# ENGINEERING DATA TRANSMITTAL

**To:** (Receiving Organization)  
Data Automation Engineering and Services  

**From:** (Originating Organization)  
Facility Configuration Management and Integration  

**5. Proj./Prog./Dept./Div.:**  
Facility Configuration Management and Integration  

**6. Cog. Engr.:**  
J. B. Jech  

**8. Originator Remarks:**  
Approval and release of MEL Phase I - System Design Description (SDD)  

**11. Receiver Remarks:**  
Design Baseline Document? [ ] Yes [X] No  

## DATA TRANSMITTED

<table>
<thead>
<tr>
<th>Item No.</th>
<th>Document/Drawing No.</th>
<th>Sheet No.</th>
<th>Rev. No.</th>
<th>Title or Description of Data Transmitted</th>
<th>Approval Designator</th>
<th>Reason for Transmittal</th>
<th>Originator Disposition</th>
<th>Receiver Disposition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HNF-SD-WM-SDD-065</td>
<td>0</td>
<td></td>
<td>MEL Phase I SDD</td>
<td>Q</td>
<td>1, 2</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

## KEY

<table>
<thead>
<tr>
<th>Reason</th>
<th>Disp.</th>
<th>(J) Name</th>
<th>(K) Signature</th>
<th>Date</th>
<th>MSIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cog. Engr. J. B. Jech</td>
<td>[Signature]</td>
<td>[Signature]</td>
<td>S4-43</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Cog. Mgr. R. L. Nelson</td>
<td>[Signature]</td>
<td>[Signature]</td>
<td>S4-43</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>QA M. L. Hermansen</td>
<td>[Signature]</td>
<td>[Signature]</td>
<td>G1-50</td>
<td></td>
</tr>
</tbody>
</table>

**Required Response Date:** 3/18/97
<table>
<thead>
<tr>
<th>BLOCK</th>
<th>TITLE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)*</td>
<td>EDT</td>
<td>Pre-assigned EDT number.</td>
</tr>
<tr>
<td>(2)</td>
<td>To: (Receiving Organization)</td>
<td>Enter the individual's name, title of the organization, or entity (e.g., Distribution) that the EDT is being transmitted to.</td>
</tr>
<tr>
<td>(3)</td>
<td>From: (Originating Organization)</td>
<td>Enter the title of the organization originating and transmitting the EDT.</td>
</tr>
<tr>
<td>(4)</td>
<td>Related EDT No.</td>
<td>Enter EDT numbers which relate to the data being transmitted.</td>
</tr>
<tr>
<td>(5)*</td>
<td>Proj./Prog./Dept./Div.</td>
<td>Enter the Project/Program/Department/Division title or Project/Program acronym or Project Number, Work Order Number or Organization Code.</td>
</tr>
<tr>
<td>(6)*</td>
<td>Cognizant Engineer</td>
<td>Enter the name of the individual identified as being responsible for coordinating disposition of the EDT.</td>
</tr>
<tr>
<td>(7)</td>
<td>Purchase Order No.</td>
<td>Enter related Purchase Order (P.O.) Number, if available.</td>
</tr>
<tr>
<td>(8)*</td>
<td>Originator Remarks</td>
<td>Enter special or additional comments concerning transmittal, or &quot;Key&quot; retrieval words may be entered.</td>
</tr>
<tr>
<td>(9)</td>
<td>Equipment/Component No.</td>
<td>Enter equipment/component number of affected item, if appropriate.</td>
</tr>
<tr>
<td>(10)</td>
<td>System/Bldg./Facility</td>
<td>Enter applicable system, building or facility number, if appropriate.</td>
</tr>
<tr>
<td>(11)</td>
<td>Receiver Remarks</td>
<td>Enter special or additional comments concerning transmittal.</td>
</tr>
<tr>
<td>(12)</td>
<td>Major Assm. Dwg. No.</td>
<td>Enter applicable drawing number of major assembly, if appropriate.</td>
</tr>
<tr>
<td>(13)</td>
<td>Permit/Permit Application No.</td>
<td>Enter applicable permit or permit application number, if appropriate.</td>
</tr>
<tr>
<td>(14)</td>
<td>Required Response Date</td>
<td>Enter the date a response is required from individuals identified in Block 17 (Signature/Distribution).</td>
</tr>
<tr>
<td>(15)*</td>
<td>Data Transmitted</td>
<td>Enter sequential number, beginning with 1, of the information listed on EDT.</td>
</tr>
<tr>
<td>(A)*</td>
<td>Item Number</td>
<td>Enter the unique identification number assigned to the document or drawing being transmitted.</td>
</tr>
<tr>
<td>(B)*</td>
<td>Document/Drawing No.</td>
<td>Enter the sheet number of the information being transmitted. If no sheet number, leave blank.</td>
</tr>
<tr>
<td>(C)*</td>
<td>Sheet No.</td>
<td>Enter the revision number of the information being transmitted. If no revision number, leave blank.</td>
</tr>
<tr>
<td>(D)*</td>
<td>Rev. No.</td>
<td>Enter the title or description of Data Transmitted.</td>
</tr>
<tr>
<td>(E)*</td>
<td>Approval Designator</td>
<td>Enter the appropriate Approval Designator (Block 15). Also, indicate the appropriate approvals for each item listed, i.e., SQ, ESQ, etc.</td>
</tr>
<tr>
<td>(F)*</td>
<td>Reason for Transmittal</td>
<td>Enter the appropriate code to identify the purpose of the data transmittal (see Block 16).</td>
</tr>
<tr>
<td>(G)</td>
<td>Originator Disposition</td>
<td>Enter the appropriate disposition code (see Block 16).</td>
</tr>
<tr>
<td>(H)</td>
<td>Receiver Disposition</td>
<td>Enter the appropriate disposition code (see Block 16).</td>
</tr>
<tr>
<td>(16)</td>
<td>Key</td>
<td>Number codes used in completion of Blocks 15 (G), (H), and (I), and 17 (G), (H) (Signature/Distribution).</td>
</tr>
<tr>
<td>(17)</td>
<td>Signature/Distribution</td>
<td>Enter the code of the reason for transmittal (Block 16).</td>
</tr>
<tr>
<td>(I)</td>
<td>Disposition</td>
<td>Enter the code for the disposition (Block 16).</td>
</tr>
<tr>
<td>(J)</td>
<td>Name</td>
<td>Enter the signature of the individual completing the Disposition 17 (H) and the Transmittal.</td>
</tr>
<tr>
<td>(K)*</td>
<td>Signature</td>
<td>Obtain appropriate signature(s).</td>
</tr>
<tr>
<td>(L)*</td>
<td>Date</td>
<td>Enter date signature is obtained.</td>
</tr>
<tr>
<td>(M)*</td>
<td>MSIN</td>
<td>Enter MSIN. Note: If Distribution Sheet is used, show entire distribution (including that indicated on Page 1 of the EDT) on the Distribution Sheet.</td>
</tr>
<tr>
<td>(18)</td>
<td>Signature of EDT Originator</td>
<td>Enter the signature and date of the individual originating the EDT (entered prior to transmittal to Receiving Organization). If the EDT originator is the cognizant engineer, sign both Blocks 17 and 18.</td>
</tr>
<tr>
<td>(19)</td>
<td>Authorized Representative for Receiving Organization</td>
<td>Enter the signature and date of the individual identified by the Receiving Organization as authorized to approve disposition of the EDT and acceptance of the data transmitted, as applicable.</td>
</tr>
<tr>
<td>(20)*</td>
<td>Cognizant Manager</td>
<td>Enter the signature and date of the cognizant manager. (This signature is authorization for release.)</td>
</tr>
<tr>
<td>(21)*</td>
<td>DOE Approval</td>
<td>Enter DOE approval (if required) by signature or control number that tracks the approval to a signature, and indicate DOE action.</td>
</tr>
</tbody>
</table>

* Asterisk denotes the required minimum items check by Configuration Documentation prior to release; these are the minimum release requirements.
System Design Description For Master Equipment List, Phase I

J. D. Sandoval
Fluor Daniel Northwest
P. O. Box 1050
Richland, WA 99352
U.S. Department of Energy Contract DE-AC06-87RL10930

EDT/ECN: 615552 UC: 605
Org Code: 77820 Charge Code: N12A4
B&R Code: EW3120071 Total Pages: 106

Key Words: System Design Description

Abstract: Approval and release for the Master Equipment List (MEL) Phase I, for the System Design Description (SDD)

TRADEMARK DISCLAIMER. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors.

Printed in the United States of America. To obtain copies of this document, contact: WHC/BCS Document Control Services, P.O. Box 1970, Mailstop H6-08, Richland WA 99352. Phone (509) 372-2420; Fax (509) 376-4989.

Approved for Public Release

A-6400-073 (10/95) GEF321
SYSTEM DESIGN DESCRIPTION

FOR

MASTER EQUIPMENT LIST PHASE I

Revision 0

Prepared by:

Data Automation Engineering & Services
Fluor-Daniel Northwest, Inc.

November 1996

Prepared for:

Lockheed Martin Hanford Corporation
PO Box 1500
Richland, Washington 99352-1505
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0 INTRODUCTION</td>
<td>4</td>
</tr>
<tr>
<td>1.1 PURPOSE</td>
<td>4</td>
</tr>
<tr>
<td>1.2 SCOPE</td>
<td>4</td>
</tr>
<tr>
<td>1.3 OVERVIEW</td>
<td>5</td>
</tr>
<tr>
<td>1.4 DEFINITIONS AND ABBREVIATIONS</td>
<td>5</td>
</tr>
<tr>
<td>2.0 DATABASE</td>
<td>7</td>
</tr>
<tr>
<td>2.1 DATA MODEL</td>
<td>7</td>
</tr>
<tr>
<td>2.2 ENTITY RELATIONSHIP DIAGRAM</td>
<td>9</td>
</tr>
<tr>
<td>2.3 PHYSICAL MODEL</td>
<td>12</td>
</tr>
<tr>
<td>2.4 DATA DICTIONARY</td>
<td>14</td>
</tr>
<tr>
<td>2.4.1 USER INTERFACE TABLES</td>
<td>16</td>
</tr>
<tr>
<td>2.4.2 SYSTEM TABLES</td>
<td>47</td>
</tr>
<tr>
<td>3.0 PROGRAMMING</td>
<td>53</td>
</tr>
<tr>
<td>3.1 PLATFORM</td>
<td>53</td>
</tr>
<tr>
<td>3.2 USER INTERFACE</td>
<td>54</td>
</tr>
<tr>
<td>3.2.1 PROGRAM UNITS</td>
<td>54</td>
</tr>
<tr>
<td>3.2.1.1 EDITVIEW UNIT</td>
<td>55</td>
</tr>
<tr>
<td>3.2.1.2 EDITVIEW STRUCTURE CHARTS</td>
<td>70</td>
</tr>
<tr>
<td>3.2.2 FOLDERS</td>
<td>79</td>
</tr>
<tr>
<td>3.2.3 SCREENS</td>
<td>80</td>
</tr>
<tr>
<td>3.2.3.1 DATA VIEW - POINTER FOLDER</td>
<td>80</td>
</tr>
<tr>
<td>3.2.3.2 DATA VIEW - REFERENCES FOLDER</td>
<td>81</td>
</tr>
<tr>
<td>3.2.3.3 DATA VIEW - NAMEPLATE FOLDER</td>
<td>82</td>
</tr>
<tr>
<td>3.2.3.4 DATA VIEW - SPECIFICATIONS FOLDER</td>
<td>83</td>
</tr>
<tr>
<td>3.2.3.5 DATA VIEW - SAFETY FOLDER</td>
<td>84</td>
</tr>
<tr>
<td>3.2.3.6 QUERY</td>
<td>85</td>
</tr>
<tr>
<td>3.2.3.7 REPORTS</td>
<td>86</td>
</tr>
<tr>
<td>3.2.3.8 LOCKS AND CHANGES PENDING</td>
<td>87</td>
</tr>
<tr>
<td>3.2.3.9 CREATE NEW EINS</td>
<td>88</td>
</tr>
<tr>
<td>3.2.3.10 STANDARD DATA DEFINITIONS</td>
<td>89</td>
</tr>
<tr>
<td>3.2.3.11 CHANGE LOG</td>
<td>90</td>
</tr>
<tr>
<td>3.2.3.12 ACCESS CONTROL</td>
<td>91</td>
</tr>
<tr>
<td>3.2.3.13 ABOUT</td>
<td>92</td>
</tr>
</tbody>
</table>
SYSTEM DESIGN DESCRIPTION FOR
MASTER EQUIPMENT LIST PHASE I

1.0 INTRODUCTION

1.1 PURPOSE

This System Design Description (SDD) is for the Master Equipment List Phase I (MEL). It has been prepared following the WHC-CM-3-10, "Software Practices," (Ref. 6). This SDD describes the internal design for implementation of the MEL Phase I.

1.2 SCOPE

The MEL Phase I application includes features to;

- Enable users to view a variety of information about equipment in various data tables using a 'folder' metaphor.
- Provide data search features to quickly locate information about the equipment.
- Provide query functions for locating key data about the equipment.
- Provide the ability to generate standard reports of folder information.
- Maintain MEL Phase I user IDs and passwords.
- Provide the ability to edit existing data, enter new data, and delete data based upon predefined privileges to do so.
- Provide audit trail information for monitoring and logging data entry and change.
1.3 OVERVIEW

The remainder of this SDD is divided into four main sections.

Section 2, "Database" contains data models, entity relationship diagrams, physical models, and a data dictionary.

Section 3, "Programming" contains platform information, screen views of the user interface, startup and execution parameter definitions, and detailed information regarding installation.

The final section, Section 4, "References", contains the references for this document.

1.4 DEFINITIONS AND ABBREVIATIONS

The following are definitions of abbreviations and terms used in this document:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>DAES</td>
<td>Data Automation Engineering &amp; Services</td>
</tr>
<tr>
<td>DCA</td>
<td>Data Collection Application</td>
</tr>
<tr>
<td>ERD</td>
<td>Entity Relationship Diagram</td>
</tr>
<tr>
<td>EP</td>
<td>Engineering Practices</td>
</tr>
<tr>
<td>MEL Phase I</td>
<td>Master Equipment List Phase I</td>
</tr>
<tr>
<td>HLAN</td>
<td>Hanford Local Area Network</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>SDD</td>
<td>System Design Description</td>
</tr>
<tr>
<td>SRS</td>
<td>System Requirements Specification</td>
</tr>
<tr>
<td>STD</td>
<td>State Transition Diagram</td>
</tr>
<tr>
<td>UA</td>
<td>User Application</td>
</tr>
</tbody>
</table>

ASCII--A standard used to represent text and control characters in a computer environment. The ASCII format relates hexadecimal numbers to these text and control characters.

Engineering Practices (EPs)--supporting document management manual WHC-CM-6-1 that establishes the manner and order for performance of common configuration management practices relating to engineering tasks.
Entity—subpiece, recognizable, with a distinct function within the MEL Phase I project.

ERD—diagram showing system/environment entities and relationships.

MEL Phase I—The project being performed by DAES to provide a means of locating and querying information about equipment in Tank Farms. See WHC-SD-WM-SDP-O12 (Ref. 4) for the work plan for this system.

Approval Designator—defined in WHC-CM-3-5, "Document Control and Records Management Manual," Section 12.7 (Ref. 7).

Module—subsystem of the MEL Phase I system that encompasses attributes of the system including hardware and software.

STD—diagram showing control sequence for the system/environment.

System Design Description—documentation of essential and implementation design that gives descriptions of entities and attributes of the design.

System Requirements Specification—the documentation of essential requirements (functions, performance, design constraints and attributes) of the computer software and its external interfaces.
2.0 DATABASE

2.1 DATA MODEL

The following data model show the relationships of the data for the MEL.
- Rectangles represent entities.
- Rounded rectangles represent relationships.
- Connectors with 'x,y' numbers represent cardinality.
- For example, the relationships between 'Pointer Data' and 'Master EIN' are read as follows:
  - "Pointer Data is identified by 1 to 1 (only 1) Master EIN".
  - "A Master EIN uniquely identifies 0 to m (many) Pointer Data".

Master Equipment List, Phase I
Semantic Data Model
2.2 ENTITY RELATIONSHIP DIAGRAM

The following entity relationship diagrams show the relationships of the MEL.
Master Equipment List, Phase I
Entity-Relationships for User Interface Tables
Master Equipment List, Phase I
Entity-Relationships for System Tables
2.3 PHYSICAL MODEL

The following physical model diagram shows the data and lookup tables for the MEL Phase I and field links.
Master Equipment List, Phase I, version 2 Database Tables
2.4 DATA DICTIONARY

This Data Dictionary (DD) provides a detailed description of the data structures and their elements which will compose the MEL Phase I application. The DD does not define explicit relationships between structure entities - those are better illustrated in Section 2.2 “Entity Relationship Diagram” of this document.

This Data Dictionary is organized into two major parts;

- Structures that support the user interface and the data gathered in its use,
- Structures that support the system as a whole - including tables used during reports generation, change log tables, session tables, and general system support tables.

These parts each contain a list of table structures. Because the number of these structures is relatively small, they are presented in the order that follows their application to the MEL Phase I application. An alphabetical ordering does not seem useful for this application.

Each structure is divided into its component data elements or fields. Fields are listed in the order that presents key fields first (as required in a Paradox implementation). Each field will be described in terms of:

- **Field Name** The field’s reference name.
- **Type & Size** The data type and size (see description below).
- **Key** An asterisk (*) next to the type & size indicates a key field.
- **Required** Data that is required (per customer requirement or design) is indicated by ‘True’.
- **Example** Example of a typical data entry.
- **Restriction** Any special restrictions or conditions regarding the data.
- **Description** Textual description of the data.

Type and size formats are typically shown with the shorthand of a single character type which is sometimes followed by a number indicating the number of bytes reserved to hold the data value. The types are defined as:

- **A** = Alphanumeric (with limited reserved space)
- **N** = Numeric (floating point)
- **I** = Long integer (4 bytes)
- **S** = Small integer (2 bytes)
- **D** = Date (Paradox internal format)
M = Memo (with limited reserved base table space and virtually unlimited memo table space)
L = Logical (boolean value, True or False)
+ = Autoincrement (small integer value, incremented automatically with each new record)

Alphanumeric (A) formats always include a number - e.g. A25 represents a 25 byte alphanumeric data field. In this example, 25 bytes are reserved in the table for each record.

For memos (M), the number represents the bytes reserved in the base table (the named table structure) although the full memo length is virtually unlimited with the use of an auxiliary memo table. The number is important since a form field will display the memo data in the base table without accessing the memo table (this saves time). The memo table is accessed only when the complete memo is required. Typically, form fields should be sized to match the base size of the memo - e.g., M120 represents a memo field of 120 bytes immediately accessible in the base table even though the actual memo length is as long as necessary to contain the complete memo data.

Restriction formats are usually made up of alphanumeric characters that represent certain values. The types are defined as:

a = any alphanumeric character
n = Number (0-9)
[] = Any character(s) surrounded by brackets are optional.
Lookup = The value must be selected from a pre-defined list.

A restriction format may be made up of a combination of alphanumeric characters, numbers, and literal values. For example, a restriction format for a H-14 drawing would be represented by the following:

H-14-nnnnnn

Therefore, H-14-012345, would be considered a valid value, while H-14-ABC123 would not.
2.4.1 USER INTERFACE TABLES

EIN_INFO.DB  This table identifies all the EINs entered.

EIN_Code
Size & Type: A12*
Required: False
Folder: 
Example: EIN000000033
Restrictions:
Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

EIN
Size & Type: A23
Required: True
Folder: Equipment Identification Number block.
Example: AN000-AS-AOD-1001
Restrictions: aa[aaaa]-aa[aaa]-aa[aaa]-nnn[a]
Information: This field is a concatenation of the Farm Facility, System, Component, and Number fields. It is generated automatically by the system.

System
Size & Type: A5
Required: True
Folder: Equipment Identification Number block.
Example: CHMA
Restrictions: lookup
Information: This field forces a value from the Systems.DB, System field.

Component
Size & Type: A5
Required: True
Folder: Equipment Identification Number block.
Example: AOD
Restrictions: Look up
Information: This field forces a value from the Comp_Des.DB, Component field.

Farm_Facility
Size & Type: A6
Required: True
Folder: Location block.
Example: AN000
Restrictions: Lookup
Information: This field forces a value from the Fac_Farm.DB, Farm_Facility field.
Number
Size & Type: A4
Required: True
Folder: Equipment Identification Number block.
Example: 1001
Restrictions: none[na]
Information: This field is the unique part of the EIN.

Noun_Name1
Size & Type: A32
Required: True
Folder: Equipment Identification Number block.
Example: XMITTER
Restrictions: Lookup
Information: This field provides a lookup to noun names from similar EINs. However, a typed value may be entered.

Noun_Name2
Size & Type: A32
Required: True
Folder: Equipment Identification Number block.
Example: XMITTER
Restrictions: Lookup
Information: This field provides a lookup to noun names from similar EINs. However, a typed value may be entered.

Status
Size & Type: A3
Required:
Folder: Equipment Identification Number block.
Example: 1
Restrictions:
Information: This field provides a lookup to number representing a status.

Comp Func
Size & Type: A20
Required: True
Folder: Equipment Identification Number block.
Example: HORIZONTAL
Restrictions: lookup
Information: This field is used as a link to the Comp_Typ.DB. Comp Func field. This is a lookup value that forces a selection from the list.
LockID
  Size & Type: I
  Required:
  Folder:
  Example: 1
  Restrictions:
  Information: This is the identifier for the lock session. This data is used internally by the system.

LockRec
  Size & Type: I
  Required:
  Folder:
  Example: 1
  Restrictions:
  Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
  Size & Type: +
  Required:
  Folder:
  Example: 1
  Restrictions:
  Information: This is a record identifier used internally by the system.

**EQP_INFO.DB** This table contains the equipment information for each EIN entered.

EIN_Code
  Size & Type: A12*
  Required: True
  Folder:
  Example: EIN000000033
  Restrictions:
  Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

Loop_Seq_No
  Size & Type: A7
  Required:
  Folder: Pointer
  Example: AN212-1
  Restrictions: aamn-n
  Information: This field is used as the key field in CBRS (Component Based Recall System). This is the reference source for historical data.
Barcode_No
Size & Type: A9
Required:
Folder: Pointer
Example: S000120WT
Restrictions:
Information: Barcode Number. Unique identifier assigned to the equipment that resides/functions in the EIN Location, used for developing equipment history information (cradle to grave tracking) versus functional location history information.

Property_No
Size & Type: A10
Required:
Folder: Pointer
Example: WC42029
Restrictions:
Information: Property Number. Unique identifier (Barcode) that property assigns to equipment for tracking purposes in the Property Management System.

Vendor_Info_No
Size & Type: A11
Required:
Folder: Pointer
Example: 0022574-004
Restrictions: nnnnnnn-nnn
Information: Vendor Information Number. Used as the key field in the Certified Vendor Information System (Soft Reporting).

Mfg_Vndr_Name
Size & Type: A25
Required:
Folder: Pointer
Example: Ingersoll-Rand
Restrictions:
Information: Manufacturer or Vendor Name.

Mfg_Vndr_No
Size & Type: A25
Required:
Folder: Pointer
Example: 37F427X484
Restrictions:
Information: Manufacturer or Vendor Number. Unique identifier Manufacturer/Vendor assigns to equipment.
Model No
Size & Type: A25
Required:
Folder: Pointer
Example: 37F427X484
Restrictions:
Information: Model Number. Unique identifier manufacturer assigns to equipment.

Serial No
Size & Type: A25
Required:
Folder: Pointer
Example: 30T-714984
Restrictions:
Information: Serial Number. Unique identifier that Manufacturer assigns to equipment.

LockID
Size & Type: I
Required:
Folder:
Example: 1
Restrictions:
Information: This is the identifier for the lock session. This data is used internally by the system.

LockRec
Size & Type: I
Required:
Folder:
Example: 1
Restrictions:
Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
Size & Type: +
Required:
Folder:
Example: 1
Restrictions:
Information: This is a record identifier used internally by the system.

**SPECS.DB** This table contains the specification information for each EIN as applicable.

EIN Code
Size & Type: A12*
Required: True
Folder:
Example: EIN000000033
Restrictions:
Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

**Design_Press**
- Size & Type: A15
- Required:
- Folder: Specifications
- Example: PSI
- Restrictions:
- Information: Design Pressure Range. Typical units of measure in the industry.

**Design_Temp**
- Size & Type: A15
- Required:
- Folder: Specifications
- Example: 80-100 F
- Restrictions:
- Information: Design Temperature Range. Fahrenheit will be the unit of measure.

**Design_Input**
- Size & Type: A15
- Required:
- Folder: Specifications
- Example: gpm
- Restrictions:
- Information: Design Input Range.

**Design_Output**
- Size & Type: A15
- Required:
- Folder: Specifications
- Example: pph
- Restrictions:
- Information: Design Output Range.

**Design_Tolerance**
- Size & Type: A15
- Required:
- Folder: Specifications
- Example: 23%
- Restrictions:
- Information: Design Accuracy. Units could be % or (+/-).
Design_Set_Pt
Size & Type: A15
Required:
Folder: Specifications
Example: 12
Restrictions:
Information: Design Set point. Single numeric value, or a range (all units).

Operating_Press
Size & Type: A15
Required:
Folder: Specifications
Example: PSI
Restrictions:
Information: Operating Pressure Range.

Operating_Temp
Size & Type: A15
Required:
Folder: Specifications
Example: 80-100 F
Restrictions:
Information: Design Temperature Range. Fahrenheit will be the unit of measure.

Operating_Input
Size & Type: A15
Required:
Folder: Specifications
Example: pph
Restrictions:
Information: Operating Input Range.

Operating_Output
Size & Type: A15
Required:
Folder: Specifications
Example: cfm
Restrictions:
Information: Operating Output Range.

Operating_Set_Pt
Size & Type: A15
Required:
Folder: Specifications
Example: 12
Restrictions:
Information: Operating Set point. Single values or a range (all units).

LockID
Size & Type: I
Required:
Folder:
Example: 1
Restrictions:
  Information: This is the identifier for the lock session. This data is used internally by the system.

LockRec
Size & Type: I
Required:
Folder:
Example: 1
Restrictions:
  Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
Size & Type: +
Required:
Folder:
Example: 1
Restrictions:
  Information: This is a record identifier used internally by the system.

SAFETY.DB This table contains the safety information for each EIN as applicable.

EIN_Code
Size & Type: A12*
Required:
Folder:
Example: EIN000000033
Restrictions:
  Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

OSR
Size & Type: A1
Required: True
Folder: Classification block.
Example: Y
Restrictions: Y or N
  Information: This field is a boolean value represented by either a Y (Yes) or N (No).

Environmental
Size & Type: A1
Required: True
Folder: Classification block.
Example: Y
Restrictions: Y or N
  Information: This field is a boolean value represented by either a Y (Yes) or N (No).
QA
Size & Type: A1
Required:
Folder: Classification block.
Example: Y
Restrictions: Y or N
Information: This field is a boolean value represented by either a Y (Yes) or N (No).

Critical
Size & Type: A1
Required: False
Folder: Safety
Example: Y
Restrictions: Y,N,1,2,3,4
Information: This is the critical classification.

Safety_Class
Size & Type: A1
Required: True
Folder: Safety
Example: Y
Restrictions: Y,N,1,2,3,4
Information: This is the safety classification.

Safety_Setpoint
Size & Type: A15
Required:
Folder: Safety
Example:
Restrictions:
Information: Safety set point determination.

Safety_Limit
Size & Type: A20
Required:
Folder: Safety
Example:
Restrictions:
Information: Safety limit determination.

Safe_System
Size & Type: A25
Required:
Folder: Safety
Example:
Restrictions:
Information: Safety system.
Surv_Freq
- Size & Type: A10
- Required:
- Folder: Safety
- Example: Weekly
- Restrictions:
- Information: Surveillance Frequency. Frequency at which equipment is inspected.

Cal_Freq
- Size & Type: A10
- Required:
- Folder: Safety
- Example: Weekly
- Restrictions:
- Information: Calibration Frequency. Frequency at which the equipment is calibrated.

Loop_Prior
- Size & Type: A10
- Required:
- Folder: Safety
- Example:
- Restrictions:
- Information: Loop priority.

LockID
- Size & Type: 1
- Required:
- Folder:
- Example: 1
- Restrictions:
- Information: This is the identifier for the lock session. This data is used internally by the system.

LockRec
- Size & Type: 1
- Required:
- Folder:
- Example: 1
- Restrictions:
- Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
- Size & Type: +
- Required:
- Folder:
- Example: 1
- Restrictions:
- Information: This is a record identifier used internally by the system.
MOTOR_DB This table contains the motor information for each EIN as applicable.

EIN_Code
Size & Type: A12*
Required: True
Folder:
Example: EIN000000033
Restrictions:
Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

Input_Volt
Size & Type: A10
Required:
Folder: Nameplate
Example: 120AC
Restrictions: AC or DC
Information: Input voltage. Volts will be the unit of measure.

Phases
Size & Type: A5
Required:
Folder: Nameplate
Example: 1
Restrictions: 1,2,3
Information: Phases.

Motor_Size
Size & Type: A10
Required:
Folder: Nameplate
Example: 25 HP
Restrictions: mmmmm
Information: Motor size. Horsepower will be the unit of measure.

Motor_Speed
Size & Type: A10
Required:
Folder: Nameplate
Example: 500 RPM
Restrictions: mmmmm
Information: Motor speed. RPM will be the unit of measure.
Frame Size
  Size & Type: A15
  Required:
  Folder: Nameplate
  Example: 284T
  Restrictions:
  Information: Frame size.

Duty Start Cycle
  Size & Type: A20
  Required:
  Folder: Nameplate
  Example:
  Restrictions:
  Information: Duty Start Cycle.

Full Load Amps
  Size & Type: A10
  Required:
  Folder: Nameplate
  Example: 12
  Restrictions: nn
  Information: Full Load Amps. AMPS will be the unit of measure.

Locked Rotor Code
  Size & Type: A10
  Required:
  Folder: Nameplate
  Example:
  Restrictions:
  Information: Locked Rotor Code.

Service Factor
  Size & Type: A20
  Required:
  Folder: Nameplate
  Example: .5
  Restrictions: 0<n<1
  Information: Design Service Factor. No unit of measure.

LockID
  Size & Type: I
  Required:
  Folder:
  Example: 1
  Restrictions:
  Information: This is the identifier for the lock session. This data is used internally by the system.
LockRec
  Size & Type: I
  Required:
  Folder:
  Example: 1
  Restrictions:
  Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
  Size & Type: +
  Required:
  Folder:
  Example: 1
  Restrictions:
  Information: This is a record identifier used internally by the system.

**ELECTRIC.DB** This table contains the motor information for each EIN as applicable.

**EIN Code**
  Size & Type: A12*
  Required: True
  Folder:
  Example: EIN000000033
  Restrictions:
  Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

**Power_Supply_Bkr**
  Size & Type: A26
  Required:
  Folder: Nameplate
  Example:
  Restrictions:
  Information: Power Supply Breaker.

**Bkr_Trip_Setting**
  Size & Type: A20
  Required:
  Folder: Nameplate
  Example: 100 AMPS
  Restrictions:
  Information: Breaker Trip Setting. AMPS will be the unit of measure.
Fuse Size
  Size & Type: A10
  Required:
  Folder: Nameplate
  Example: 100 AMPS
  Restrictions:
  Information: Fuse Size. AMPs will be the unit of measure.

Mfg. Overload Size
  Size & Type: A20
  Required:
  Folder: Nameplate
  Example:
  Restrictions:
  Information: Manufacturer Overload Size.

LockID
  Size & Type: I
  Required:
  Folder:
  Example: 1
  Restrictions:
  Information: This is the identifier for the lock session. This data is used internally by the system.

LockRec
  Size & Type: I
  Required:
  Folder:
  Example: 1
  Restrictions:
  Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
  Size & Type: +
  Required:
  Folder:
  Example: 1
  Restrictions:
  Information: This is a record identifier used internally by the system.
MECHANIC.DB  This table contains the mechanical information for each EIN as applicable.

EIN Code
  Size & Type: A12*
  Required: True
  Folder: Example: EIN000000033
  Restrictions:
  Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

Bearings_Sealed_F
  Size & Type: A15
  Required:
  Folder: Nameplate
  Example: Y
  Restrictions: Y or N
  Information: Front Bearing Sealed. This field is a boolean value represented by either a Y (Yes) or N (No).

Bearings_Sealed_R
  Size & Type: A15
  Required:
  Folder: Nameplate
  Example: Y
  Restrictions: Y or N
  Information: Rear Bearing Sealed. This field is a boolean value represented by either a Y (Yes) or N (No).

Bearings_Vndr_No_F
  Size & Type: A20
  Required:
  Folder: Nameplate
  Example:
  Restrictions:
  Information: Front Bearing Number.

Bearings_Vndr_No_R
  Size & Type: A20
  Required:
  Folder: Nameplate
  Example:
  Restrictions:
  Information: Rear Bearing Number.
LockID
   Size & Type: I
   Required:
   Folder: 1
   Example:
   Restrictions:
   Information: This is the identifier for the lock session. This data is used internally by the system.

LockRec
   Size & Type: I
   Required:
   Folder:
   Example: 1
   Restrictions:
   Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
   Size & Type: +
   Required:
   Folder:
   Example: 1
   Restrictions:
   Information: This is a record identifier used internally by the system.

**PMS.DB**  This table contains 1 to many preventative maintenance records for each EIN as applicable.

EIN_Code
   Size & Type: A12*
   Required: True
   Folder:
   Example: EIN000000033
   Restrictions:
   Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

PMS_No
   Size & Type: A8*
   Required: True
   Folder: Pointer
   Example: 9X-00003
   Restrictions: aa-nnnnn
   Information: Unique number JCS assigns to a specific job or activity.
**LockID**
- **Size & Type:** I
- **Required:**
- **Folder:**
- **Example:**
- **Restrictions:**
  - **Information:** This is the identifier for the lock session. This data is used internally by the system.

**LockRec**
- **Size & Type:** I
- **Required:**
- **Folder:**
- **Example:**
- **Restrictions:**
  - **Information:** This is the identifier for the record being modified. This data is used internally by the system.

**RecordId**
- **Size & Type:** +
- **Required:**
- **Folder:**
- **Example:**
- **Restrictions:**
  - **Information:** This is a record identifier used internally by the system.

**PROCED.DB** This table contains 1 to many procedure records for each EIN as applicable.

**EIN Code**
- **Size & Type:** A12*
- **Required:** True
- **Folder:**
- **Example:** EIN000000033
- **Restrictions:**
  - **Information:** This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

**Procedure No**
- **Size & Type:** A12*
- **Required:** True
- **Folder:** References
- **Example:** 6-TF-058-2E
- **Restrictions:**
  - **Information:** Procedure Number. Unique identifier for procedures, operating procedures, maintenance procedures, test procedures, etc.
LockID
Size & Type: I
Required:
Folder:
Example:
Restrictions:
Information: This is the identifier for the lock session. This data is used internally by the system.

LockRec
Size & Type: I
Required:
Folder:
Example:
Restrictions:
Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
Size & Type: +
Required:
Folder:
Example:
Restrictions:
Information: This is a record identifier used internally by the system.

PROJECT.DB This table contains 1 to many project records for each EIN as applicable.

EIN_Code
Size & Type: A12*
Required: True
Folder:
Example: EIN000000033
Restrictions:
Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

Project_No
Size & Type: A10*
Required: True
Folder: Pointer
Example: B-534A
Restrictions:
Information: Unique identifier assigned to every Hanford Project.
**LockID**
Size & Type: I
Required: True
Folder: 
Example: 
Restrictions: 
Information: This is the identifier for the lock session. This data is used internally by the system.

**LockRec**
Size & Type: I
Required: True
Folder: 
Example: 
Restrictions: 
Information: This is the identifier for the record being modified. This data is used internally by the system.

**RecordID**
Size & Type: +
Required: True
Folder: 
Example: 
Restrictions: 
Information: This is a record identifier used internally by the system.

**SYS_DWG.DB** This table contains 1 to many system drawing records for each EIN as applicable.

**EIN Code**
Size & Type: A12*
Required: True
Folder: 
Example: EIN000000033
Restrictions: 
Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

**Drawing_No**
Size & Type: A11*
Required: True
Folder: References
Example: H-14-011733
Restrictions: H-14-
Information: System Drawing Number. Unique number assigned to the H-14 O&M system drawing.
Sheets
  Size & Type: A3*
  Required: True
  Folder: References
  Example: 001
  Restrictions: nnn
  Information: System Drawing Sheet Number. Unique number assigned to the different sheets of the same System Drawing Number.

Revs
  Size & Type: A3
  Required: True
  Folder: References
  Example: 001
  Restrictions: nnn
  Information: System Drawing Revision Number. A unique number used to track the revision of a System Drawing.

Zones
  Size & Type: A2
  Required:
  Folder: References
  Example: A1
  Restrictions: An-In
  Information: System Drawing Zone Location. Zone area where the EIN appears on the System Drawing.

XYZ_Coordinates
  Size & Type: A20
  Required:
  Folder: References
  Example: 26.975 21.398 00.000
  Restrictions:
  Information: System Drawing XYZ Coordinate Location. AutoCAD XYZ. Coordinates where the EIN appears on the System Drawing.

Drawing_Status
  Size & Type: A3
  Required:
  Folder:
  Example:
  Restrictions: lookup
  Information: This is a link to the Dwgstat.DB, Status Description field.
LockID
  Size & Type: I
  Required:
  Folder:
  Example:
  Restrictions:
  Information: This is the identifier for the lock session. This data is used internally by the system.

LockRec
  Size & Type: I
  Required:
  Folder:
  Example:
  Restrictions:
  Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
  Size & Type: +
  Required:
  Folder:
  Example:
  Restrictions:
  Information: This is a record identifier used internally by the system.

**REF_DWG.DB** This table contains 1 to many reference drawing records for each EIN as applicable.

EIN Code
  Size & Type: A12*
  Required: True
  Folder:
  Example: EIN00000033
  Restrictions:
  Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

Old_ID
  Size & Type: A20*
  Required:
  Folder: References
  Example: WFT-03C-2
  Restrictions:
  Information: Alias or old identification number.
Reference Source
Size & Type: A20
Required: True
Folder: References
Example: H-14-010112
Restrictions: H-14-
Information: System Drawing Number. Unique number assigned to the H-14 O&M System Drawing.

Sheet
Size & Type: A3
Required: True
Folder: References
Example: 001
Restrictions: nnn
Information: Reference Drawing Sheet Number. Unique number assigned to the different sheets of the same Reference Drawing Number.

Rev
Size & Type: A3
Required: True
Folder: References
Example: 001
Restrictions: nnn
Information: Reference Drawing Revision Number. A unique number used to track the revision of a System Drawing.

Zoner
Size & Type: A2
Required:
Folder: References
Example: A1
Restrictions: An-Hn
Information: Reference Drawing Zone Location. Zone area where the EIN appears on the Reference Drawing.

XYZ_Coorinater
Size & Type: A20
Required:
Folder: References
Example: 26.975 21.398 00.000
Restrictions:
Information: Reference Drawing XYZ Coordinate Location. AutoCAD XYZ Coordinates where the EIN appears on the Reference Drawing.
LockID
  Size & Type: I
  Required:
  Folder:
  Example:
  Restrictions:
  Information: This is the identifier for the lock session. This data is used internally by the system.

LockRec
  Size & Type: I
  Required:
  Folder:
  Example:
  Restrictions:
  Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
  Size & Type: +
  Required:
  Folder:
  Example:
  Restrictions:
  Information: This is a record identifier used internally by the system.

SAFE_SRC.DB This table contains 1 to many safety source document records for each EIN as applicable.

EIN_Code
  Size & Type: A12*
  Required: True
  Folder:
  Example: EIN000000033
  Restrictions:
  Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

Source_Doc
  Size & Type: A25*
  Required: True
  Folder: Safety
  Example:
  Restrictions:
  Information: Safety Source Document. Source documents are used to determine Safety Class.
Source_Sec
Size & Type: A10*
Required: True
Folder: Safety
Example:
Restrictions:
Information: Safety Source Document Section. Source Document Section used to determine Safety Class.

LockID
Size & Type: I
Required:
Folder:
Example:
Restrictions:
Information: This is the identifier for the lock session. This data is used internally by the system.

LockRec
Size & Type: I
Required:
Folder:
Example:
Restrictions:
Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
Size & Type: +
Required:
Folder:
Example:
Restrictions:
Information: This is a record identifier used internally by the system.

SAFE_TNK.DB This table contains 1 to many tank records for each EIN as applicable.

EIN_Code
Size & Type: A12*
Required: True
Folder:
Example: EIN000000033
Restrictions:
Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.
Tank_ID
   Size & Type: A10*
   Required: True
   Folder: Safety
   Example:
   Restrictions:
   Information: Tank Identification. Associated tanks.

LockID
   Size & Type: I
   Required:
   Folder:
   Example:
   Restrictions:
   Information: This is the identifier for the lock session. This data is used internally by the system.

LockRec
   Size & Type: I
   Required:
   Folder:
   Example:
   Restrictions:
   Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
   Size & Type: +
   Required:
   Folder:
   Example:
   Restrictions:
   Information: This is a record identifier used internally by the system.

SPR_PRTS.DB This table contains 1 to many spare parts records for each EIN as applicable.

EIN_Code
   Size & Type: A12*
   Required: True
   Folder:
   Example: EIN000000033
   Restrictions:
   Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.
Sp_Parts_No
Size & Type: A10*
Required: True
Folder: Pointer
Example: 6167-1411-11
Restrictions: nnnn-nnnn-
Information: Spare Part Number. Unique identifier used as the key field in the Spare Parts system.

LockID
Size & Type: I
Required:
Folder:
Example:
Restrictions:
Information: This is the identifier for the lock session. This data is used internally by the system.

LockRec
Size & Type: I
Required:
Folder:
Example:
Restrictions:
Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
Size & Type: +
Required:
Folder:
Example:
Restrictions:
Information: This is a record identifier used internally by the system.

ECN.DB  This table contains 1 to many procedure records for each EIN as applicable.

EIN_Code
Size & Type: A12*
Required: True
Folder:
Example: EIN000000033
Restrictions:
Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.
ECN
  Size & Type: A10*
  Required: True
  Folder: References
  Example: 608038
  Restrictions:
  Information: ECN Number. Unique identifier for ECNs.

LockID
  Size & Type: I
  Required:
  Folder:
  Example:
  Restrictions:
  Information: This is the identifier for the lock session. This data is used internally by the system.

LockRec
  Size & Type: I
  Required:
  Folder:
  Example:
  Restrictions:
  Information: This is the identifier for the record being modified. This data is used internally by the system.

RecordID
  Size & Type: +
  Required:
  Folder:
  Example:
  Restrictions:
  Information: This is a record identifier used internally by the system.

SYSTEMS.DB This table contains the lookup values for the various systems.

System
  Size & Type: A5*
  Required: True
  Folder: Utility / Definitions / EIN Systems
  Example: CHMA
  Restrictions:
  Information: This is the unique link field for the system value. It is an abbreviation of the system name.
System_Name
  Size & Type: A30
  Required: True
  Folder: Utility / Definitions / EIN Systems
  Example: FUEL OIL
  Restrictions: Information: System Name.

System_Number
  Size & Type: A2
  Required: True
  Folder: Utility / Definitions / EIN Systems
  Example: 10
  Restrictions: Information: System Number.

COMP_DES.DB  This table contains the lookup values for the various components.

Component
  Size & Type: A5*
  Required: True
  Folder: Utility / Definitions / EIN Components
  Example: AOD
  Restrictions: Information: This is the unique link field for the component value. It is an abbreviation of the component name.

Comp_Name
  Size & Type: A55
  Required: True
  Folder: Utility / Definitions / EIN Components
  Example: BATTERY
  Restrictions: Information: Component Name.

FAC_FARM.DB  This table contains the lookup values for the various farm facilities.

Farm_Facility
  Size & Type: A5*
  Required: True
  Folder: Utility / Definitions / Farm/Facilities
  Example: 242A
  Restrictions: Information: This is the unique link field for the farm facility value.
Structure_Description
Size & Type: A65
Required: True
Folder: Utility / Definitions / Farm/Facilities
Example: This is the AN Farm.
Restrictions:
Information: Description of the structure.

EINSTAT.DB This table contains the lookup values for the various EIN statuses.

Status
Size & Type: A3*
Required: True
Folder: Utility / Definitions / EIN Statuses
Example: 1
Restrictions:
Information: This is the unique link field for the status value.

Status_Description
Size & Type: A30
Required: True
Folder: Utility / Definitions / EIN Statuses
Example: Label Made
Restrictions:
Information: Status Description

Default_Initial_Status
Size & Type: L
Required: True
Folder: Utility / Definitions / EIN Statuses
Example: 
Restrictions:
Information: Default Initial Status, boolean value.

NOUN_ABB.DB This table contains the lookup values for the various noun names.

Noun_Name_Abbreviation
Size & Type: A10*
Required: True
Folder: Utility / Definitions / Noun Name Abbreviations
Example: ADJ
Restrictions:
Information: This is the unique link field for the noun name value.
Abbreviation_Definition
Size & Type: A45
Required:
Folder: Utility /Definitions /Noun Name Abbreviations
Example: ADJUST
Restrictions:
Information: Definition of the noun name.

Equip_Sys_Name
Size & Type: A30
Required:
Folder: Utility /Definitions /Noun Name Abbreviations
Example: IA REG
Restrictions:
Information: Equipment System Name.

**COMP_TYP.DB** This table contains the lookup values for the various component types.

Comp_Func
Size & Type: A20*
Required: True
Folder: Utility /Definitions /Component Functions
Example: VERTICAL
Restrictions:
Information: Component Type.

Comp_Code
Size & Type: A2
Required: True
Folder: Utility /Definitions /Component Functions
Example: 04
Restrictions: nn
Information: This is the unique link field for the component type value.

**DWGSTAT.DB** This table contains the lookup values for the various drawing statuses.

Status
Size & Type: A3*
Required: True
Folder: Utility /Definitions /Drawing Statuses
Example: 1
Restrictions:
Information: This is the unique link field for the status value.
StatusDescription
Size & Type: A30*
Required: True
Folder: Utility/Definitions/Drawing Statuses
Example: Label Made
Restrictions:
Information: Status Description

Default_Initial_Status
Size & Type: L
Required: True
Folder: Utility/Definitions/Drawing Statuses
Example:
Restrictions:
Information: Default Initial Status, boolean value.

USER.DB This table contains the MEL users and related information.

Hanford_No
Size & Type: A12*
Required: True
Folder: Admin/Access Control
Example: H0068719
Restrictions: Hnnnnnn
Information: Hanford ID. Unique link field.

Name
Size & Type: A40
Required: True
Folder: Admin/Access Control
Example: Doe, J.D.
Restrictions:
Information: User name.

Privilege
Size & Type: A8
Required: True
Folder: Admin/Access Control
Example: 11111111
Restrictions: mnnnnnn
Information: User privilege. Automatically generated by the system.
Password
  Size & Type: A10
  Required:
  Folder: Admin / Access Control
  Example: TESTER
  Restrictions:
  Information: User password.

REASONS.DB This table contains the values for the various reasons for data editing.

Reasons
  Size & Type: A30
  Required: True
  Folder: Utilities / Definitions / Edit Reasons
  Example: Delete Record
  Restrictions:
  Information: Reason description.

2.4.2 SYSTEM TABLES

EIN_IN.DB This table contains the next available EIN_Code.

EIN_Code
  Size & Type: A12*
  Information: This record identifier will be generated automatically by the system. It is a counter used to grant a unique ID for each EIN record.

LockID
  Size & Type: I
  Required:
  Folder:
  Example:
  Restrictions:
  Information: This is the identifier for the lock session. This data is used internally by the system.

SessionNo
  Size & Type: I
  Required:
  Folder:
  Example:
  Restrictions:
  Information: This is the identifier for the session. This data is used internally by the system.
RPT_SQLS.DB  This table contains the SQL queries.

Table_Name
    Size & Type: A8*
    Information: This value specifies which table the query is specified for.

SQL_Query
    Size & Type: M240
    Information: This field contains the SQL code for the query associated with the appropriate table.

SESSNLOG.DB  This table logs data relevant to the edit session.

SessionID
    Size & Type: I*
    Information: This record identifier will be generated automatically by the system. It uniquely identifies a session.

SessionReleaseNo
    Size & Type: I*
    Information:

SessionUser
    Size & Type: A40
    Information: User name for the edit session.

SessionDate
    Size & Type: A20
    Information: Date for the edit session.

SessionTime
    Size & Type: A20
    Information: Time for the edit session.

SessionReason
    Size & Type: A30
    Information: Reason for the edit session.

SessionDocument
    Size & Type: A25
    Information: Document related to the edit session.

ReleaseUser
    Size & Type: A40
    Information: User name of release verification individual.

ReleaseDate
    Size & Type: A20
    Information: Date of release.
ReleaseTime
  Size & Type: A20
  Information: Time of release.

LOCKS.DB This table logs lock user information relevant to set of locked data records.

LockID
  Size & Type: +*
  Information: Automatically incremented unique lock identifier.

LockUser
  Size & Type: A40
  Information: User name of individual locking data for modification.

LockDate
  Size & Type: A20
  Information: Date of lock session.

LockTime
  Size & Type: A20
  Information: Time of lock session.

LockReason
  Size & Type: A30
  Information: Reason for locking data.

LockDocument
  Size & Type: A25
  Information: Document related to lock session.

CHANGLOG.DB This table logs all data modifications.

SessionID
  Size & Type: I*
  Information: Identifier for change session.

SessionReleaseNo
  Size & Type: I*
  Information: Identifier for the release number.

SessionRec
  Size & Type: I*
  Information: Identifier for changed record.
EIN
   Size & Type: A23*
   Information: Modified EIN.

ChangedTable
   Size & Type: A12*
   Information: Name of table containing change.

ChangedField
   Size & Type: A25*
   Information: Name of field containing change.

Action
   Size & Type: A6
   Information: Type of change.

NewValue
   Size & Type: A30
   Information: New value.

OldValue
   Size & Type: A30
   Information: Old value.

LOCKDATA.DB This table logs the data relevant to a lock session.

LockID
   Size & Type: I*
   Information: Link field to the Lock information in Locks.DB.

LockRec
   Size & Type: I*
   Information: Identifier for the locked record.

EIN
   Size & Type: A23*
   Information: Locked EIN.

LockedTable
   Size & Type: A12*
   Information: Locked Table.

LockedField
   Size & Type: A25*
   Information: Locked Field.
Action
  Size & Type: A6
  Information: Reason for locking data.

PendingValue
  Size & Type: A60
  Information: New data value pending verification.

CurrentValue
  Size & Type: A60
  Information: Data value before lock.

**QRYFLDS.DB** This table contains field information associated with a query.

Order
  Size & Type: I*
  Information: Order of table records.

TableName
  Size & Type: A12
  Information: Name of table.

FieldName
  Size & Type: A25
  Information: Name of field.

DataGroup
  Size & Type: A15
  Information: Name of data group.

DataType
  Size & Type: A1
  Information: Type of data.

LookupTableName
  Size & Type: A12
  Information: Name of lookup table.

LookupFieldName
  Size & Type: A25
  Information: Name of lookup field.

DisplayName
  Size & Type: A25
  Information: User-friendly display name.
EditMask
  Size & Type: A75
  Information: System compatible field mask identifying required and optional character specifications.

DisplayMask
  Size & Type: A60
  Information: User-friendly version of the edit mask used for user help.
3.0 PROGRAMMING

3.1 PLATFORM

The MEL Phase I will operate on the user’s desktop PC under the current Hanford End User Operating Environment (ESOE). Currently, this includes Windows Version 3.11 (Windows for Workgroups) and DOS Version 6.22 and Windows95. Users that do not operate under this configuration will be unable to execute the MEL system.

All MEL data will be stored in Borland Paradox for Windows Version 5.0 data tables. These are binary files that may only be read using a database table interpreter or “engine”.

Borland Delphi Version 1.0 will be used as the application development tool for the MEL system. Delphi is a fourth generation object oriented programming language (4GL) that provides the developer with the flexibility needed to satisfy all the MEL requirements in the most efficient and cost-effective manner.

Several third-party component packages will be used in building the MEL system. A component is a single binary file that may be directly integrated into the Borland Delphi development environment. Although these components are transparent to the common user, they provide the developer with enhanced programming capabilities. Components used in the MEL include InfoPower Version 1.0 and Crystal Reports Component Version 7.0. InfoPower is a package of components which provide the developer with enhanced data aware modules, while Crystal Reports Component is a single component which allows the integration of custom reports.

Installation of the MEL will be accomplished using the OnDemand WINStall Version 5.0 program. This application has the flexibility to be configured specifically for the needs of the MEL system. The WINStall program files specific to MEL will be maintained on the MEL fileserver.

Crystal Reports Version 4.5 by Seagate Inc. will be used for the development of the MEL reports. Crystal Reports is a separate program that is flexible enough to support multiple types of databases and programming languages. These reports will be integrated into the MEL system using the Crystal Reports Component discussed above.

The MEL database tables will be stored on a fileserver using Windows NT Version 3.51 as the operating system. This configuration allows the fileserver administrator to setup “user groups” which are groups of individuals allowed access the MEL data.
3.2 USER INTERFACE

3.2.1 PROGRAM UNITS

The MEL system, written in Borland Delphi 1.0, is compiled into a single executable file which is placed on the user's desktop PC during the MEL installation process. However, the single executable file is actually made up of several Pascal files (Pascal files end with a .pas extension). Pascal files contain the actual source code and are native to Borland Delphi, since Pascal was used in the development of Delphi.

Pascal files (which are referred to as “units” in Delphi) will be strategically designed for the MEL system. A description of each is as follows:

ABOUT.PAS - This unit contains the source code for the “About” dialog screen.

ADDEIN.PAS - This unit contains the source code for the “Add New EINs” dialog screen.

CHG_ACTV.PAS - This unit contains the source code for the “Change Control Activity” dialog screen.

CHGPNDG.PAS - This unit contains the source code for the “Changes Pending” dialog screen.

EDITVIEW.PAS - This unit contains the source code for the editview (folders) screen.

FIND.PAS - This unit contains the source code for the find dialog screen.

INSPNDG.PAS - This unit contains the source code for the new records dialog screen.

JUSTIFY.PAS - This unit contains the source code for the edit reason dialog screen.

LOGON.PAS - This unit contains the source code for the “Logon” dialog screen.

MAIN.PAS - This unit contains the source code for the MEL Phase I main screen. This includes the navigation tools, main menu and control buttons.

PRNTFILE.PAS - This unit contains the source code for the “Export File Type” dialog screen.
This unit contains the source code for the "Print Data Locks" dialog screen.

This unit contains the source code for the "Query Results" dialog screen.

This unit contains the source code for the "Preparing your report, please wait" dialog screen.

This unit contains the source code for the "Query" dialog screen.

This unit contains the source code for the "Reports" dialog screen.

This unit contains the source code for the MEL splash screen, which is visible at startup.

This unit contains the source code for the "Standard Data Definitions" dialog screen.

This unit contains the source code for the "User Information" dialog screen.

3.2.1.1 EDITVIEW UNIT

The complexity of the Editview unit requires that it be documented in explicit detail. Other units in the MEL are not as complex, and are therefore self-documenting.

Below is a list of the procedures and functions in the frmEditView unit for the MEL, phase I, version 2.0. The list contains routine declarations as they appear (and in the same order they appear) in the unit's source file, EDITVIEW.PAS. Where useful, a brief description of the routine's function is provided. Also, they are cross-referenced to the MEL, Phase I Data Editing Structure Charts. One or more numbers in square brackets, e.g., [2, 4], indicate the structure chart pages that illustrate the function or procedure.

Version 1.0 routines kept...

procedure   FormCreate (Sender: TObject);
procedure   SetPointerFolder (mode: String);
procedure   SetReferenceFolder (mode: String);
procedure   SetNameplateFolder (mode: String);
procedure SetSpecificationsFolder (mode: String);
procedure SetSafetyFolder (mode: String);
procedure SetLabelAdminAccess (mode: String);
procedure MEL_TabsChange (Sender: TObject; NewTab: Integer;
var AllowChange: Boolean);

Data entry format checking (where 'masks' leave off)...

procedure fldSafetyClassChange (Sender: TField);
procedure fldOSRChange (Sender: TField);
procedure fldEnvironmentalChange (Sender: TField);
procedure fldQAChange (Sender: TField);
procedure fldBearingsSealedFChange (Sender: TField);
procedure fldBearingsSealedRChange (Sender: TField);
procedure fldZonerChange (Sender: TField);
procedure fldSysZonesChange (Sender: TField);
function CheckZoneValue (zone: String):Boolean;
procedure ForceUpperCase (Sender: TField);
function AutoFillLead (s: string; sz: integer; c: string): string;
procedure fldSysRevsChange (Sender: TField);
procedure fldSysSheetsChange (Sender: TField);
procedure fldRefRevrChange (Sender: TField);
procedure fldRefSheetrChange (Sender: TField);

Event handlers to capture changes...

procedure dsUpdateData (Sender: TObject);

This handler is fired (by the Post or UpdateRecord methods of a dataset) when the current record is about to be updated in the database. This event occurs early enough that an actual post can be canceled.

procedure grdKeyDown (Sender: TObject; var Key: Word;
Shift: TShiftState);

Capture attempts to insert and delete records. These operations must be performed via the pop up menu.

procedure System_LU_boxEnter (Sender: TObject);
procedure Comp_LU_boxEnter (Sender: TObject);
procedure Status_LU_boxEnter (Sender: TObject);
procedure Fac_LU_boxEnter (Sender: TObject);
procedure System_LU_boxCloseUp (Sender: TObject; LookupTable: TwwTable;
FillTable: TTable; modified: Boolean);
procedure Comp_LU_boxCloseUp (Sender: TObject; LookupTable: TwwTable;
FillTable: TTable; modified: Boolean);
procedure Status_LU_boxCloseUp (Sender: TObject; LookupTable: TwwTable;
FillTable: TTable; modified: Boolean);
procedure Fac LU_boxCloseUp (Sender: TObject; LookupTable: TwwTable;
FillTable: TTable; modified: Boolean);
procedure EvaluateEIN;

Pop up menu invocation and function setup support...

The pop up menu of options for data fields is always accessible to the user when a right
button mouse click is made on an edit box or grid. Upon right click, the following processes
are invoked.

procedure edMouseDown (Sender: TObject; Button: TMouseButton;
[1]
    Shift: TShiftState; X, Y: Integer);

This handler is attached to all TDBEdit components used to edit MEL data fields. If the
mouse click is on the right button, this save the associated table and data source and call
invoke the pop up menu, popEditor. Although it would seem more direct to specify the pop
up menu in each of the TDBEdit components (which is then automatically invoked by a right-
click on the edit box), this prevents the table and data source associated with the edit box
from being identified.

procedure grdMouseDown (Sender: TObject; Button: TMouseButton;
[1]
    Shift: TShiftState; X, Y: Integer);

This handler is attached to the TwwDBGrid components used to edit MEL data and serves
the same purpose as the edMouseDown handler but for grids.

procedure popEditPopup (Sender: TObject);
[1]
This handler is attached to the popEditor pop up menu component. When called, this procedure determines the current user session conditions and enables or disables the individual menu items appropriately.

After this handler executes, the pop up menu is displayed to the user. Processes are suspended until the user makes a menu selection or moves off the menu and activates another handler elsewhere (effectively canceling the menu).

When the pop up menu is displayed, the user may select an enabled menu item or click elsewhere on the display. Each menu item has its own event handler named ‘piXXXClick’ where XXX is the name of the selected item.

Pop up menu item handlers...

procedure piChangesClick (Sender: TObject);
[2]

The user has requested to view the changes pending for the selected table, record, and field. Make sure any edits in progress are cleaned up, get the pertinent information, then display the dialog for changes pending. If the user is the owner of the data, we're in edit mode, and the changes pending do not represent a deletion, allow editing within the dialog. Update displayed data accordingly upon return.

procedure piNewRecordsClick (Sender: TObject);
[2]

The user has requested to view all new records (insertions) for the selected table. Display the dialog for new records pending.

procedure piLockClick (Sender: TObject);
[3]

Lock the selected table & record for the user. Do not enable for editing.

procedure piEditClick (Sender: TObject);
[3]

The user wants to edit the selected table & record. Lock the record for the user and enable editing. If the selected field is in the EIN INFO table, all dependent records in the other tables must be free or owned by the editing user.
procedure piPostClick (Sender: TObject);

The user wants to post any changes pending for the single selected record. Confirm post with user.

procedure piCancelClick (Sender: TObject);

The user wants to cancel any changes pending for the single selected record. Confirm cancellation with user.

procedure piDeleteClick (Sender: TObject);

The user wants to mark the selected record for deletion. Get confirmation, clean up edits, lock the record, and mark it for deletion.

procedure piNewClick (Sender: TObject);

The user wants to insert a new record. Clean up any open edits. Initialize a new record for the table, lock it, enable editing, etc. Note - the record will later be deleted after field entries are recorded.

Data operations access support...

function IsPrivileged (pIdx: integer): Boolean;

Return true only if the session user has correct privilege to edit the folder indexed by the given index.

function IsOwner (lockID: integer): Boolean;

Return true if the session user owns the selected record.
function GetPendingAction ( lockID, lockRec: integer;
           ein, tName, fName: string ): string;

Return the action pending for the first LOCKDATA record matching the given parameters. If no matching record is located, return 'Error'.

function AnyNewRecords ( ein, tName: string ): Boolean;

Determine if there are any new records pending (insertions) for the given EIN and table. If so, return true.

Lock and change indication support...

procedure DetermineFieldColors ( tbl: Tdatabase; var fg, bg: TColor );

Determine the appropriate foreground (font) and background colors for the current record in the given table. This is the only function used to determine these colors.

procedure UpdateFieldColors;

Force an update of all field colors by triggering the OnChange handler for the tables.

procedure EINInfoLockIDChange (Sender: TObject);

This handler fires whenever the LockID in the EIN_Info table changes as the result of navigating records or changing the current record's lock status. Every edit box associated with the table must be updated separately.

procedure EqpInfoLockIDChange (Sender: TObject);

This handler fires whenever the LockID in the Eqp_Info table changes as the result of navigating records or changing the current record's lock status. Every edit box associated with the table must be updated separately.
procedure MotorLockIDChange (Sender: TObject);
[2]

This handler fires whenever the LockID in the Motor table changes as the result of navigating records or changing the current record's lock status. Every edit box associated with the table must be updated separately.

procedure ElectricLockIDChange (Sender: TObject);
[2]

This handler fires whenever the LockID in the Electric table changes as the result of navigating records or changing the current record's lock status. Every edit box associated with the table must be updated separately.

procedure MechanicLockIDChange (Sender: TObject);
[2]

This handler fires whenever the LockID in the Electric table changes as the result of navigating records or changing the current record's lock status. Every edit box associated with the table must be updated separately.

procedure grdCalcCellColors (Sender: TObject; Field: TField; State: TGridDrawState; highlight: Boolean; AFont: TFont; ABrush: TBrush);

Set the font and background colors for each cell. Function DeterminFieldColors will determine the appropriate colors based upon the lockID for each record.

procedure ShowPreviousChanges ( tbl: TwwTable );
[2,3]

Replace the viewed data with values from the changes pending. This display of changes is temporary and local.

Lock and Edit support...

function LockSingleRecord ( tbl: TwwTable; dsrc: TwwDataSource;
[3]
lockedit: integer; einCode: string ): Boolean;
If available, lock the current record in the given table. Create a new lockID if needed. Return true if record is locked to vUser.

function LockManyRecords ( tbl: TwwTable; dsr: TwwDataSource; lockedit: integer; einCode: string ): Boolean;

Lock all records in the given table that match the given einCode. Create a new LockID if necessary. Return true if successful, false if not.

function LockEINRecord ( einCode: string ): Boolean;

Check all associated records for the EIN before locking. If any records are locked, return false. If not, lock them and return true.

function GetSingleLockStatus ( tbl: TwwTable; einCode: string ): integer;

Return the status of the current single record in the given table. If the record's ein_code does not match, return cNoRecord.

function GetManyLockStatus ( tbl: TwwTable; einCode: string ): integer;

Return the combined status for all records in the given table, tbl, that match the given einCode.

function CreateLock ( user, reason, document: string ): integer;

Create an entry in the LOCKS table and return the new lockID.

procedure CopyRecord ( tbl: TwwTable );

Copy the names of the selected table fields and then copy the contents of the fields into a memory array.
procedure Detour ( dsrc: TwwDataSource );
[3,4,6,9]

This routine acts as a detour for data when a normal post of changes would occur. Record changes in the LOCKDATA table and cancel the impending post. If the record was just inserted, delete it from the database table.

procedure SaveDatabaseRecord ( tbl: TwwTable; action: string );
[3,8]

Save the current record for the given table, 'tbl', in LOCKDATA with the given 'action'. At this point, the tbl is already pointing to the record of interest, the current field values have been saved in the vRcdValues TStringList, and the field names have been copied into the vRcdFields TStringList.

procedure SaveDatabaseChanges ( tbl: TwwTable );
[9]

Save the values of changed fields for the current record of the given table, 'tbl', in LOCKDATA. At this point, the tbl is already pointing to the record of interest, the original field values have been saved in the vRcdValues TStringList, and the field names have been copied into the vRcdFields TStringList.

procedure SaveDatabaseDeletion ( tbl: TwwTable );
[7]

Save the values of fields for the current record of the given table, 'tbl', in LOCKDATA - they are to be deleted. At this point, the tbl is already pointing to the record of interest, the original field values have been saved in the vRcdValues TStringList, and the field names have been copied into the vRcdFields TStringList.

procedure SaveDatabaseDeletions ( tbl: TwwTable; einCode: string );
[7]

Save all records in the given table that match the current EIN.

procedure SaveEINDeletion ( einCode: string );
[7]

Save each of the constituent records for an EIN.
procedure InitDatabaseRecord ( tbl: TwwTable );

Initialize the new record in the given table.

procedure CreateLockDataRecord ( lockID, lockRec: integer;
newValue, oldValue, action, ein, tblName, fldName: string );

Insert a new record in the LOCKDATA table and fill with the given data.

procedure UpdateLockDataRecord ( newValue, oldValue, action: string );

The correct LOCKDATA record is already pointed at by tblLockData. Change the pending value, current value, and action as given.

function LocateLockDataRecord ( ein, tName, fName: string;
lockID, lockRec: integer ): Boolean;

Given the match parameters required, locate the LOCKDATA record and leave the tblLockData pointing at the record. Return true if the match is successful, false if not found.

Cancel support...

function CancelChangesPendingRecord ( lockID, lockRec: integer;
ein, tName: string ): Boolean;

Remove the lock on the given table and record and remove the associated LOCKDATA entries. Remove the LOCKS entry if appropriate. Return true if successful, false if not.

function CancelEINChangesPending ( lockID: integer; ein: string ): Boolean;

Remove the locks on all records in all database tables for the given EIN. Then remove the associated LOCKDATA and the LOCKS entry if appropriate. Return true if successful, false if not.
function CancelSessionChangesPending ( lockID: integer ): Boolean;

Cancel all locks and changes made in one session by a specific user. Return true if successful, false if not.

function CancelUsersChangesPending ( user: string): Boolean;

Cancel all locks and changes made in any sessions by a specific user. Return true if successful, false if not.

function CancelAllChangesPending: Boolean;

Cancel all locks and changes pending. Return true if successful, false if not.

function UnlockDatabaseRecord ( lockID, lockRec: integer;
[6]
einCode, tName: string ): Boolean;

Remove the LockID and LockRec from a single data table record. Return true if successful, false if not.

function UnlockEINDatabaseRecords ( einCode: string): Boolean;
[6]

Remove the LockID (and LockRec) from ALL data tables records with the matching einCode. Return true if successful, false if not.

function UnlockSessionDatabaseRecords ( lockID: integer ): Boolean;

Remove the LockID (and LockRec) from the data tables records with the given lockID. Return true if successful, false if not.

function UnlockAllDatabaseRecords: Boolean;

Remove the LockID (and LockRec) from ALL data tables records. Return true if successful, false if not.

function DeleteLockDataLockRec ( lockID, lockRec: integer;
[5,6]
ein, tName: string ): Boolean;
Remove all LockData entries for the given LockID and LockRec. Return true if successful, false if not.

```haskell
defunction DeleteEINLockData ( lockID: integer; ein: string ): Boolean;
```

Remove all LOCKDATA entries for the given LockID and EIN. Return true if successful, false if not.

```haskell
defunction DeleteSessionLockData ( lockID: integer ): Boolean;
```

Return true if successful, false if not.

```haskell
defunction DeleteAllLockData: Boolean;
```

Return true if successful, false if not.

```haskell
defunction CleanupLocks ( LockID: integer ): Boolean;
```

If no LockData entries exist for the LockID, clear the associated Locks entry. Return true if successful, false if not.

```haskell
defunction ClearLocks: Boolean;
```

Clear the Locks.db table. Return true if successful, false if not.

Post support...

```haskell
defunction PostChangesPendingRecord ( lockID, lockRec: integer; ein, tName: string ): Boolean;
```

Post the pending changes to a specific data table & record and release. Return true if successful, false if not.

```haskell
defunction PostEINChangesPending ( lockID: integer; ein: string ): Boolean;
```

Post all changes made for the given EIN. Return true if successful, false if not.
function PostSessionChangesPending ( lockID: integer ): Boolean;
Post all the changes made in one session. Return true if successful, false if not.

function PostUsersChangesPending ( user: string ): Boolean;
Post all changes made by a given user. Return true if successful, false if not.

function PostAllChangesPending: Boolean;
Post all changes pending. Return true if successful, false if not.

function RecordIsLockedOnly ( lockID, lockRec: integer; 
ein, tName: string ): Boolean;
If all matching LOCKDATA records have 'Lock' action, return true.

function LogPostingSession ( lockID: integer ): integer;
Create a session log record based upon the locks record and the current user making the release. Return the SessionID.

function LogAction ( lockID, releaseNo, lockRec: integer; 
ein, tName: string ): Boolean;
Copy the LOCKDATA records to the CHANGLOG for the given record set. Return true if successful, false if not.

function GetSessionReleaseNo ( lockID: integer ): integer;
Search SESSNLOG for lockID. If found get largest associated SessionReleaseNo, increment it and return it. If lockID not found, return 1.

function GetFirstAction ( lockID, lockRec: integer; 
ein, tName: string ): string;
For a given lockID and lockRec, get the action for the first matching LOCKDATA record.

function **DeleteDatabaseRecord** ( lockID, lockRec: integer;
[5]
einCode, tName: string ): Boolean;

If the table is not EIN_Info, delete the single record. If it is the EIN_Info table, delete the record and all associated records in all of the other data tables (matching EIN_Codes). Return true if successful, false if not.

function **InsertDatabaseRecord** ( lockID, lockRec: integer;
[5]
ein, tName: string ): Boolean;

If the table is not EIN_Info, insert a new record in the table and fill it with the values in LOCKDATA. If the table is EIN_Info, a new EIN_Code must be created. Return true if successful, false if not.

function **UpdateDatabaseRecord** ( lockID, lockRec: integer;
[5]
ein, tName: string ): Boolean;

Modify the fields in the single table/record which have 'Change' action LOCKDATA records. Leave 'Lock'ed fields unchanged. Return true if successful, false if not.

**General support...**

procedure **EnableTable** ( dsSrc: TwwDataSource );
[2,3,8]
Turn on the auto-edit property for the data source.

procedure **DisableTable** ( dsSrc: TwwDataSource );
[3,9]
Turn off the auto-edit property for the data source.

procedure **CheckForEdits**;
[3,4,6,7,8]
If a table is open for editing, call Detour to record and clean up.

function GetNewEINCode: String;
[5,8]
Open the EIN_No.DB table to obtain the next available number to be applied to an EIN_Code. Build the new EIN_Code value. If successful, post the new value back to the EIN_No.DB.

function OkTOAddRecord ( tbl: TwwTable ): Boolean;
[8]
Determine if a new record is appropriate for the given table. This function does not consider the contents of LOCKDATA. Return true if Ok.

function GetOwner ( lockID: integer ): string;
[1,2,3]
Return the name of the user that owns the LockID.

function GetEINCode ( ein: string ): string;
[4,5,6]
Retrieve the EIN_Code for a given EIN. Don't change the EINInfo table position but check it first to see if the EIN matches.

procedure RefreshAll;
[4,6,7]
function Stripper ( textin: string ): string;
3.2.1.2 EDITVIEW STRUCTURE CHARTS

Right-click on any data edit box.

Right-click on any data grid cell.

This is not a direct call - rather, the procedure is invoked by calling the Popup method for the popEditor component.
MEL, Phase I
Data Editing Structure Charts
frmEditView - Page 6

User selects 'Cancel' popup menu item.

CheckForEdits

CancelEIN
ChangesPending

CancelChangesPendingRecord

RefreshAll

Detour
(see page 9)

GetEINCode
(see Page 4)

DeleteEIN
LockData

UnlockEIN
DatabaseRecords

CleanupLocks
(see Page 5)

UnlockDatabaseRecord

DeleteLockData
LockRec

CheckData

vTableNames

All Data Tables

LockData

EinCode

Ein

EinInfo

EinEditing

EinClickTable

EinLockTable

EinRec

EinName

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein

EinID

EinPending

EinFront

EinBack

Ein
The 'dsUpdateData' routine is called whenever a database table is about to be updated. The update is intercepted by the 'Detour' so that changes can be recorded as 'changes pending' in the LockData table. 'CheckForEds' is called by several routines whenever a significant change occurs.
3.2.2 FOLDERS

The MEL system divides its data into five major categories. These categories are represented in the application by “folders”. A folder is a single page of data contained on a screen with a tab-like index feature. The folders in the MEL include Pointer, References, Nameplate, Specifications, and Safety. Each folder contains data grouped specific to these topics.
3.2.3 SCREENS

The following sections contain pictures of the actual MEL user interface screens. Any data displayed is purely for example only.

3.2.3.1 DATA VIEW - POINTER FOLDER
### 3.2.3.2 DATA VIEW - REFERENCES FOLDER

![Diagram of Data View - References Folder]

#### Equipment Identification
- **TEST**
- **TEST15**
- **DS**
- **103**
- **008**
- **CHANGE TRAILER POWER (IN 273)**

#### Master Equipment List Phase I (BETA) [Data View and Entry]

<table>
<thead>
<tr>
<th>Procedure</th>
<th>System Branding No.</th>
<th>Sheet</th>
<th>XYZ Coordinates</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATP/SB0919</td>
<td>H-14-085641</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMP-333202</td>
<td>H-14-085659</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>H-14-085730</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>H-14-085750</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DM-M (ALM)</td>
<td>Reference Source</td>
<td>Sheet</td>
<td>XYZ Coordinates</td>
<td>Status</td>
</tr>
<tr>
<td>10-A</td>
<td>H-2-79962</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101-2-AN</td>
<td>H-2-70351</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.2.3.3 DATA VIEW - NAMEPLATE FOLDER

Master Equipment List Phase I (BETA) [Data View and Entry]

Equipment Identification

<table>
<thead>
<tr>
<th>Classification</th>
<th>Safety Case</th>
<th>Site</th>
<th>Critical</th>
<th>On/Off</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>Y</td>
<td></td>
<td>N</td>
</tr>
</tbody>
</table>

Name/Number

CHANGE TRAILER POWER (IN 273)

Motor

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Voltage (Volts)</td>
<td>115 V</td>
</tr>
<tr>
<td>Phase</td>
<td>3</td>
</tr>
<tr>
<td>Motor Size (HP)</td>
<td>1.5</td>
</tr>
<tr>
<td>Motor Speed (RPM)</td>
<td>1755</td>
</tr>
<tr>
<td>Unit (rpm)</td>
<td></td>
</tr>
<tr>
<td>Full Load (Amps)</td>
<td>240</td>
</tr>
<tr>
<td>Locked Rotor (Amps)</td>
<td>120</td>
</tr>
<tr>
<td>Service Factor</td>
<td>180</td>
</tr>
</tbody>
</table>

Mechanical

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testing</td>
<td>N</td>
</tr>
<tr>
<td>Reverse</td>
<td>Y</td>
</tr>
</tbody>
</table>

Electrical

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Supply Voltage</td>
<td></td>
</tr>
<tr>
<td>Line Voltage (Volts)</td>
<td></td>
</tr>
<tr>
<td>Rated Current</td>
<td>240 AMPS</td>
</tr>
<tr>
<td>Full Load (Amps)</td>
<td>120</td>
</tr>
<tr>
<td>Locked Rotor (Amps)</td>
<td>180</td>
</tr>
</tbody>
</table>

Date: 1/21 AM 5:23:34 AM
3.2.3.4 DATA VIEW - SPECIFICATIONS FOLDER

![Diagram of Master Equipment List Phase 1 (BETA) with specifications folder and data view]

- **Design**
  - Pressure: 60 PSI
  - Temperature: 140 F
  - Flow: 10 MCV
  - Output: 0-1 INMC
  - Service: 125 F

- **Operating**
  - Pressure: 125 PSI
  - Temperature: 90 F
  - Flow: 100 PSI
  - Output: 80 PSI
  - Service: 110 PSI

- **Classification**
  - Entry Class: 2
  - Other: Y
  - Other: N
3.2.3.5 DATA VIEW - SAFETY FOLDER
### 3.2.3.6 QUERY

![Master Equipment List Phase I (BETA) Query New Query](image)

<table>
<thead>
<tr>
<th>Field</th>
<th>Option</th>
<th>All Values</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Farm/Facility</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIN System</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIN Component</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EIN Number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component Function</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noun Name</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.2.3.7 REPORTS
3.2.3.8 LOCKS AND CHANGES PENDING

![Screen capture of Master Equipment List Phase I (BEIA) with a list of locks and changes pending]

<table>
<thead>
<tr>
<th>User</th>
<th>Field</th>
<th>Type</th>
<th>Status</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>THORNTON, MICHAEL W</td>
<td>EN_INFO.DB</td>
<td>Building</td>
<td>Label</td>
<td>Insert</td>
</tr>
<tr>
<td>THORNTON, MICHAEL W</td>
<td>EN_INFO.DB</td>
<td>Cal_Freq</td>
<td>Safety</td>
<td>Insert</td>
</tr>
<tr>
<td>THORNTON, MICHAEL W</td>
<td>EN_INFO.DB</td>
<td>Component</td>
<td>Label</td>
<td>Insert AO</td>
</tr>
</tbody>
</table>
| THORNTON, MICHAEL W | EN_INFO.DB | EIN     | Label  | Insert AS-
| THORNTON, MICHAEL W | EN_INFO.DB | Environmental | Safety | Insert |
| THORNTON, MICHAEL W | EN_INFO.DB | Farm_Facility | Label  | Insert 242 |
| THORNTON, MICHAEL W | EN_INFO.DB | Location_Desc | Insert  | |
| THORNTON, MICHAEL W | EN_INFO.DB | Loop_Prior | Safety | Insert |
| THORNTON, MICHAEL W | EN_INFO.DB | Loop_Seq_No | Pointer | Insert |
| THORNTON, MICHAEL W | EN_INFO.DB | Noun_Name | Label  | Insert AIR |
| THORNTON, MICHAEL W | EN_INFO.DB | OSR     | Safety | Insert |

Notice: Produced Changes are not approved until they are posted with the level and stamp.
3.2.3.9 CREATE NEW EINS

![Create New EINS Diagram]
### 3.2.3.10 STANDARD DATA DEFINITIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOD</td>
<td>AIR OPERATED DAMPER</td>
</tr>
<tr>
<td>ADV</td>
<td>AIR OPERATED VALVE</td>
</tr>
<tr>
<td>ARM</td>
<td>AREA RADIATION MONITOR</td>
</tr>
<tr>
<td>ASH</td>
<td>ANALYSIS SWITCH HIGH</td>
</tr>
<tr>
<td>ATS</td>
<td>AUTO TRANSFER SWITCH</td>
</tr>
<tr>
<td>BATT</td>
<td>BATTERY</td>
</tr>
<tr>
<td>BFP</td>
<td>BACK FLOW PREVENTER</td>
</tr>
<tr>
<td>BTL</td>
<td>BOTTLE</td>
</tr>
<tr>
<td>CAH</td>
<td>CONDUCTIVITY ALARM HIGH</td>
</tr>
<tr>
<td>CAL</td>
<td>CONDUCTIVITY ALARM LOW</td>
</tr>
<tr>
<td>CAM</td>
<td>CONTINUOUS AIR MONITOR</td>
</tr>
<tr>
<td>CASS</td>
<td>COMPUTER AUTOMATED SURVEILLANCE STATION</td>
</tr>
<tr>
<td>CBS</td>
<td>COMPONENT BUS C2</td>
</tr>
<tr>
<td>CCDS</td>
<td>CONTAINER DOCUMENT SYSTEM</td>
</tr>
<tr>
<td>CDS</td>
<td>CENTRAL DISTRIBUTION SYSTEM</td>
</tr>
<tr>
<td>CE</td>
<td>CONDUCTIVITY ELEMENT</td>
</tr>
</tbody>
</table>
### 3.2.3.11 CHANGE LOG

<table>
<thead>
<tr>
<th>Session/User</th>
<th>SessionDate</th>
<th>SessionTime</th>
<th>SessionAction</th>
<th>SessionDocument</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORDHAM, CHUCK</td>
<td>11/19/96</td>
<td>1:54:13 PM</td>
<td>MODIFY RECORD</td>
<td>ECN 55555</td>
<td>FO</td>
</tr>
<tr>
<td>FLYNN, DON</td>
<td>11/20/96</td>
<td>1:24:50 PM</td>
<td>EQUIPMENT MODIFICATION</td>
<td></td>
<td>FLY</td>
</tr>
<tr>
<td>FORDHAM, CHUCK</td>
<td>11/20/96</td>
<td>3:42:51 PM</td>
<td>DELETE RECORD</td>
<td>123454</td>
<td>FDI</td>
</tr>
<tr>
<td>JECH, JAMES B</td>
<td>11/20/96</td>
<td>4:05:01 PM</td>
<td>EDITORIAL CHANGE</td>
<td></td>
<td>JEC</td>
</tr>
</tbody>
</table>

#### Change Log

<table>
<thead>
<tr>
<th>EIN</th>
<th>Action</th>
<th>Field</th>
<th>Old</th>
<th>New Value</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>AN241-AS-V-116</td>
<td>Lock</td>
<td>Bar Code #</td>
<td>800</td>
<td>Pointer</td>
<td>MASONEILAN</td>
</tr>
<tr>
<td>AN241-AS-V-116</td>
<td>Lock</td>
<td>Loop/Seq #</td>
<td>801</td>
<td>Pointer</td>
<td>MASONEILAN</td>
</tr>
<tr>
<td>AN241-AS-V-116</td>
<td>Lock</td>
<td>Mfg. Name</td>
<td>MASONEILAN</td>
<td>Pointer</td>
<td>MASONEILAN</td>
</tr>
<tr>
<td>AN241-AS-V-116</td>
<td>Change</td>
<td>Mfg. #</td>
<td>1234</td>
<td>Pointer</td>
<td>40556</td>
</tr>
<tr>
<td>AN241-AS-V-116</td>
<td>Change</td>
<td>Model #</td>
<td>90876</td>
<td>Pointer</td>
<td>657</td>
</tr>
<tr>
<td>AN241-AS-V-116</td>
<td>Lock</td>
<td>Property #</td>
<td>90876</td>
<td>Pointer</td>
<td>657</td>
</tr>
<tr>
<td>AN241-AS-V-116</td>
<td>Change</td>
<td>Serial #</td>
<td>Pointer</td>
<td>657-112-3454-6670</td>
<td>657-112-3454-6670</td>
</tr>
<tr>
<td>AN241-AS-V-116</td>
<td>Lock</td>
<td>Vendor Info #</td>
<td>Pointer</td>
<td>657-112-3454-6670</td>
<td>657-112-3454-6670</td>
</tr>
</tbody>
</table>


3.2.3.12 ACCESS CONTROL
3.2.3.13 ABOUT

![Screenshot of the About dialog box showing version 1.1 BETA, September 26, 1996, and copyright information.]
3.2.4 MENUS

The following sections provide detail regarding the functionality of the MEL menus.

3.2.4.1 MAIN MENU

The MEL has several functions accessible through the menu aligned access the top of the MEL application main screen. This menu has the following options: File, Edit, Data, View, Utility, Admin., and Help. These menu items have specific options available once the user accesses them. They are as follows:

- **File / Exit** - This option allows the user to exit the MEL application.
- **Edit / Find** - This option allows the user to perform a search for a specific EIN by a secondary field.
- **Edit / Find Next** - Allows the user to automatically search for the next value matching the previously defined search criteria.
- **Data / Data View** - This function puts the database into view mode, posting any changes.
- **Data / Data Edit** - This function puts the database into edit mode, prompting the user for an edit reason.
- **View / Folders** - This function allows the user the ability to view folder data.
- **View / Queries** - This function allows the user the ability to perform a query.
- **View / Reports** - This function allows the user the ability to print a report.
- **View / Locks** - This function allows the user the ability to view and disposition locked data, if privileges are applicable.
- **View / Hint Boxes** - This function toggles the view of the yellow hint boxes providing user information when the mouse is held over certain fields.
- **Utility / Create EINs** - This function allows the user to create new EINs.
Utility / Definitions - This function allows the user to add, modify and remove master data definitions.

Utility / Change Log - This function allows the user to view the change log table.

Admin. / Access Control - This function allows the user to configure the MEL users list and access control.

Help / About - This function allows the user to view the release information regarding the current version of the MEL.

3.2.4.2 QUERY MENU

The menu on the “Query” page has the following options; File, Query, and Help. These menu items have specific options available once the user accesses them. They are as follows:

File / New Query - This option allows the user to create a new query.
File / Open Query... - This option allows the user to open an existing query.
File / Save Query - This option allows the user to save a query.
File / Save Query As... - This option allows the user to save a query with a different name.
File / Delete Query - This option allows the user to delete an existing query.
Query / Execute - This option allows the user to execute the current query.
Help - This option displays help regarding the use of the query page.

3.2.4.3 LOCKS AND PENDING CHANGES MENU

The menu on the “Locks and Pending Changes” page has the following options; View, Disposition, and Help. These menu items have specific options available once the user accesses them. They are as follows:

View / By User - This option allows the user to view locked data by user name.
View / By EINs - This option allows the user to view locked data by EIN.

View / Show All - This option allows the user to view all the data records, as opposed to only ones representing changes.

View / Cancel Session - This option allows the user to cancel the data for the selected session.

View / Cancel EIN - This option allows the user to cancel the data for the selected EIN.

View / Cancel User - This option allows the user to cancel the data for the selected user.

View / Cancel All - This option allows the user to cancel all the selected data.

View / Post Session - This option allows the user to post the data for the selected session.

View / Post EIN - This option allows the user to post the data for the selected EIN.

View / Post User - This option allows the user to post the data for the selected user.

View / Post All - This option allows the user to post all the selected data.

Help - This option displays help regarding the use of the query page.

3.2.5 REPORTS

The MEL Reports screen is built to accommodate several specific functions. First, the user may select to print data from one or many folders. Secondly, the user has the option to choose which EINs to report on by selecting them from a list. The user may also choose to print EINs generated from previously executed queries. Finally, the user may opt to print the report to the Screen or Printer.

The user may opt to print the following reports: Label Data, Pointer Data, References Data, Nameplate Data, Specifications Data, Safety Data, All Folders Data. Each report provides data from the respective folder. Report Examples are included in the following sections.
### 3.2.5.1 LABEL DATA

**9/25/96**

<table>
<thead>
<tr>
<th>EIN</th>
<th>Noun Name 1</th>
<th>Noun Name 2</th>
<th>Component Function</th>
<th>Status</th>
<th>Safety Class</th>
<th>Env Crit.</th>
<th>QA</th>
<th>OSR</th>
<th>Label Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZZZ-ALC-200 CAUSTIC PUMP</td>
<td>10</td>
<td>GENERIC (D10)</td>
<td>ECN</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>ZZZ-ALC-200 COND</td>
<td>11</td>
<td>GENERIC (D10)</td>
<td>001</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>ZZZ-ALC-200 FLUSH TK</td>
<td>12</td>
<td>GENERIC (D10)</td>
<td>111</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

Master Equipment List (Label Info)
### 3.2.5.2 POINTER DATA

**Master Equipment List (Pointer Info)**

<table>
<thead>
<tr>
<th>Farm/Facility: AN FARM</th>
<th>EIN: CHEMB-VFD-100</th>
<th>Component Function: VERTICAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>VFD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Farm/Facility: AN FARM</th>
<th>EIN: CHEMB-VFD-100</th>
<th>Component Function: VERTICAL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Farm/Facility: AN FARM</th>
<th>EIN: CHEMB-VFD-100</th>
<th>Component Function: VERTICAL</th>
</tr>
</thead>
</table>
### 3.2.5.3 REFERENCE DATA

**Master Equipment List (Reference Info)**

<table>
<thead>
<tr>
<th>Farm\Facility: AN FARM</th>
<th>EIN: CHEMB-VFD-100</th>
<th>Component Function: VERTICAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun Name: CAUSTIC XFER PMP SPEED CONT</td>
<td>Status: 006</td>
<td>Safety Class: 3</td>
</tr>
<tr>
<td>OSR: N</td>
<td>Environmental: Y</td>
<td>QA: Y</td>
</tr>
<tr>
<td>Sheet Rev Zone</td>
<td>XYZ Coordinates</td>
<td>Procedure No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Farm\Facility: AN FARM</th>
<th>EIN: CHEMB-VFD-100</th>
<th>Component Function: VERTICAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun Name: CAUSTIC XFER PMP SPEED CONT</td>
<td>Status: 006</td>
<td>Safety Class: 3</td>
</tr>
<tr>
<td>OSR: N</td>
<td>Environmental: Y</td>
<td>QA: Y</td>
</tr>
<tr>
<td>Sheet Rev Zone</td>
<td>XYZ Coordinates</td>
<td>Procedure No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Farm\Facility: AN FARM</th>
<th>EIN: CHEMB-VFD-100</th>
<th>Component Function: VERTICAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun Name: CAUSTIC XFER PMP SPEED CONT</td>
<td>Status: 006</td>
<td>Safety Class: 3</td>
</tr>
<tr>
<td>OSR: N</td>
<td>Environmental: Y</td>
<td>QA: Y</td>
</tr>
<tr>
<td>Sheet Rev Zone</td>
<td>XYZ Coordinates</td>
<td>Procedure No</td>
</tr>
</tbody>
</table>
### 3.2.5.4 NAMEPLATE DATA

**9/25/96**

**Master Equipment List (Nameplate Info)**

<table>
<thead>
<tr>
<th>Facility</th>
<th>Chemb-VFD-100</th>
<th>Component Function: VERTICAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Farm</td>
<td>CAUSTIC XFER PMP SPEED CONT</td>
<td></td>
</tr>
</tbody>
</table>

**Motor**

<table>
<thead>
<tr>
<th>Input Voltage (Volts):</th>
<th>2</th>
<th>Phases:</th>
<th>2</th>
<th>Front:</th>
<th>10</th>
<th>Bearings No.:</th>
<th>Y</th>
<th>Sealed?:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor Size:</td>
<td>3</td>
<td>Motor Size:</td>
<td>3</td>
<td>Rear:</td>
<td>11</td>
<td>N</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Motor Speed (RPM):</td>
<td>4</td>
<td>Frame Size:</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Duty Start Cycle:</td>
<td>6</td>
<td>Electrical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full Load (Amps):</td>
<td>7</td>
<td>Power Supply Breaker:</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Locked Rotor Code:</td>
<td>8</td>
<td>Breaker Trip Setting (Amps):</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Factor:</td>
<td>9</td>
<td>Fuse Size (Amps):</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Manufacturer Overload Size:</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**3.2.5.5 SPECIFICATION DATA**

9/25/96

Master Equipment List (Specification Info)

**FarmlFacility:** AN FARM  
**EIN:** CHEMB-VFD-100  
**Component Function:** VERTICAL

**Noun Name:** CAUSTIC XFER PMP SPEED CONT  
**Status:** 006  
**Safety Class:** 3  
**OSR:** N  
**Environmental:** Y  
**QA:** Y  
**Critical:** Y

**Design**

**Operating**

<table>
<thead>
<tr>
<th>Design</th>
<th>Operating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure:</td>
<td>Pressure:</td>
</tr>
<tr>
<td>Temperature:</td>
<td>Temperature:</td>
</tr>
<tr>
<td>Input:</td>
<td>Input:</td>
</tr>
<tr>
<td>Output:</td>
<td>Output:</td>
</tr>
<tr>
<td>Set point:</td>
<td>Set point:</td>
</tr>
<tr>
<td>Accuracy:</td>
<td></td>
</tr>
<tr>
<td>Full Load (Amps):</td>
<td></td>
</tr>
<tr>
<td>Locked Rotor Code:</td>
<td></td>
</tr>
<tr>
<td>Service Factor:</td>
<td></td>
</tr>
</tbody>
</table>
### 3.2.5.6 SAFETY DATA

**Master Equipment List (Safety Info)**

**9/25/96**

<table>
<thead>
<tr>
<th>Farm/Facility: AN FARM</th>
<th>EIN: CHEMB-VFD-100</th>
<th>Component Function: VERTICAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun Name: CAUSTIC XFER PMP SPEED CONT</td>
<td>Status: 006</td>
<td>Safety Class: 3</td>
</tr>
<tr>
<td>Safety Level:</td>
<td>OSR: N</td>
<td>Environmental: Y</td>
</tr>
<tr>
<td>QA: Y</td>
<td>Critical: Y</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Farm/Facility: AN FARM</th>
<th>EIN: CHEMB-VFD-100</th>
<th>Component Function: VERTICAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun Name: CAUSTIC XFER PMP SPEED CONT</td>
<td>Status: 006</td>
<td>Safety Class: 3</td>
</tr>
<tr>
<td>Safety Level:</td>
<td>OSR: N</td>
<td>Environmental: Y</td>
</tr>
<tr>
<td>QA: Y</td>
<td>Critical: Y</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Farm/Facility: AN FARM</th>
<th>EIN: CHEMB-VFD-100</th>
<th>Component Function: VERTICAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun Name: CAUSTIC XFER PMP SPEED CONT</td>
<td>Status: 006</td>
<td>Safety Class: 3</td>
</tr>
<tr>
<td>Safety Level:</td>
<td>OSR: N</td>
<td>Environmental: Y</td>
</tr>
<tr>
<td>QA: Y</td>
<td>Critical: Y</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
3.2.6 PRIVILEGES

Every MEL user has a set of pre-defined privileges. This information is stored in the user table (user.db). Each record in this table includes the user's name, Hanford ID, password, and privilege. The privilege field is a string of eight characters, in which each individual character represents access to a different data group. Each character is either a "0" or a "1", where a "0" represents no access, and a "1" allows access. The data groups are represented as follows:

1st Character =
   Pointer data
2nd Character =
   Reference data
3rd Character =
   Name data
4th Character =
   Specifications data
5th Character =
   Safety data
6th Character =
   not used
7th Character =
   Label data
8th Character =
   System Administrator

3.2.7 ACCESS CONTROL

Access to the MEL system and its different functions are controlled according to pre-defined privileges. These privileges are setup and maintained by the system administrator (see the "Privileges" section for more information).

Access for a user depends on the privilege code, which is applied to the MEL system. Users only have write access to those data groups of the application which are represented in their respective privilege code. Any data group in which access is not allowed is automatically set to read-only. The privilege code is checked after user logon, and write access to the different parts of the system are set at that time.
3.2.8 CHANGES PENDING & CHANGE LOG

3.3 LAUNCHER

The MEL system contains an executable file which is used to start or “launch” the main program. This file is needed not only to start the client application, but to handle network connections as well.

Since the MEL is a multiple user database, the data must be centrally located on a common file server. Connections to this file server must occur prior to starting the MEL. Furthermore, these connections need to be closed once the user terminates the MEL. These tasks are all accomplished by the MEL launcher application.

The MEL launcher has settings configured within an initialization (INI) file to launch a specified version of the application. This INI file is communicated to the launcher on the Windows command line. For example, “C:\WINAPPS\MEL\LAUNCH.EXE MELBETA.INI” would tell the launcher to initiate the beta version of the MEL.

After determination of the version, the launcher attempts to make a connection to every file server that the MEL requires. If these connections cannot be made, the user is notified and the session is terminated. Otherwise, the MEL application is initiated. Once the MEL has been started, the launcher waits in the background until the user terminates the session, at this point, the launcher disconnects the MEL network connections and returns the user to the Windows environment. The DAES Launcher Application User Manual can be found in the DAES technical library.

3.4 DISTRIBUTION / INSTALLATION

3.4.1 RELEASE AREAS & VERSIONS

The MEL system has several release areas associated with it. These include Alpha, Beta, Alt1, and Production. They are defined as follows:

Alpha Release Area - This area contains the most recent development work. The source code is untested and contains data which may be discarded at any time. It is for developers' use only, and available only to selected users for testing purposes.
Beta Release Area - This area contains the version of the MEL that is currently in the process of acceptance testing. Source code for this area must be kept under configuration control, with a copy being maintained by the software custodian. Access to this area is available to selected beta testers only.

Alt1 Release Area - This area is a temporary holding area for a tested copy of the MEL application. In this area, the application has been acceptance tested, however, the system may still need data transferred in from other sources, or may need operational testing. Data in this area will not be removed without the system administrators confirmation.

Production Release Area - This area houses the production version of the MEL. This is the area in which the MEL is used by only authorized users and data is strictly maintained. There should be no testing in this area, since the data is quality affecting.

Besides having separate release areas, each area may have a different version of the MEL. For example the Production Release Area might contain Version 1.0. However, the Beta Release Area might have Version 2.0, and the Alpha Release Area might be on Version 3.0. Nevertheless, a user may always find out what release and version they are currently running by looking in the Help / About section of the application.

3.4.2 INSTALLATION

The Installation process for the MEL system involves several functions. First, the installation program or "Installer" is a third-party software package called "WINStall" developed by OnDemand software. This program, which is the Hanford Site standard for network installations, places files from a network file server to a location on the client machine. These files and subdirectories include:

- MEL_PHS1.EXE - MEL executable file
- LAUNCH.EXE - MEL Launcher
- MEL*.INI - MEL Initialization file
- MEL1*.VBR - MEL Version File
- MEL.ICO - MEL Icon
- LOC_DATA subdirectory - Subdirectory for temporary files, reports, local tables
- PREFS subdirectory - Subdirectory for user preferences
All files are located on the client machine in the directory C:\WINAPPS\MEL\RELEASE NAME, where release name is the name of the installed release. The MEL1*.VER file depends on the installed release. The * character is replaced with a representation of the release type.

Each release contains a *.ver file which is used for version identification. If a user has not installed the most recent version of a release, they are prompted to install the new version before continuing. The comparison is performed by using the date and time stamp of the *.ver files, with the most recent one being available on the network fileserv er.

3.4.3 USER COMPUTER RESOURCE REQUIREMENTS

The following configuration is recommended for user computers:

- 486 or higher IBM-Compatible Personal Computer (PC).
- A hard disk drive configured as "C:"
- At least 8 megabytes of RAM, 16 megabytes is recommended.
- At least 30 megabytes free disk space on "C:"
- Hanford End System Operating Environment (ESOE).
4.0 REFERENCES

1) WHC-CM-6-1, *Standard Engineering Practices.*
   EP-5.8, Rev. 3, "Engineering Document Content Guidelines",
   Exhibit 12, February 7, 1992.

2) WHC-CM-4-2, *Quality Assurance Manual.*
   QR-19.0, Rev. 0, "Software Quality Assurance Requirements,
   December 6, 1991.


