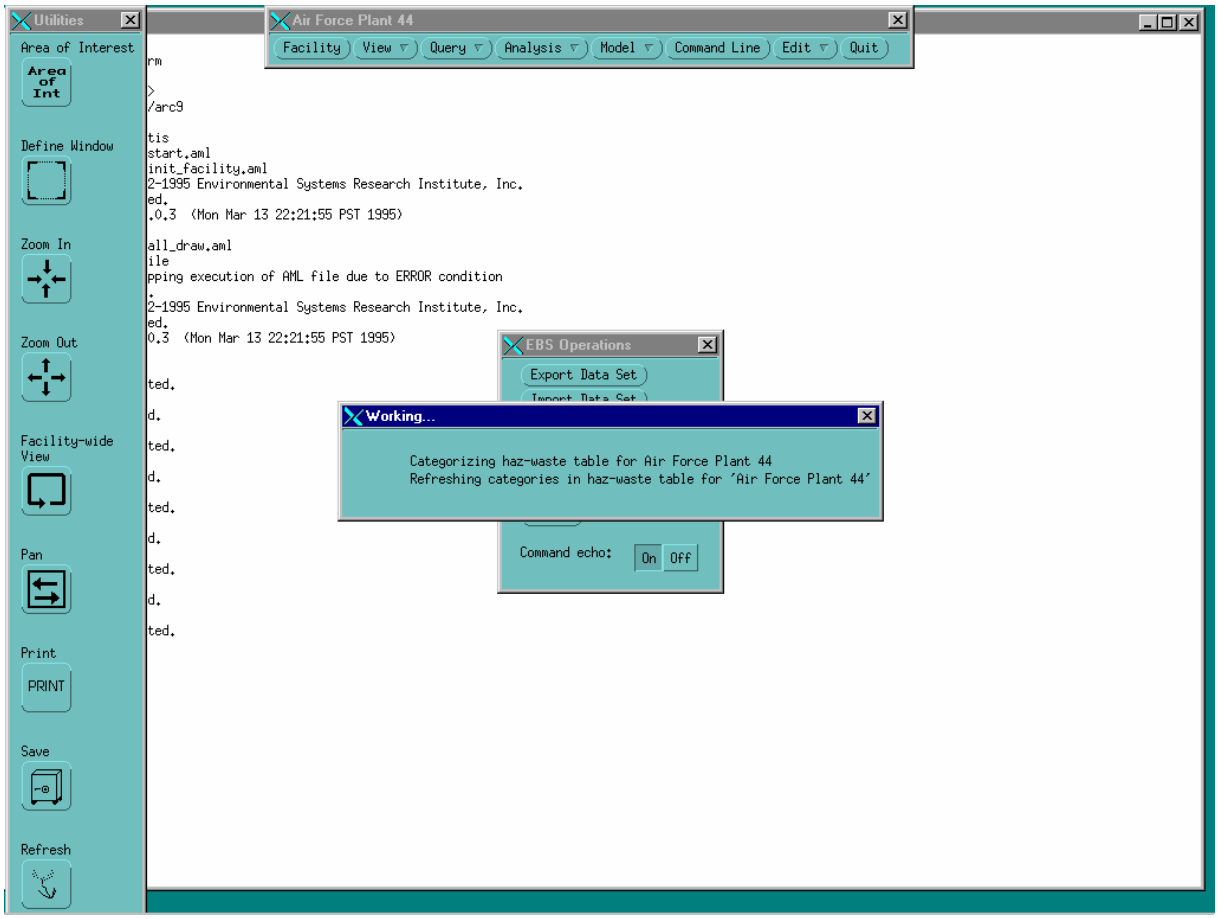


Figure 6. Screen 2 of 3 showing the process for a successful recategorization

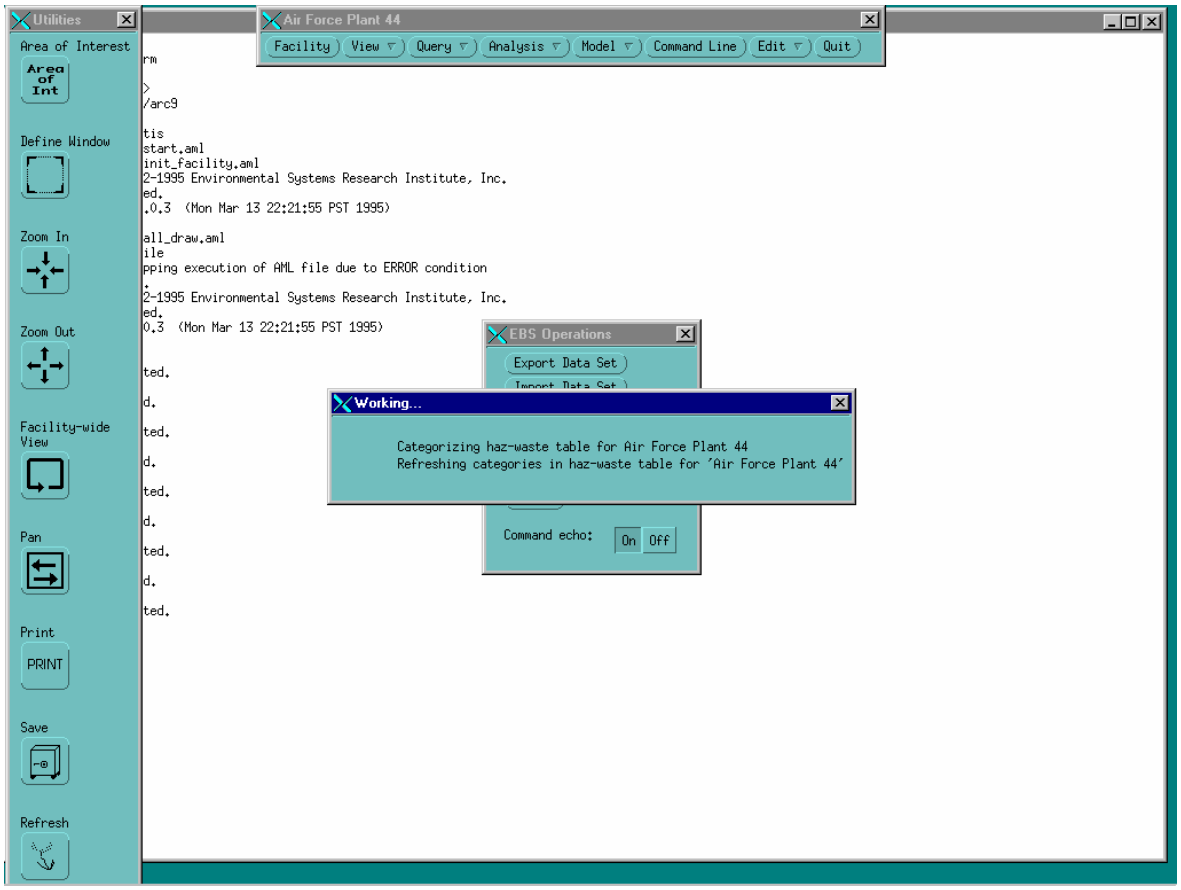


Figure 7. Screen 3 of 3 showing the process for a successful recategorization

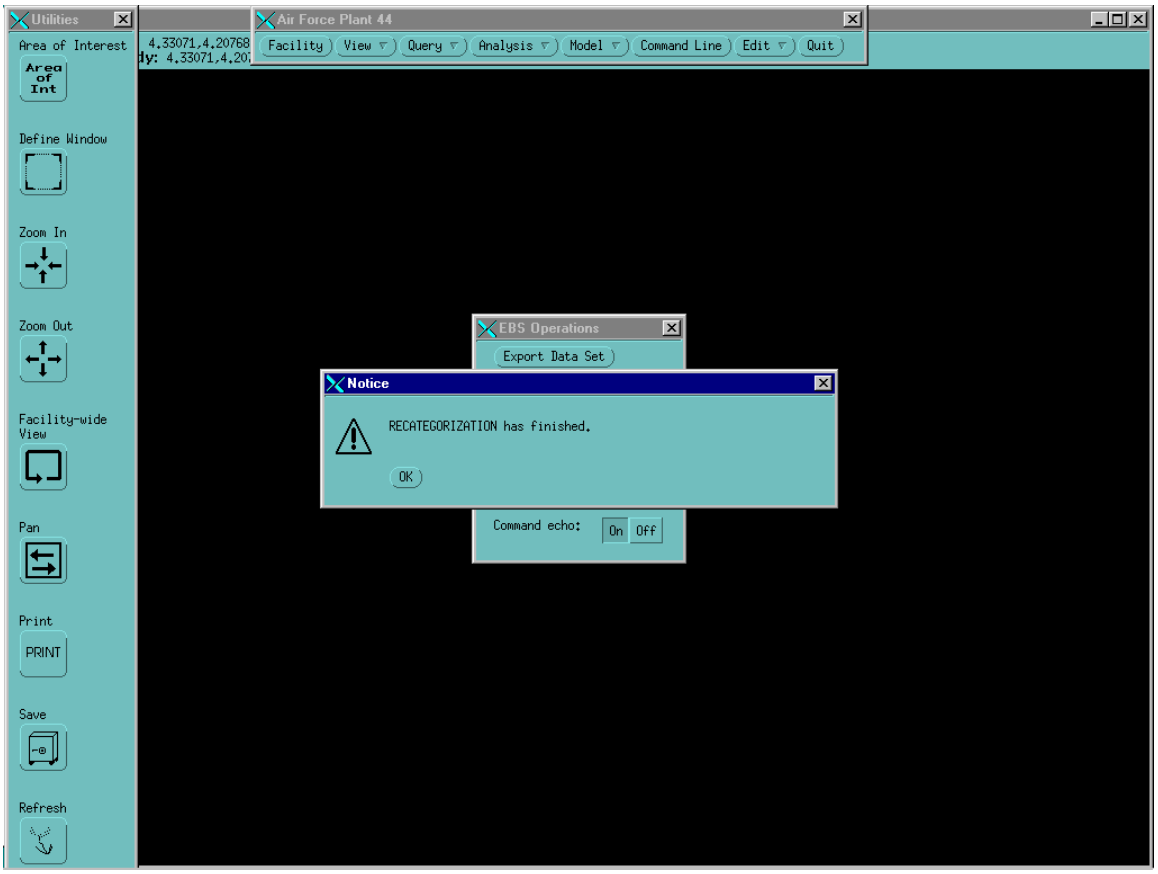


Figure 8. Screen 1 of 3 showing the process for a successful creation of
the condition of property coverage

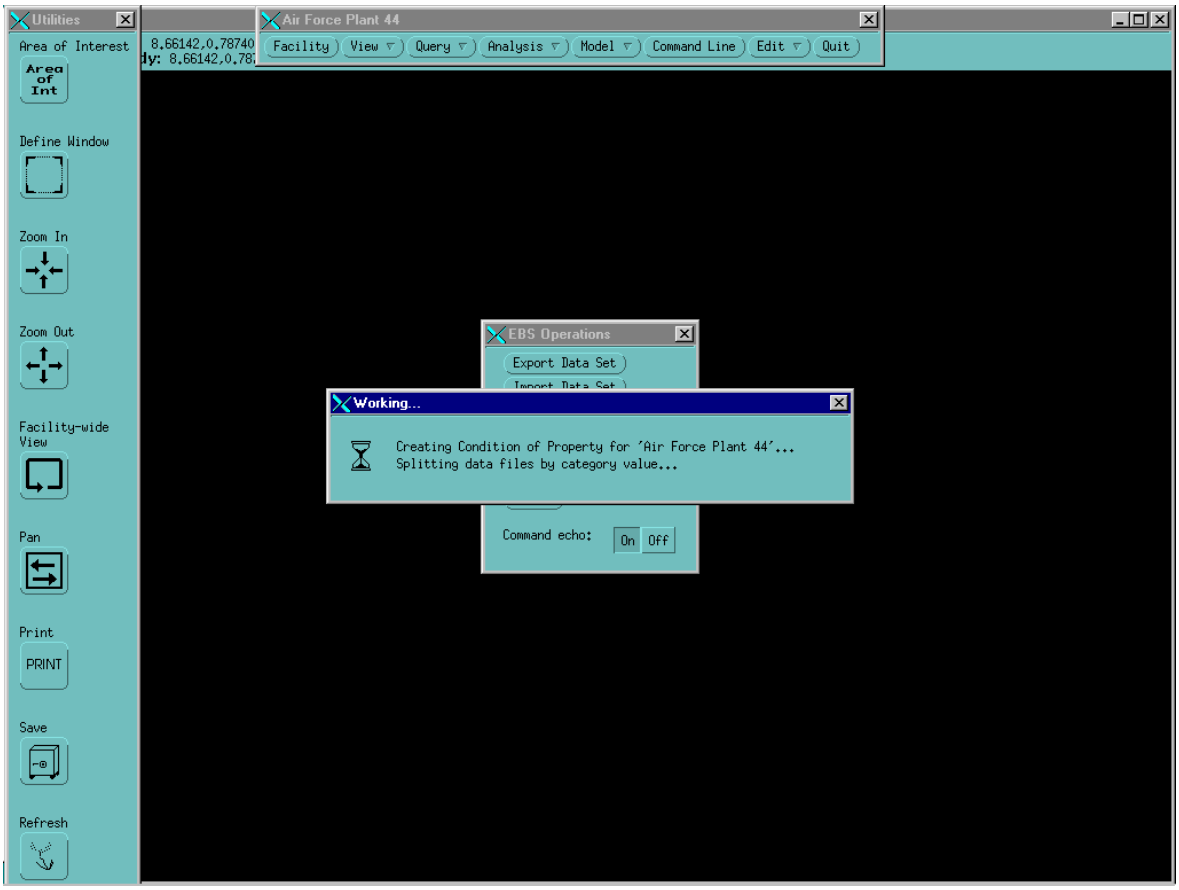


Figure 9. Screen 2 of 3 showing the process for a successful creation of
the condition of property coverage

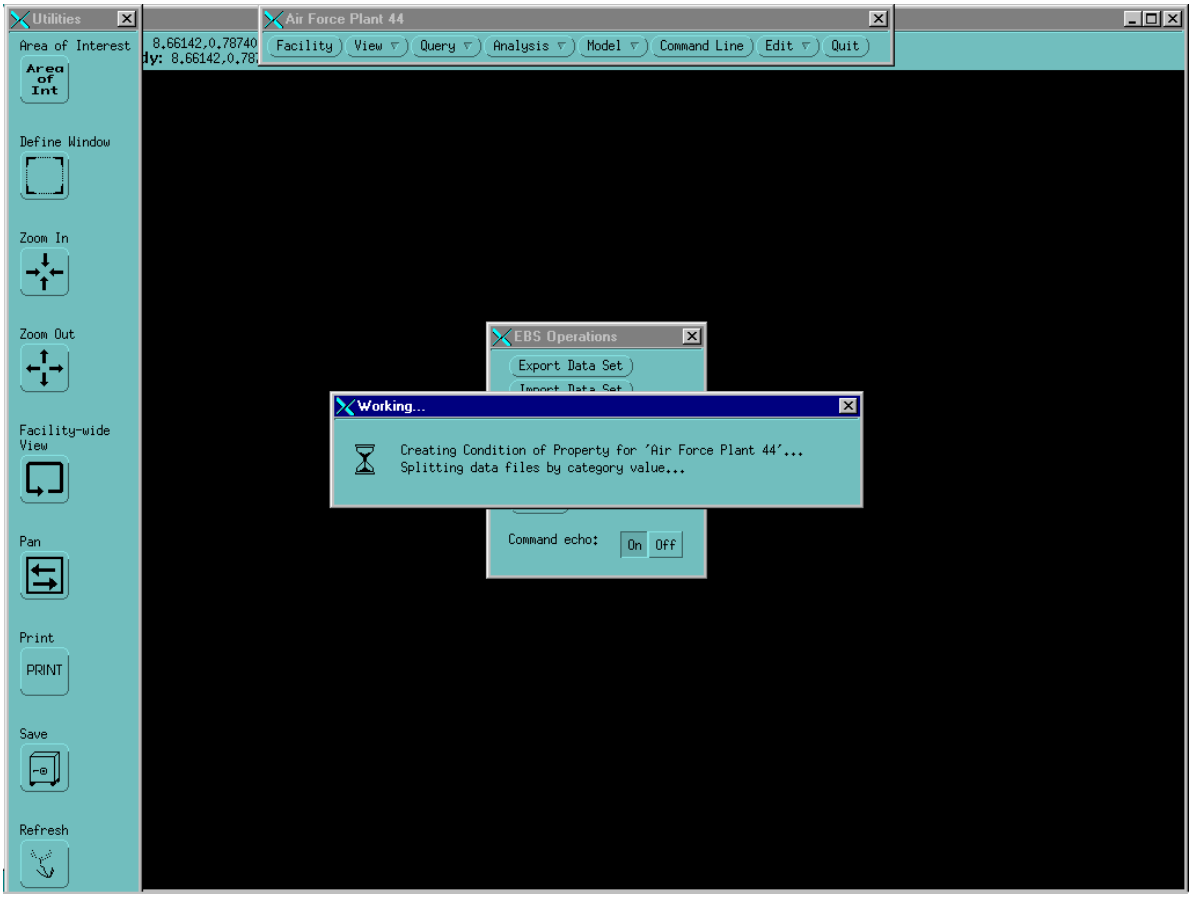


Figure 10. Screen 3 of 3 showing the process for a successful creation of the condition of property coverage

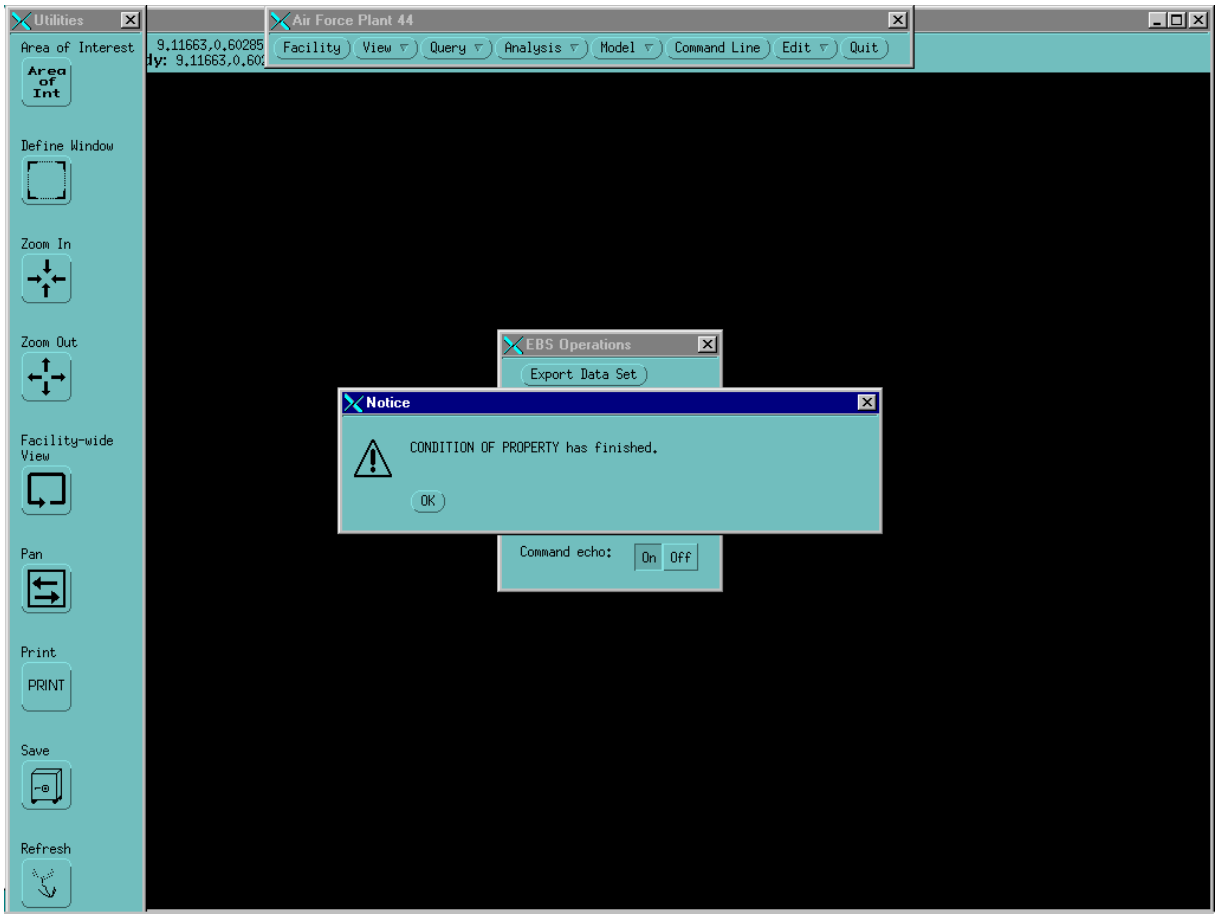


Figure 11. Screen 1 of 4 showing the process for a successful export of coverages

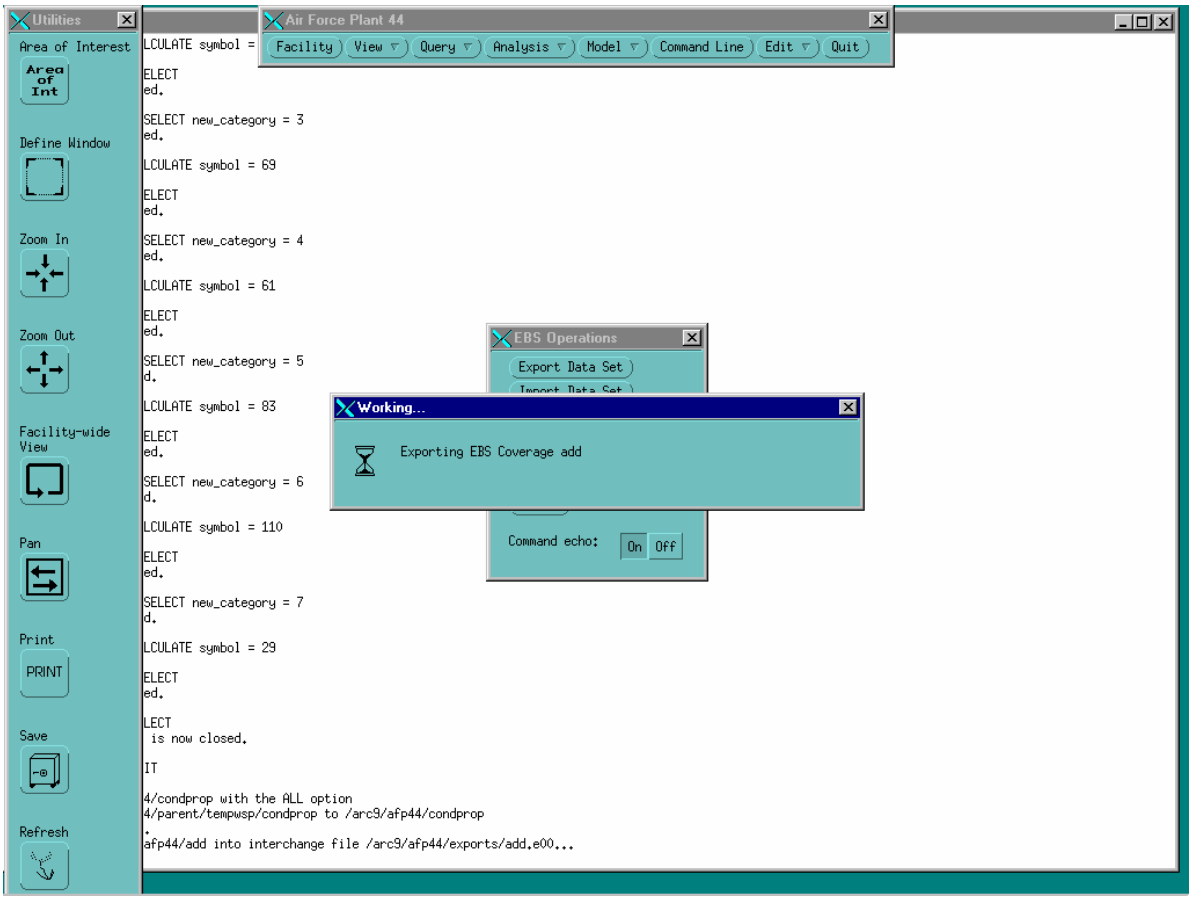


Figure 12. Screen 2 of 4 showing the process for a successful export of coverages

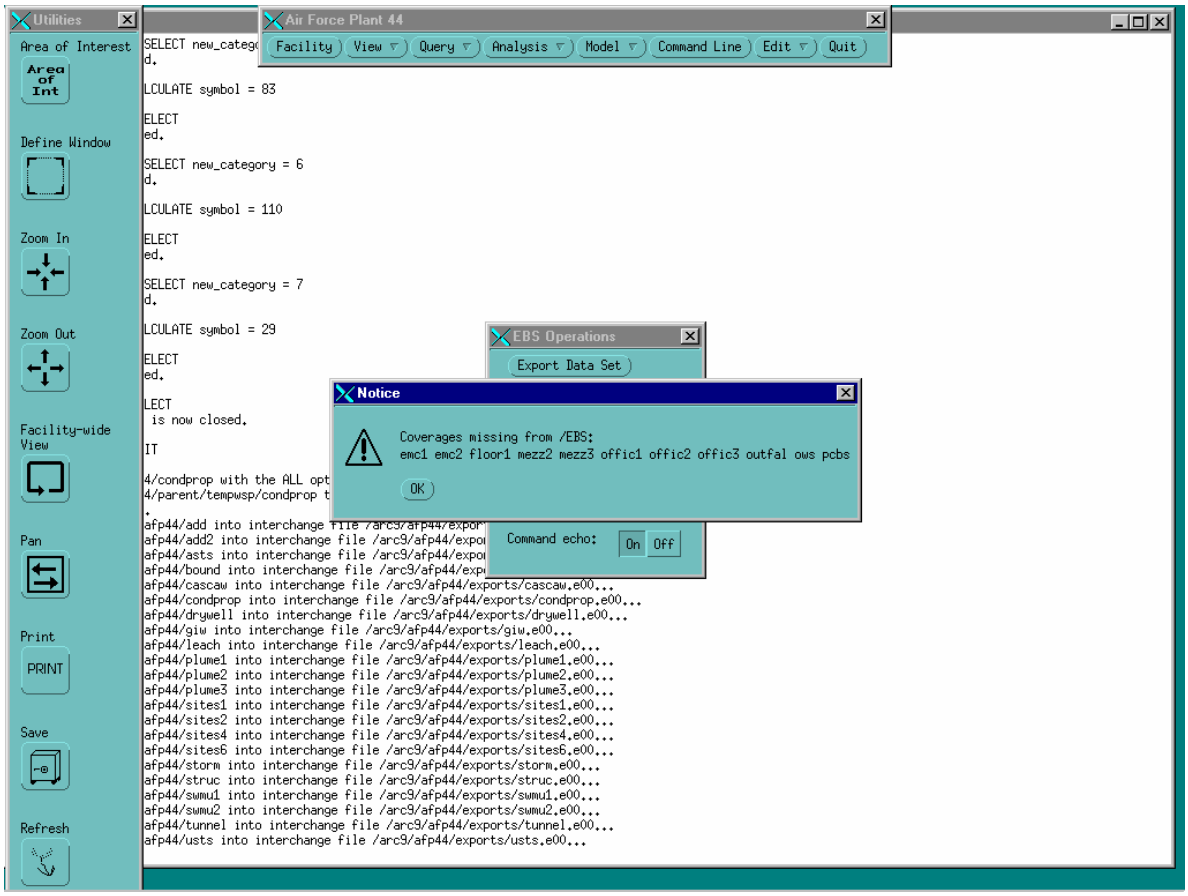


Figure 13. Screen 3 of 4 showing the process for a successful export of coverages

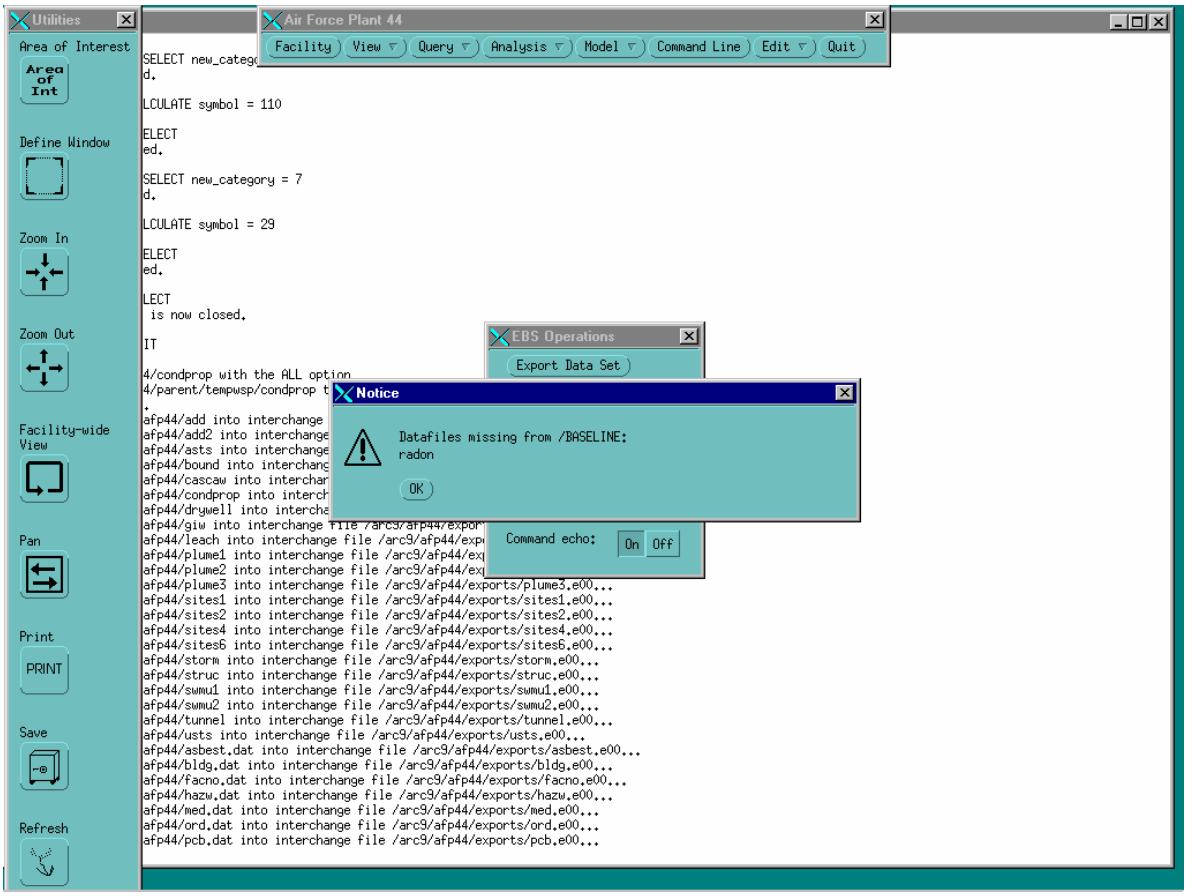


Figure 14. Screen 4 of 4 showing the process for a successful export of coverages

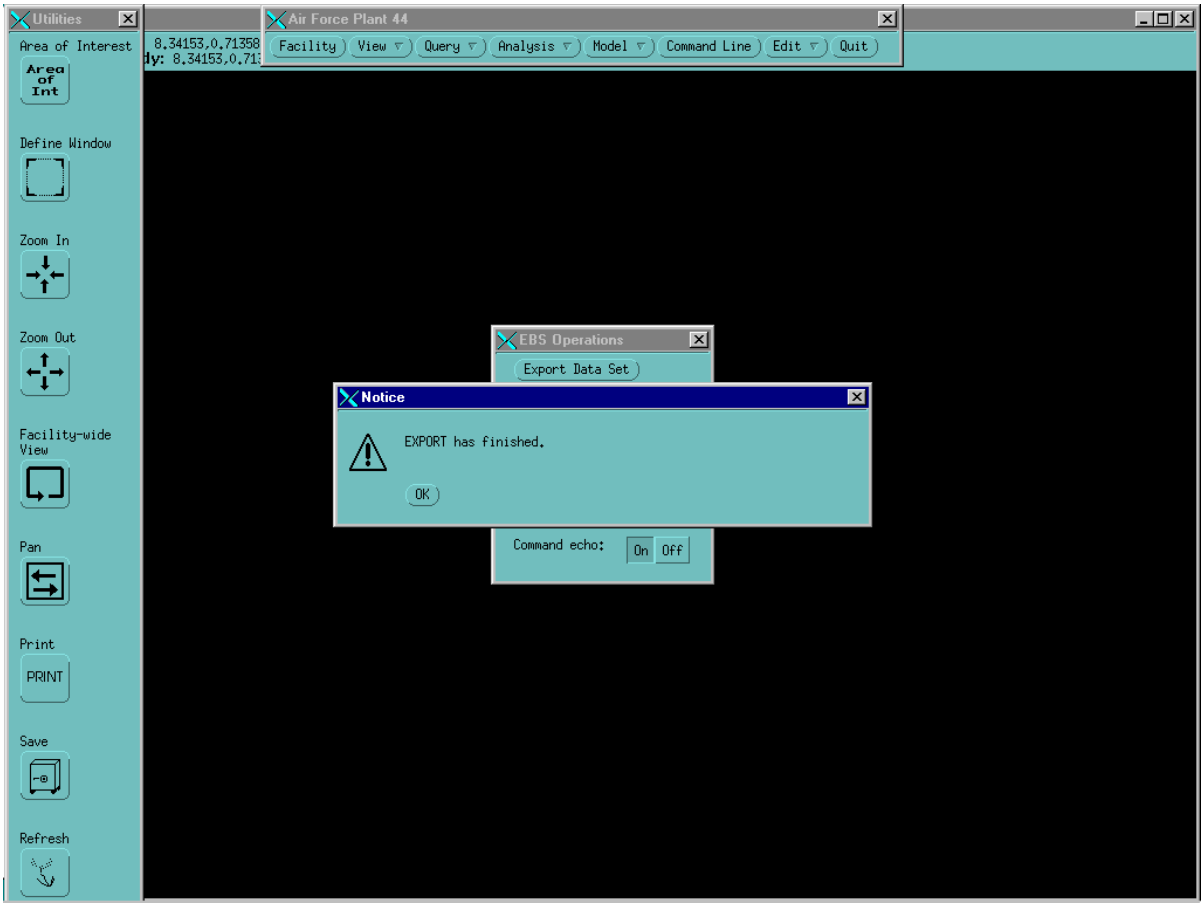


Figure 15. Screen 1 of 4 showing the process for a successful import of coverages and dBase files

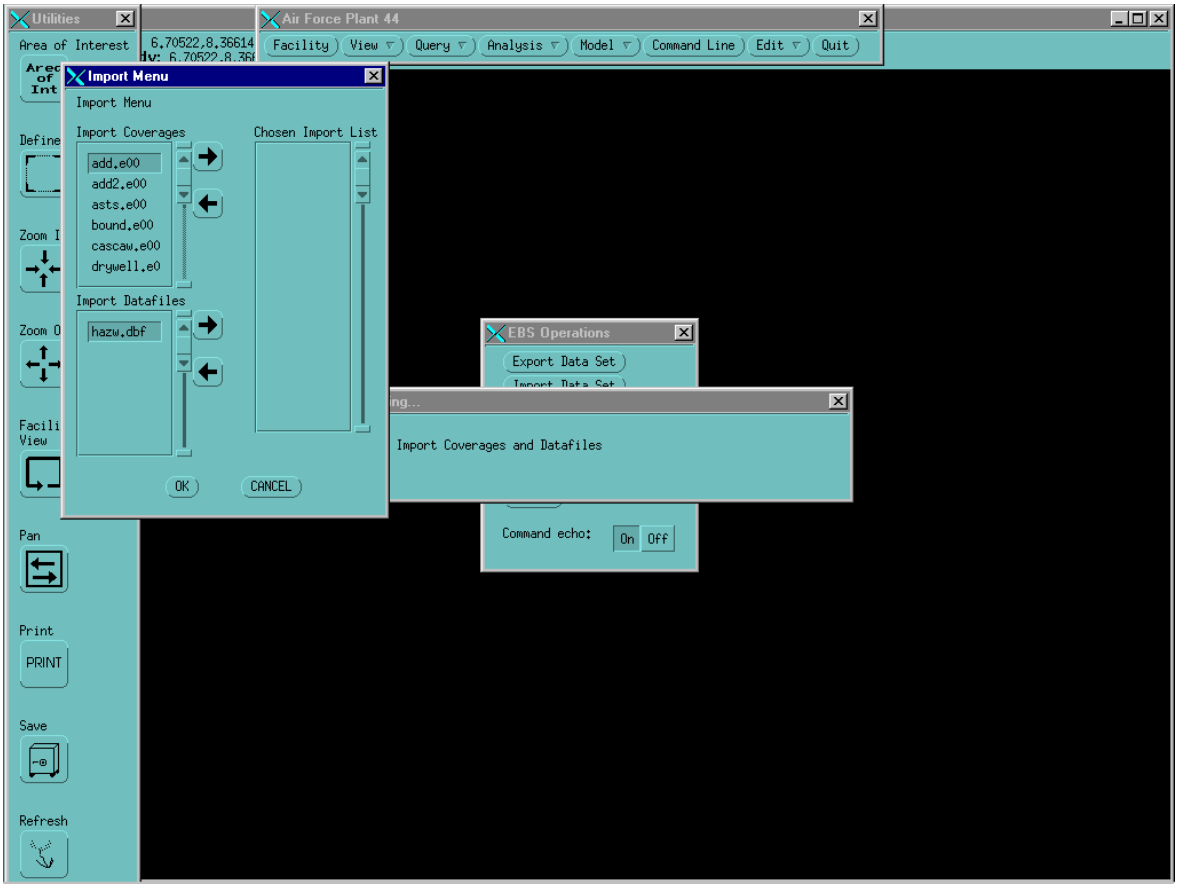


Figure 16. Screen 2 of 4 showing the process for a successful import of coverages and dBase files

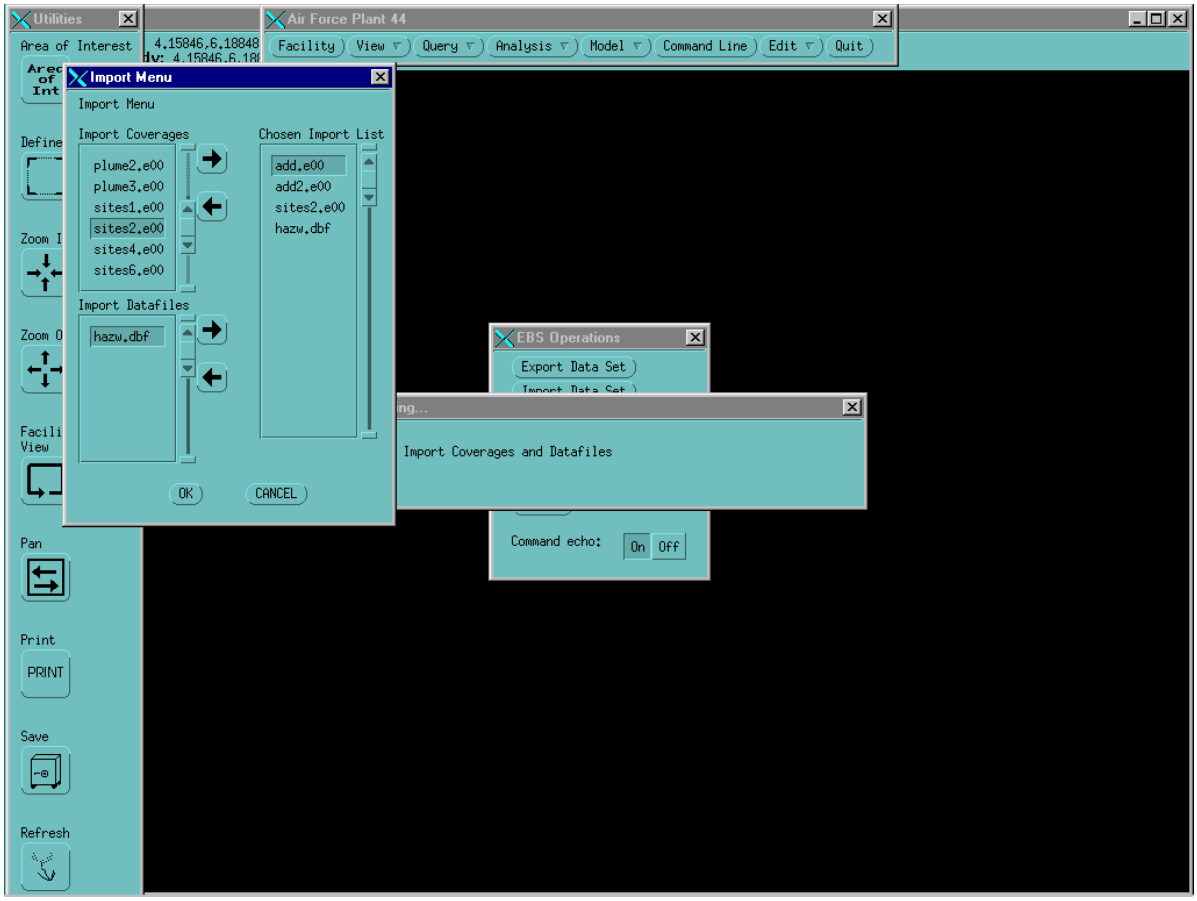


Figure 17. Screen 3 of 4 showing the process for a successful import of coverages and dBase files

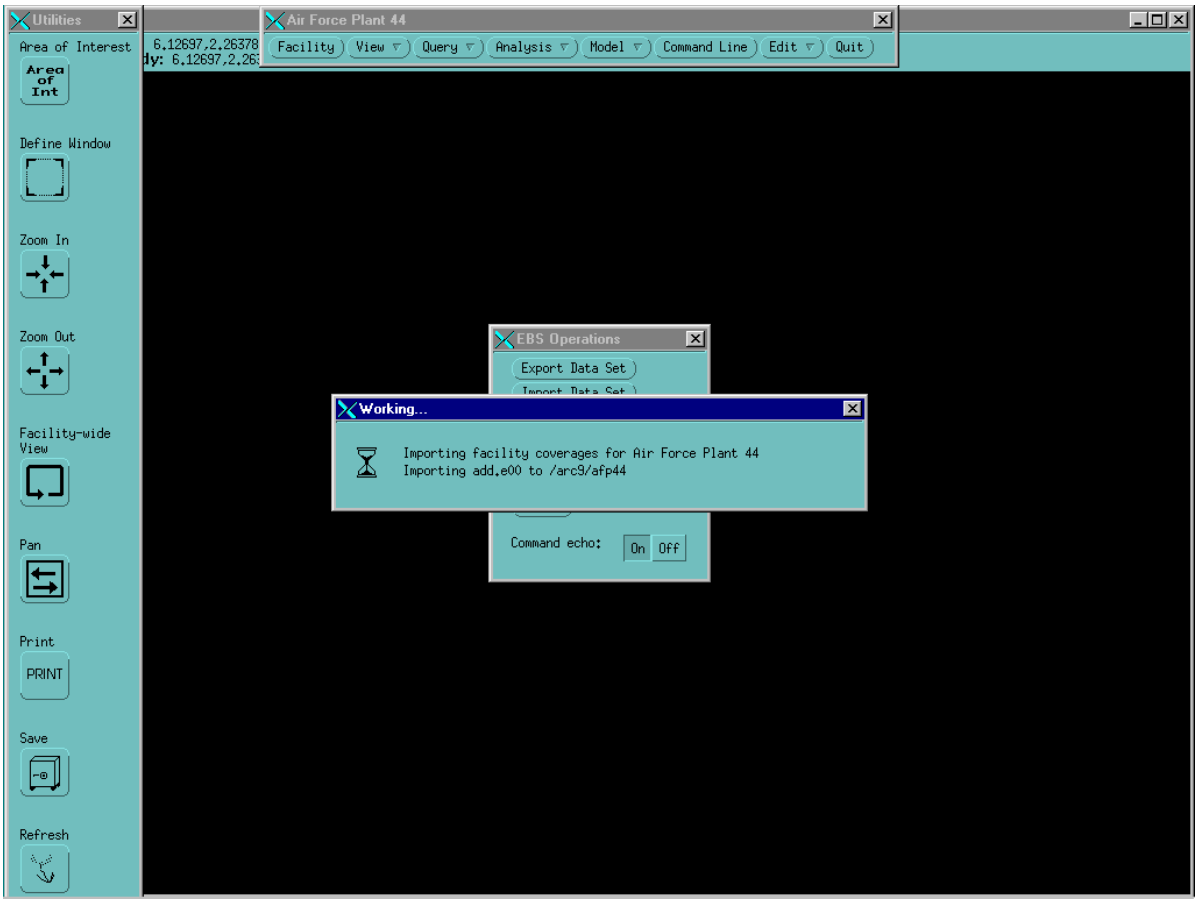


Figure 18. Screen 4 of 4 showing the process for a successful import of coverages and dBase files

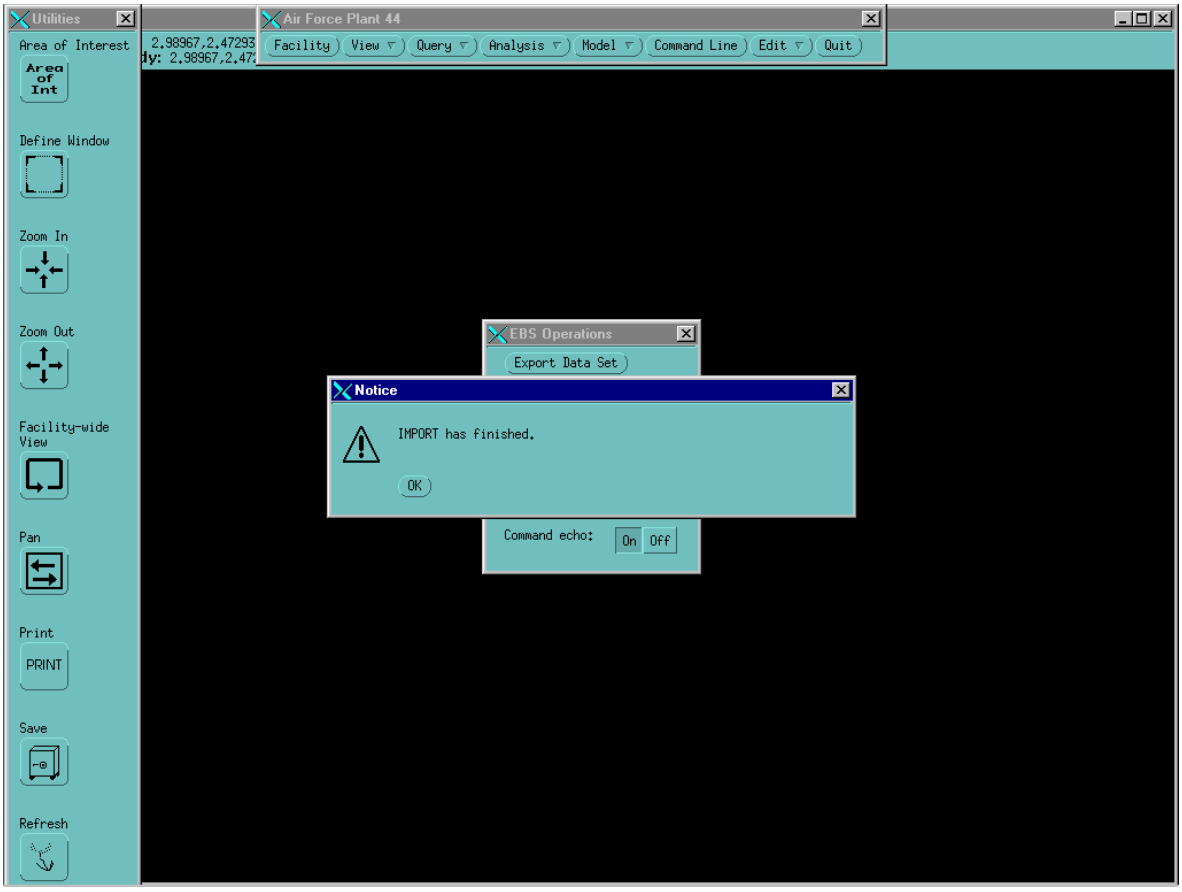


Figure 19. Screen 1 of 2 showing the screen as it appears when invalid field entries for the HISTORY, STATUS, RASTATUS, and TYPE fields are encountered

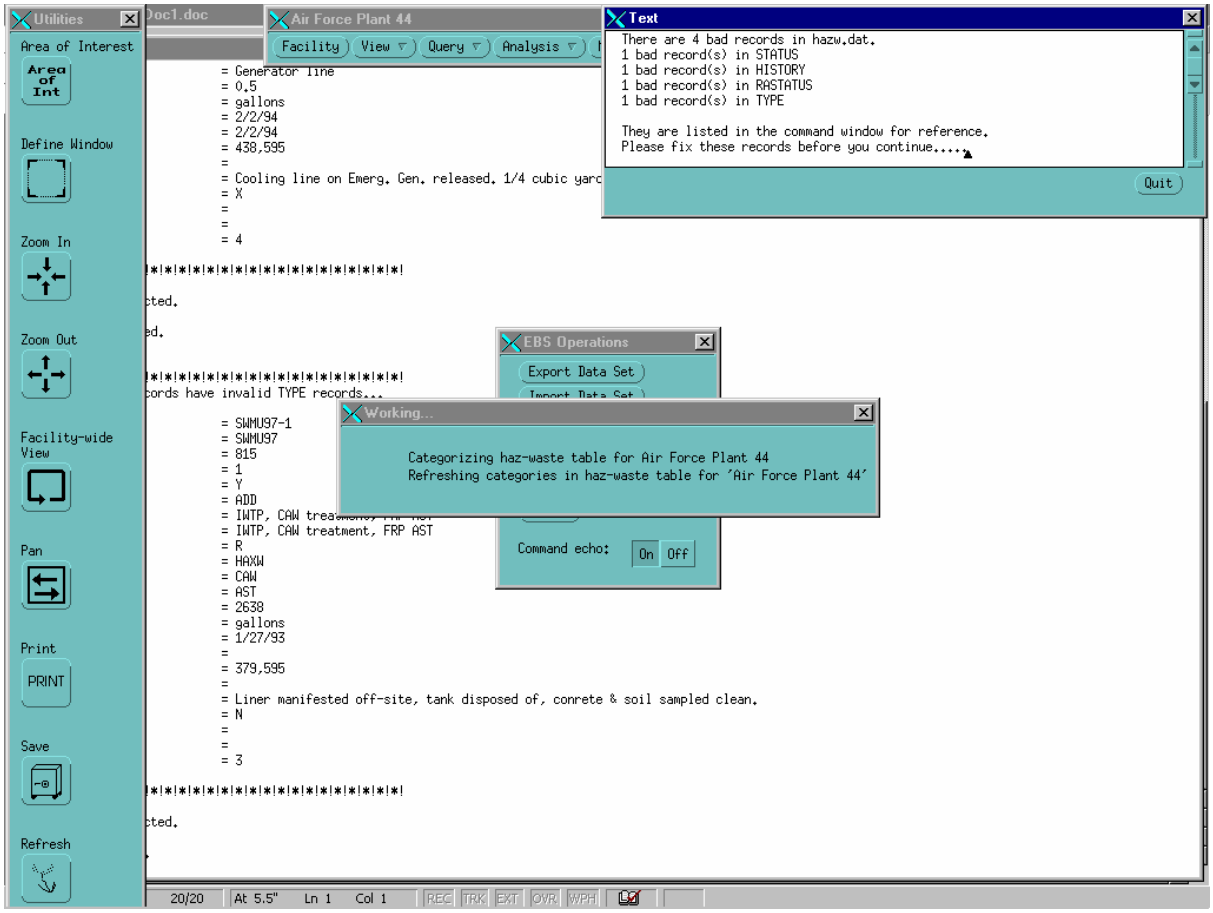


Figure 20. Screen 2 of 2 showing the screen as it appears when invalid field entries for the HISTORY, STATUS, RASTATUS, and TYPE fields are encountered

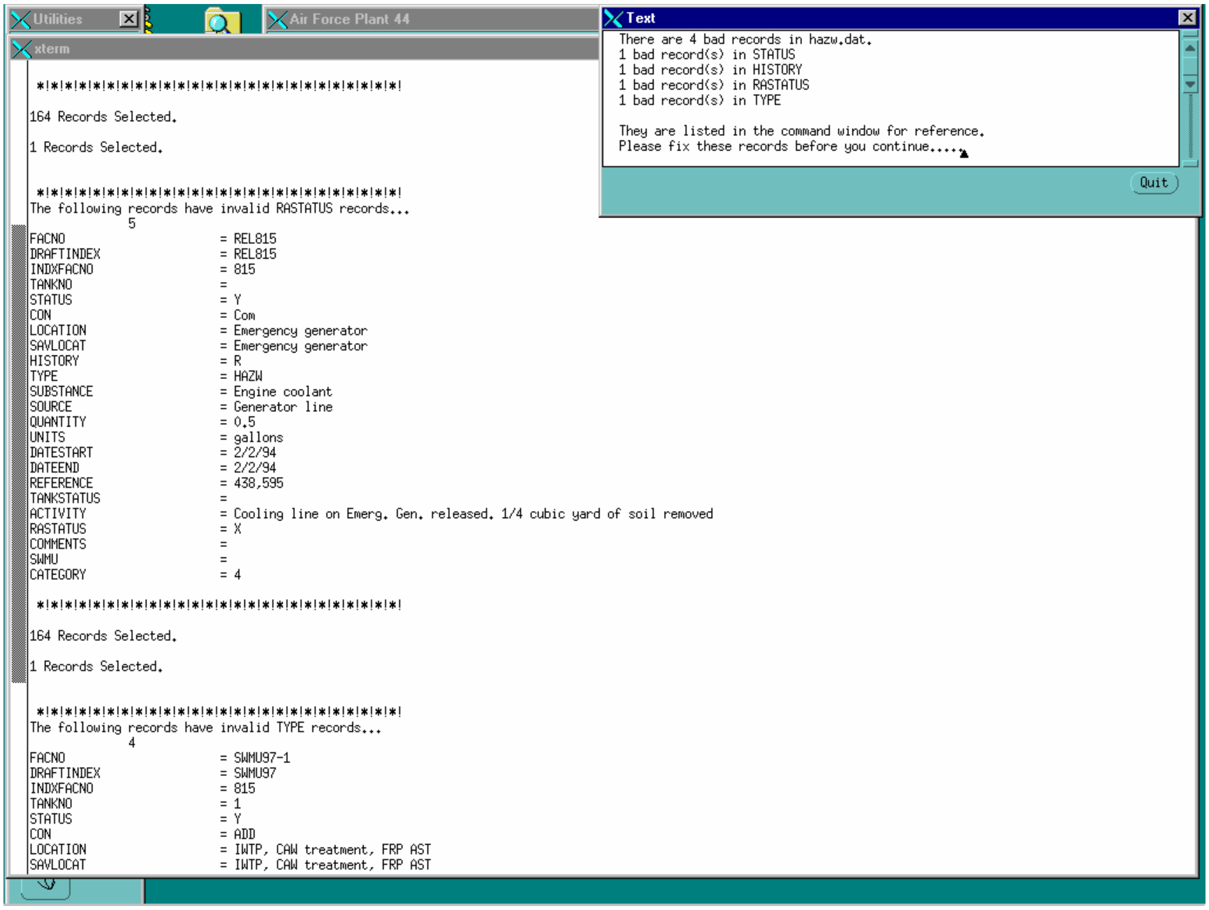
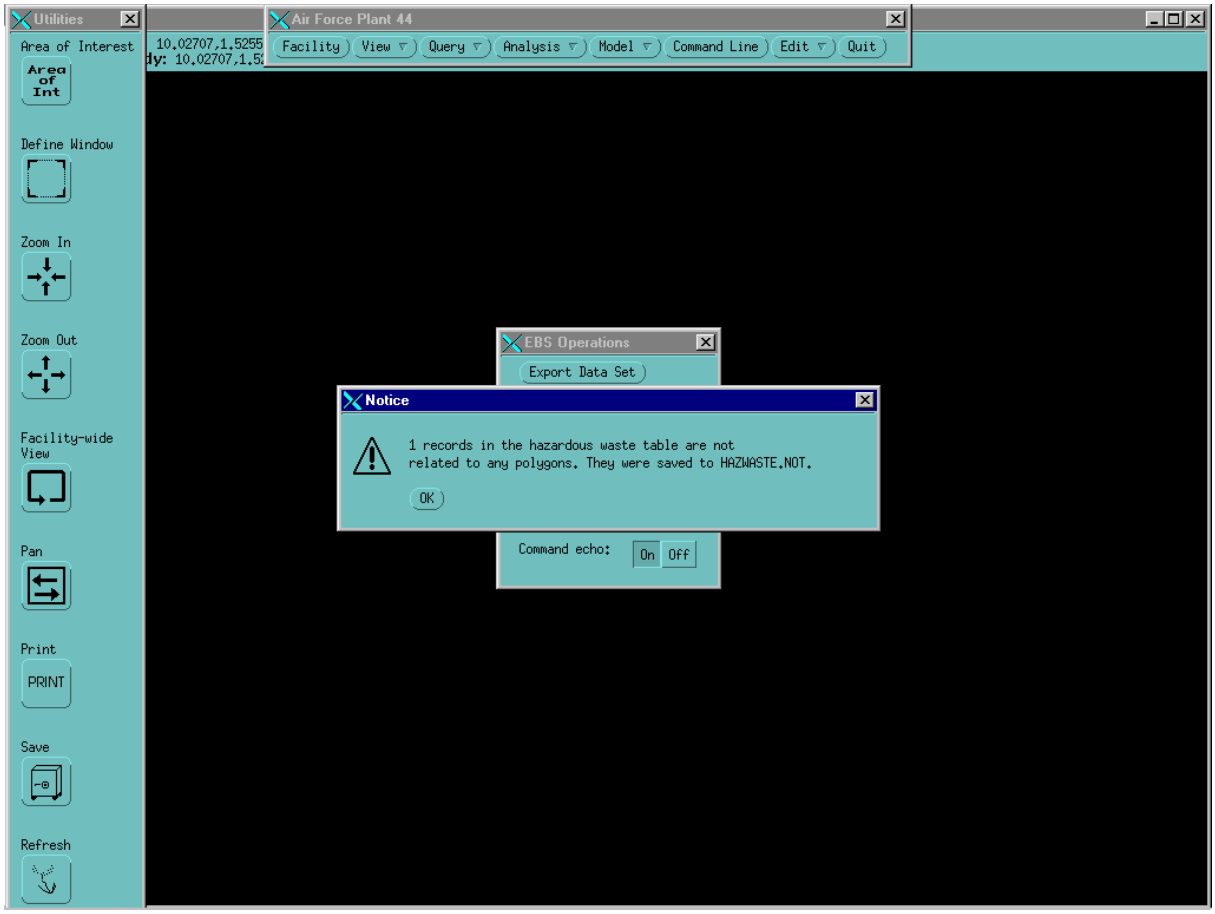


Figure 21. Screen 1 of 1 showing the screen as it appears when facnos in the hazw.dat file do not have corresponding facnos in one of the spatial data files



APPENDIX I
WILBUR'S SYSTEM ADMINISTRATOR GUIDE
NAMING CONVENTIONS FOR ALL COVERAGE/ATTRIBUTE FILES USED IN THE EBS PROCESS

Table I-1 Air Force Plant 44	
Attribute/Coverage Files AFP44 (Tucson, Arizona)	**Files Used in Property Categorization**
ATTRIBUTE NAMES	
ASBEST	
BLDG	
FACNO	
HAZW	●
PCB	
COVERAGE NAMES	
ADD	●
ADD2	●
ASTS	●
BOUND	●
CASCAW	●
DRYWELL	●
GIW	●
LEACH	●
PLUME1	●
PLUME2	●
PLUME3	●
SITES1	●
SITES2	●
SITES4	●
SITES6	●
STORM	●
STRUC	●
SWMU1	●
SWMU2	●
TUNNEL	●
USTS	●
Attribute/Coverage Files AFP44 (Tucson, Arizona)	**Files Used in Property Categorization**
OTHER NAMES USED FOR EBS PROCESS	
COVNAME.TAB	
CONDPROP	

APPENDIX J
WILBUR'S SYSTEM ADMINISTRATOR GUIDE
ARC/INFO EBS DATA PROCESSING SYSTEM EBS_OPS.AML SOURCE CODE

′′\′′\′′′\′\′\′\′‶″’’’’‘’,‘’‘’arcarc‘’‘’‘’‘’‘’‘’‘’‘’‘’‘’‘’‘’‘’‘’“‘’‘
’‘’ \ / \ / \ / \ / \ / ‘’‘’‘‘‘’‘’‘‘‘’“”“”“”““
”“”“”“”’ arc′arc”””””””””””

APPENDIX K
WILBUR'S SYSTEM ADMINISTRATOR GUIDE
ARC/INFO EBS DATA PROCESSING SYSTEM EBS_OPS.MENU SOURCE CODE

4.35.1 7 +ebs_ops.menu

```
/* ************************************************************************
/*                The Earth Technology Corporation
/*                 200 Sparkman Drive NW, Suite 1
/*                      Huntsville, AL
/*                       205-837-0199
/* ----------------------------------------------------------------
/* Menu    : +ebs_ops.menu
/* Purpose  :  To provide user interface to +EBS_OPS.AML.
/* Requires : ARC/INFO Rev. 7.0.3.
/*  ----------------------------------------------------------------
/* History  : written by Tim Rourke, TETC, August 9, 1995
/*        updated by John Cooley, Earth Tech, Sept 22, 1999
/*
/*           Modified: 09/8/1999 by John Cooley, Earth Tech - Alexandria
/*             1) The draw option on the EBS menu was commented out per
/*                request from Sandra Eberts, GIS Data Base Administrator
/*                for ASC, WPAFB.
/*
/*
/* ************************************************************************
/*
/* User: %input1
/* %button6_1
/*        %button7
/*    This input field was used during the testing process.
/*    It can be used to set a user name for storage in exported coverages.
/*    The name, along with the date and time, can be stored in export
/*    files to track exported versions of the same plant. The routines that
/*    use these items are ADD_LOG_ITEMS, FILL_LOG ITEMS and DROP_LOG_ITEMS.
/*    These routines are available by uncommenting the lines in the
/*    EXPORT routine that call ADD_LOG_ITEMS.
/* %input1 INPUT .+EBS_OPS$USER 24 TYPEIN YES SCROLL NO ~
/*   REQUIRED ~
/*   HELP 'Name of the user checking out/in' ~
/*   CHARACTER
/*    These buttons were used during the testing process. The IMPORT
/*    routine now calls these routines automatically.
/* %button6_1 BUTTON KEEP 'Popup Category Legend' &RUN +EBS_OPS LEGEND

/* %button7 BUTTON KEEP 'Help' &RUN +EBS_OPS HELP

    %button1
    %button1_1
    %button1_2
    %button1_4
/* %button1_3

    %button8_1

    %button6
```

4.35.1.1 Command echo: %choice1

```
%button1 BUTTON KEEP 'Export Data Set' &RUN +EBS_OPS EXPORT
%button1_1 BUTTON KEEP 'Import Data Set' &RUN +EBS_OPS IMPORT
```

%button1_2 BUTTON KEEP ‘Recategorization’ &RUN +EBS_OPS CATEGORY
%button1_4 BUTTON KEEP ‘Condition of Property’ &RUN +EBS_OPS CONDITION
/* %button1_3 BUTTON KEEP ‘Display EBS Datasets’ &RUN +EBS_DRAW INIT
%button8_1 BUTTON KEEP ‘Archive Data’ &RUN +EBS_OPS REPLACE
%button6 BUTTON KEEP ‘Cancel’ &RUN +EBS_OPS EXIT
%choice1 CHOICE .IF_ECHO PAIRS KEEP ~

4.35.1.1.1 INITIAL '.FALSE.' ~

HELP 'Set &ECHO ON or OFF for testing' ~
RETURN '&ECHO [UNQUOTE %.if_echo%]' ~
'On' '&ON' 'Off' '&OFF'