OCT 01 1996 20
Sta. 37

S

ENGINEERING DATA TRANSMITTAL

Page 1 of 1
1. EDT 618229

2. To: (Receiving Organization) Distribution	3. From: (Originating Organization) TWRS SE Risk Management & Modeling	4. Related EDT No.: N/A
5. Proj./Prog./Dept./Div.: TWRS	6. Design Authority/ Design Agent/Cog. Engr.: D. D. Frank	7. Purchase Order No.:
8. Originator Remarks: Document transmitted for release.		9. Equip./Component No.:
		10. System/Bldg./Facility:
11. Receiver Remarks: 11A. Design Baseline Document? [] Yes [X] No		12. Major Assm. Dwg. No.:
		13. Permit/Permit Application No.:
		14. Required Response Date:

15. DATA TRANSMITTED

(A) Item No.	(B) Document/Drawing No.	(C) Sheet No.	(D) Rev. No.	(E) Title or Description of Data Transmitted	(F) Approval Designator	(G) Reason for Transmittal	(H) Originator Disposition	(I) Receiver Disposition
1	WHC-SD-WM-SDR-012		0	Requirements Management System Browser Software Design Description	N/A	2	1	

16. KEY

Approval Designator (F)	Reason for Transmittal (G)		Disposition (H) & (I)	
E, S, Q, D or N/A (see WHC-CM-3-5, Sec.12.7)	1. Approval 2. Release 3. Information	4. Review 5. Post-Review 6. Dist. (Receipt Acknow. Required)	1. Approved 2. Approved w/comment 3. Disapproved w/comment	4. Reviewed no/comment 5. Reviewed w/comment 6. Receipt acknowledged

17. SIGNATURE/DISTRIBUTION
(See Approval Designator for required signatures)

(G) Reason	(H) Disp.	(J) Name (K) Signature (L) Date (M) MSIN	(G) Reason	(H) Disp.	(J) Name (K) Signature (L) Date (M) MSIN
		Design Authority	1	1	Technical Lead B. C. Gneiting B. C. Gneiting 9/30/96
		Design Agent			
1	1	Cog.Eng. D. D. Frank Duane D. Frank 9/30/96			
1	1	Cog. Mgr. N. G. Awadalla N G Awadalla 9/30/96			
		QA			
		Safety			
		Env.			

18. Duane D. Frank 9/30/96 Signature of EDT Originator Date	19. Authorized Representative Date for Receiving Organization	20. N G Awadalla 9/30/96 Design Authority/ Cognizant Manager Date	21. DOE APPROVAL (if required) Ctrl. No. [] Approved [] Approved w/comments [] Disapproved w/comments

S

WHC-SD-WM-SDR-012, Rev. 0

Requirements Management System Browser Software Design Description

D. D. Frank
Westinghouse Hanford Company, Richland, WA 99352
U.S. Department of Energy Contract DE-AC06-87RL10930

EDT/ECN: 618229	UC: 905
Org Code: 74400	Charge Code: D1M1M
B&R Code: EW3120075	Total Pages: 87

Key Words: Systems Engineering; Requirements Management; RMACS; TWRS; Browser; Parser; SE Tools

Abstract: The purpose of this document is to provide an "as-built" design description for the Requirements Management System Browser (RMSB) application. The Graphical User Interface (GUI) and database structure design are described for the RMSB application, referred to as the "Browser." The RMSB application provides an easy to use PC-based interface to browse systems engineering data stored and managed in a UNIX software application. The system engineering data include functions, requirements, and architectures that make up the Tank Waste Remediation System (TWRS) technical baseline.

Trademarks: Access, Visual Basic, Windows, and SQL are trademarks of Microsoft Corporation, Redmond, WA.

Unix is a trademark of IBM Corporation.

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.

V. L. Birkland 9/30/96
Release Approval Date

Release Stamp

Approved for Public Release

CONTENTS

LIST OF TABLES

1.0 INTRODUCTION

Systems Engineering (SE) information for the Tank Waste Remediation System (TWRS) is currently stored and managed with the RDD-100[1] software application. The RDD-100 application manages data supporting the systems engineering process including functions, requirements, and architecture definitions for specifications. The RDD-100 application runs on a UNIX workstation, costs over $30K per license, and is not an open system. Therefore, a personal-computer (PC)-based system was developed to increase the ability of engineers and managers on-site to access and interact with the information stored in the RDD-100 system. To provide a more open and cost-effective access to the RDD-100 data for a wide range of users, a relational database with a graphical user interface (GUI) was developed for PC users running Microsoft Windows. This software application is titled, "Requirements Management System Browser" (RMSB) or "Browser".

The Browser data structure and interface was designed to closely mimic the RDD-100 data structure and terminology. The database tables were designed to match each element type in RDD-100 and the relationships between elements were recorded in a table named after the RDD-100 relationship. To transfer data from RDD-100 to the Browser, a PC-based GUI was developed titled - RMSB Data Loader (RMSB-DL) or the "Parser". The Parser reads and parses an RDD-100.rdt file and loads the RMSB database.

[1] RDD-100 is a registered trademark of Ascent Logic Corporation

This page intentionally left blank.

2.0 PURPOSE AND SCOPE

The purpose of this document is to provide an "as-built" design description for the RMSB application commonly referred to as the "Browser". This includes a specification of the database structures and the GUI. The description contains sufficient detail to describe the processes used by the software. This document does not include the Parser, the RMSB installation software, or the RMSB updating software. Installation and update of the RMSB application is covered in the documentation on the TWRS SE Tools generic installation program and generic automatic update software. Revisions of the software are covered by the draft TWRS Systems Engineering Software Configuration Management Plan (WHC-SD-WM-CSCM-034) for the TWRS SE Tools.

This page intentionally left blank.

3.0 ABBREVIATIONS AND ACRONYMS

The table below identifies and defines the abbreviations and acronyms used within this document.

Acronym	Definition
ERA	Element-Relation-Attribute, the database model used
GUI	Graphical User Interface
RMACS	Requirements Management and Assured Compliance System
RMSB	Requirements Management System Browser, the Browser
RMSB-CMU	RMSB Change Management Utility
RMSB-DL	RMSB Data Loader, the Parser
SE	Systems Engineering
TWRS	Tank Waste Remediation System

This page intentionally left blank.

4.0 DESIGN DESCRIPTION

This section of the document describes the design of the Browser. The section on database structure is also applicable to the Parser. Section 4.1 includes background information on Visual Basic[2], which is necessary to understand most aspects of the design description. The database structures used by the software are described in Section 4.2. Section 4.3 describes the design of the various forms (.FRM files) that make up the software application, including objects and events associated with each form, and navigation to other forms. Section 4.4 overviews the various Visual Basic subroutine files (.BAS files) to be developed, and describes some of the details of more complex routines and functions. Sections 4.3 and 4.4 contains "pseudo-code" to identify the procedural or functional steps for the various event procedures and supporting routines.

4.1 BACKGROUND INFORMATION

The development of the software was done with Visual Basic Professional edition, version 3.0. Visual Basic applications are built starting with window forms that have controls (buttons, tables, menus) placed on them. Each of the forms and controls have event procedures that can be used for input, output, or processing. This leads to a program structure that is more object-oriented than procedural or functional. Because of the object-oriented paradigm, the suggested format for the Software Design Description, which is found in WHC-CM-3-10, *Software Practices*, Appendix H, is not followed. This document is organized around the forms/windows (.FRM files) that are used to develop the GUI. Support routines and functions are organized into Basic modules (.BAS files) that can be called from any form.

Visual Basic also does not have parameter passing mechanisms between window forms. For this reason, all of the table pointers and many of the variables and constants in the software are defined globally. All global declarations for variables and constants are defined in one file, **GLOBALS.BAS**, which is automatically loaded when the application is run. The Main Window (**MAINWIND.FRM**) is also automatically loaded as the first window displayed by the software.

Table 1 identifies the various event procedures that are automatically created when windows and controls are created. Initially, the source code that is defined for each of these event procedures is empty. The software developer enters the appropriate Visual Basic code to be executed for the desired events. The table also identifies which object types (forms, controls, etc.) use the event procedures. There are additional event procedures for each object type, however, Table 1 identifies those events that are most commonly used in the software.

[2] Visual Basic is a registered trademark of Microsoft Corporation.

Table 1. Event Procedure Definitions.

Event Procedure	Object Types	Description
Load	Forms	This event is triggered the first time a form is displayed
Resize	Forms	This event is triggered when a form is resized either by the user or within code
Unload	Forms	This event is triggered when a form is closed
Click	Forms, Most Control Types	This event is triggered when a form or control is clicked by a mouse button press (usually only left mouse button)
DblClick	Forms, Lists, Tables, Labels, Pictures	This event is triggered when a form or control is double-clicked with the mouse (usually only left mouse button)
Validate	Data Aware Controls	This event is triggered during the validation phase on a database record associated with a data aware control
Reposition	Data Aware Controls	This event is triggered when a reposition on the database table associated with a data aware control occurs
Change	Text Boxes, List Boxes	This event occurs when a change to the data within a control is made, typically by the user entering a new value.

4.2 DATABASE STRUCTURE DESIGN DESCRIPTION

This section identifies the design structure for the database tables used within the software. For this discussion, the database structures are organized into three groups: element tables, relation tables, and change tables. All database tables are in Microsoft Access format. This format is used because Visual Basic provides a direct interface to the Access "jet engine" through built-in functions and routines. The database structure is the same for both the Browser and the Parser.

4.2.1 Element Tables

For each element type currently used in RDD-100, one element table is created for the Browser and the Parser. Each element table has fields that correspond to the attributes of the RDD-100 elements. Note that the elements, attributes, and relations used within RDD-100 will likely change during the life of the software. Table 2 identifies the element tables that are used within the Browser, and the corresponding RDD-100 and RMACS nomenclature.

Table 2. Element Tables.

Element Table Name	RDD Element Name	RMACS Element Name
Architecture	Component	Architecture
Category	Category	Category
Decision	Decision	Decision
Function	TimeFunction	Function
Input/Output	TimeItem	Input/Output
Issue	CriticalIssue	Issue
Note	Comment	Note
Organization	Organization	Organization
Required Analysis	RequiredAnalysis	Required Analysis
Requirement	SystemRequirement	Requirement
Source	Source	Source
Engineer	Engineer	Engineer
Interface	Interface	Interface
ItemLink	ItemLink	ItemLink

Each of the element types listed in Table 2 has a set of attributes associated with it. These attributes correspond to the fields of each element record in the database. A number of the attributes are used by all element types and are known as the "common" or "core" attributes. Table 3 identifies all of the attributes to be used in the software, both the common and element-specific attributes.

Table 3. Element Attributes.

Attribute	Data Type	Size, Bytes	Description
Name	String	250	The unique name of the element record, which cannot be empty. Common to all element types.
Number	String	50	Number, sometimes hierarchical. Common to all element types.
Creator	String	40	Creator of the record in RDD-100. Common to all element types.
CreationDate	Date	8	Date the record was created in RDD-100. Common to all element types.
ModificationDate	Date	8	Date the record was last modified in RDD-100. Common to all element types.
ModificationTime	Time	8	Time the record was last modified in RDD-100. Common to all element types.
Description	Memo	32K	The description associated with the element record. Common to all element types.
Color	Integer	4	Color of the element, if a change has occurred. Common to all element types.
Status	List 6	25	The status of a Decision record.
Choice	Memo	32K	The choice, or description of how a choice was made for a Decision.
Alternatives	Memo	32K	The alternatives considered for a Decision.
IssueType	List 1	20	The type of Issue record.
ActualDate	Date	8	The actual date of the resolution of an issue.
Priority	List 2	15	The priority of an issue.
ScheduledDate	Date	8	The scheduled date for the resolution of an issue.
Status	List 4	10	The status of a Requirement.
RequirementType	List 5	15	The type of Requirement.
ApprovedBy	String	100	Who approved the Requirement.
SourceType	List 3	25	The document type for a Source.

As noted in Table 3, some attributes are of an enumerated type (i.e., there is a finite list of values that are allowed for the attribute). The lists referenced in Table 3 are shown below. Note that the value "[nil]" means that the value contains the empty string.

List 1: IssueType List
[nil]
Issue
Required Analysis

List 2: Priority List
[nil]
A (Very High)
B (High)
C (Medium)
D (Low)

List 3: SourceType List
[nil]
Change Request
Meeting Minutes
Originating Requirements
Other
Project Memo
Standard
Trade-off Study Report

List 4: Requirement Status List
[nil]
Approved
Changed
Pending
Rejected

List 5: RequirementType List
[nil]
Design
Information

List 6: Decision Status List
[nil]
Open
Resolved
Enabling Assumption

The Element tables also have two indexes. The first index, "Name Index" is an index on the "Name" attribute of a table. The "Number Index" is an index on the "Number" and "Name" attributes of a table. These indexes are used for faster searching and sorting of the element tables.

4.2.2 Relation Tables

Within RDD-100, elements records are "linked" to each other using various relations. For example, a Source element *documents* a Requirement or Decision element. All relations are bi-directional, so that every relation has a converse relation. For example, the converse of the *documents* relation is the relation *documented_by*. Table 4 identifies all of the relation pairs currently used in RDD-100 that are implemented in the Browser and Parser software.

Table 4. Relation Tables.

Relation Table and RDD Name	Description
analyzed_by / analyzes	A RequiredAnalysis analyzes a Decision or Issue
annotated_by / annotates	A Note annotates any other element type
built_from / built_in	Defines a hierarchy of Architecture elements
categorized_by / categorizes	A Category categorizes any other element type
documented_by / documents	A Source documents any other element type
incorporated_by / incorporates	Defines a hierarchy of Requirement elements
input_to / inputs	An Input/Output is input_to a Function
output_from / outputs	An Input/Output is output_from a Function
performed by / performs	An Architecture performs a Function
primary_is / primary_for	An Organization is primary_for any element type
secondary_is / secondary_for	An Organization is secondary_for any element type
traced_from / traces_to	A Decision, Issue, or Requirement traces to any element type
owned by / owns	An Engineer owns any element type
contains / is contained by	An Interface contains an ItemLink
carries / carried by	An ItemLink carries an Input/Output

For each of the relations listed in Table 4, a relation table is created with the name of the relation. All relation tables have the same fields, as identified below in Table 5. All relation tables have two indexes, the first being the "Source Index" on the source element type and name, and a "Related Index" on the related element type and name.

Table 5. Relation Table Attributes.

Attribute	Data Type	Size	Description
Source Type	Integer	4	The element type of the source element of the relation.
Source Element	String	250	The name of the source element.
Source Number	String	50	The number of the source element.
Related Type	Integer	4	The element type of the related, or target, element of the relation.
Related Element	String	250	The name of the related element.
Related Number	String	50	The number of the related element.
Color	Integer	4	The color for a changed relation.

4.2.3 Change Tables

To generate "delta" files for RDD-100, the Browser software retains information on all of the changes made by the user. Changes are organized into three categories: element changes, attribute changes, and relation changes. Element changes include the creation, renaming, or deleting of element records. Attribute changes are changes to the attributes of an existing element. Relation changes occur when relation links are created or deleted between elements.

Table 6 identifies the attributes for the ElementChanges table. The ElementChanges table has the following indexes:

ChangeIndex:	ChangeType field
DateTimeIndex:	ModificationDate and ModificationTime fields
NewNameIndex:	NewElementName field
PrimaryKey:	ElementType and ElementName fields
StatusIndex:	Status field

Table 6. ElementChanges Table Attributes.

Attribute	Data Type	Size	Description
ElementType	Integer	4	The element type of the change.
ElementName	String	250	The name of the changed element.
ChangeType	String	6	The type of change to the element (Create, Rename, or Delete).
NewElementName	String	250	The new element name if a Rename.
ModificationDate	Date	8	The date the change was made.
ModificationTime	Date	8	The time the change was made.
ModifiedBy	String	50	Who made the change.
Status	String	20	The status of the change.
Justification	Memo	32K	The justification or reason for the change.

The AttributeChanges table is used to retain information associated with data changes to the attributes of elements. The fields of the AttributeChanges table are shown below in Table 7. The AttributeChanges table has the three indexes given below.

DateTimeIndex:	ModificationDate and ModificationTime fields
PrimaryKey:	ElementType and ElementName fields
StatusIndex:	Status field

Table 7. AttributeChanges Table Attributes.

Attribute	Data Type	Size	Description
ElementType	Integer	4	The element type of the change.
ElementName	String	250	The name of the changed element.
Changes	Integer	4	The attribute modified.
NewValue	Memo	32K	The new value for the attribute.
ModificationDate	Date	8	The date the change was made.
ModificationTime	Date	8	The time the change was made.
ModifiedBy	String	50	Who made the change.
Status	String	20	The status of the change.
Justification	Memo	32K	The justification or reason for the change.

The RelationChanges table identifies all relations that have been inserted or removed between element records. Table 8 identifies the attributes of the RelationChanges table. Below are the indexes used for this table.

ChangeIndex:	ChangeType field
DateTimeIndex:	ModificationDate and ModificationTime fields
SourceIndex:	SourceType and SourceName fields
StatusIndex:	Status field
TargetIndex:	TargetType and TargetName fields

Table 8. RelationChanges Table Attributes.

Attribute	Data Type	Size	Description
RelationTable	Integer	4	Index to the relation table changed.
ChangeType	String	6	The type of change made to the relation (Insert or Remove).
SourceType	Integer	4	The element type of the source element changed.
SourceName	String	250	The name of the source element changed.
TargetType	Integer	4	The element type of the target (related) element changed.
TargetName	String	250	The name of the target (related) element changed.
ModificationDate	Date	8	The date the change was made.
ModificationTime	Date	8	The time the change was made.
ModifiedBy	String	50	Who made the change.
Status	String	20	The status of the change.
Justification	Memo	32K	The justification or reason for the change.

4.3 FORM (WINDOW) DESIGN DESCRIPTIONS

This section discusses the design of each of the windows (.FRM files) associated with the software. Note that this does not include message boxes, which are provided by Visual Basic, file open, and save dialogue boxes, or other simple windows such as the **Welcome** window. Pseudo-code for event procedures for forms and controls are described in this section as well. Note that many of the controls are named by the function they perform rather than the actual implemented name of the control. For example, a Print button may have an actual name of IEPrint. The actual name of the control is listed between square brackets, for example [IEPrint]. Routines or functions called by the various events are also listed between square brackets. These routine names and their arguments can be found in Appendix A.

The following forms are not discussed because they are fairly simple and have very little source code associated with them:

Welcome window (**WELCOME.FRM**)
New User window (**NEWUSER.FRM**)
Element selection window (**FRMELSEL.FRM**)
Expanded text value window (**FRMEXPAN.FRM**)
List relations window (**LISTRELA.FRM**)
Get Reason/Justification window (**GETREASO.FRM**)
Get New Page Number window (**GETNEWPA.FRM**)
Print Destination window (**PRINTDES.FRM**)

4.3.1 MainWindow (MAINWIND.FRM)

The form **MainWindow** is the main window of the software. This is also the window that is automatically displayed when the program begins execution. This is a resizable window that is dominated by a table of elements of a currently selected type. Most of the command menus are on the MainWindow. Below is a brief description of each of the items for the window, and any non-trivial event procedures associated with the items.

Menus: The Browser **MainWindow** has the following menus:

File Menu
- Open Database: Open any Browser database file
- Repair Database: Attempt to repair current database file
- Preferences: Display the **Preferences** window
- Export: Menu for exporting (has submenus)
  - Changes to RDD-100: Export changes in .RDT format
  - Changes to RMSB: Export changes to RMSB-CMU
  - Changes to Text: Export changes as ASCII text
  - F&R Matrix: Matrix of functions and requirements
  - Record Count Report: Number of records for each table
- Erase History: Erase all change history
- Exit: Exit from the Software

Edit Menu (Used for all windows with **Edit** menu)
Cut: Cut information to clipboard
Copy: Copy information to clipboard
Append: Copy and append information to clipboard
Paste: Paste information from clipboard
Select All: Select all text in current control
Display Clipboard: Display the text contents of the clipboard

Records Menu
Add New: Add a new record of the selected type
Delete: Delete a set of selected records
View Changes: View change information (display **Changes** window)

Query Menu:
Single Table: Displays single table search window **frmSingleQuery**
Two Table Search: Displays two table search window **MultiSearch**
Multiple Tables: Displays **SearchWindow**

Help Menu
Using Help: Help on using the Microsoft *WinHelp*[3] software
Contents: Table of Contents
General Info: General information (overview)
Edit Menu: Information on the **Edit** menu
Main Window: Information on the **MainWindow**
Menus: Information on the **MainWindow** menus
Buttons: Information on the **Mainwindow** buttons
Database Structure: Information on the database structure
This Element: Information on the current element type
About: Version information for the Browser software
Data Version: Version information on the current database file
Change History: Browser software change history information

Menu Event Procedures: The following are the events associated with each of the menus of the **MainWindow** as listed above.

File Menu, Open Database [mnuFileOpen]
Get file name to open from the user
Open new file name [ChangeDatabase]

File Menu, Repair Database [mnuRepair]
Query the user if they really want to repair the database
Close the current database [RepairTheDatabase]
Use the Visual Basic RepairDatabase command [RepairTheDatabase]
Re-open the current database [RepairTheDatabase]

File Menu, Preferences [mnuPrefs]
Show the preferences window **PrefWindow**
File Menu, Export - Changes to RDD-100 [mnuSupRDT]
Get output file name from the user
Export .RDT file as a subordinate [ToSuperiorRDT]

[3] WinHelp is a registered trademark of Microsoft Corporation.

File Menu, Export - Changes to RMSB [mnuExportRMSB]
Export change tables to empty database [ExportRMSB]

File Menu, Export - Changes to Text [mnuFECtoText]
Get output file name from the user
Export changes as ASCII Text [ToASCIIText]

File Menu, Export - F&R Matrix [mnuFRMatrix]
Get output file name from the user
Export Functions and Requirements Matrix [ExportMatrix]

File Menu, Export - Record Count Report [mnuDBReport]
Get output file name from the user
Print record counts to file [DBReport]

File Menu, Erase History [mnuMainErase]
Query the user if they really want to erase all history
Erase all change records in the database
Set the "color" of all elements to unchanged (black)
Refill the **MainWindow** table [RefillMain]

File Menu, Exit [mnuExit]
Close the currently open database file [CloseTheDatabase]
Save the user's preferences [SaveUserInfo]
Exit the software

Edit Menu, Cut [mnuMainCut]
Cut the selected text to the Windows clipboard [CutToClipboard]

Edit Menu, Copy [mnuMainCopy]
Copy the selected text to the Windows clipboard [CopyToClipboard]

Edit Menu, Append [mnuMWAppend]
Append the selected text to the Windows clipboard [AppendToClipboard]

Edit Menu, Paste [mnuMainPaste]
Paste the selected text from the Windows clipboard [PasteFromClipboard]

Edit Menu, Select All [mnuMainSelectAll]
Select all text in the current control [SelectAllText]

Edit Menu, Display Clipboard [mnuMWDC]
Display current contents of the Windows clipboard [ShowClipboard]

Records Menu, Add New [mnuMainNew]
Set NewRecord flag to true and display empty record [ShowElement]

Records Menu, Delete [mnuMainDelete]
Query user if want to be queried for more than 1 record deleted
Delete all records selected (query user where applicable)
Update the **MainWindow** table

Records Menu, View Changes [mnuShowChanges]
Display the **ChangeWindow**

Query Menu, Single Table [mnuMWQSingle]
Get element type for search from user
Display single table search window **frmSingleQuery**

Query Menu, Two Table Search [mnuTwoTable]
Display two table search window **MultiSearch**

Query Menu, Multiple Tables [mnuMWQMultiple]
Get element type for search from user
Display multiple table search window **SearchWindow**

Help Menu, Using Help to This Element, and Change History commands
Use *WinHelp* with the appropriate topic number [Get_Help]

Help Menu, About [mnuAbout]
Display the **Welcome** window

Help Menu, Data Version [mnuDataVersion]
Display message box with current data version information

Controls: The following are the controls associated with the **MainWindow.**

ElementList List Box: Select the element type to display
ElementGrid Table: Display the Name and Number fields in a table
SortButton Button: Select the sorting order for the table
Relations Button [RelationsB]: Display the **RelWindow** for relations
OldViews Button [Command1]: Display the **OldViews** window
Print Button [IEPrint]: Print current table
Exit Button [QuitButton]: Exit the software
FirstPage Button [Mh3dButn1(0)]: Move to first Virtual Page
PreviousPage Button [Mh3dButn1(1)]: Move to the previous Virtual Page
NextPage Button [Mh3dButn1(2)]: Move to the next Virtual Page
LastPage Button [Mh3dButn1(3)]: Move to the last Virtual Page
PageLabel Text Box: Display current Virtual Page

Form and Control Event Procedures: The following are the events associated with the **MainWindow** form and the controls listed above.

Sub Form_Load - Triggered when the form is initially loaded
Show **Welcome** window
Initialize program parameters [InitProgram]

Sub Form_Resize - Triggered when the form is resized by user or software
Check for minimum window width
Resize **ElementGrid** table
Align control buttons at bottom of window

Sub Form_LinkExecute - Triggered on a Dynamic Data Exchange (DDE) "Execute" command
Parse the DDE Link command and handle accordingly (show, quit, or error)

```
Sub ElementList_Click - User selects new item in the ElementList
     Set new Current Element Type
     Refill ElementGrid with records from new element  [UpdateMainGrid]

Sub ElementGrid_DblClick - User double-clicks on a table row/record
     Determine record clicked
     If record deleted, then display warning and ignore event
     Otherwise, display the selected record  [ShowElement]

Sub SortButton_Click - User clicks on the "Sort by Name/Number" button
     Sort and redisplay the ElementGrid by Name or Number  [UpdateMainGrid]

Sub RelationsB_Click - User clicks on the "Relations" button
     Display the relations window RelWindow

Sub OldViews_Click  [Command1] - User clicks on the "Old Views" button
     Display the OldViews window

Sub IEPrint_Click - User clicks on the "Print" button
     Determine format to print (labeled text or two columns)
     Print Number and Name fields for all rows

Sub QuitButton_Click - User clicks on the "Exit" button
     Close current database file  [CloseTheDatabase]
     Save user preferences  [SaveUserInfo]
     Exit software

Sub PageLabel_DblClick - User double-clicks on the page label text box
     Get virtual page number from user
     Set new top virtual page
     Refill main window table  [UpdateMainGrid]

Sub FirstPage_Click  [Mh3dButn1(0)] - User clicks on the "First Page" button
     Set top virtual page to 1
     Refill main window table  [UpdateMainGrid]

Sub PreviousPage_Click  [Mh3dButn1(1)] - User clicks "Previous Page" button
     Decrement top virtual page by 1
     Refill main window table  [UpdateMainGrid]

Sub NextPage_Click  [Mh3dButn1(2)] - User clicks "Next Page" button
     Increment top virtual page by 1
     Refill main window table  [UpdateMainGrid]

Sub LastPage_Click  [Mh3dButn1(3)] - User clicks "Last Page" button
     Set top virtual page to last page
     Refill main window table  [UpdateMainGrid]
```

4.3.2 PrefWindow (PREFWIND.FRM)

The **PrefWindow** is used to set user preferences for the various software options. The preferences window is displayed when a user clicks on the "Preferences" command of the "File" menu from the **MainWindow**.

Menus: The **PrefWindow** form only the "Help" menu.

Menu Event Procedures: Each of the commands of the "Help" menu will display the appropriate on-line help topic.

Controls: The controls of the preferences allow the user to set various program options. The controls used by this window are:

DataView List [List1]:Select viewing mechanism
SortOrder List [List2]:Set sorting order for current element type
NamesView List [List3]:Select nomenclature used for browser
ElementCreate Check Box [Check1]:Query reason on element create
ElementDelete Check Box [Check2]:Query reason on element delete
ElementModify Check Box [Check3]:Query reason on element modified
RelationCreate Check Box [Check4]:Query reason on relation create
RelationDelete Check Box [Check5]:Query reason on relation delete
LastSelected Option [Option1]:Always show last selected relation
OnlyWhenAsked Option [Option2]:Show only when asked (reset to none)
VMode Check Box: Turns on and off the virtual mode feature
VSize Text Box: Sets the virtual page size for virtual mode
Cancel Button [Command1]: Exit the window without saving changes
OK Button [Command2]: Exit the window and save changes

Form and Control Event Procedures: The following are events associated with the preferences window form and the controls listed above.

Sub Form_Load - Triggered when window is first displayed
Fill in the **Dataview**, **SortOrder**, and **NamesView** lists
Set check box values and virtual mode values

Sub Command1_Click - User clicks the "Cancel" button
Unload the window

Sub Command2_Click - User clicks the "OK" button
Update the element detail viewing parameters if changed
Update the nomenclature if changed
Update the sorting order of the main window if changed [UpdateMainGrid]
Set reason/justification query values
Update the virtual mode settings if changed
Unload the window

Sub List1_Click - User clicks on the data view list
If "Custom..." was selected, display the **CustomViews** window

Sub VMode_Click - User clicks on the virtual mode check box
Enable (Disable) VSize text box

4.3.3 ItemEdit Window (ITEMEDIT.FRM)

The **ItemEdit** window, often referred to as the "Record Details" window, displays the attributes and relations of a single element record. This window also allows the user to modify the attributes, create and delete relations, delete the element record, or add a new element record. The window is resizable, and which attributes are displayed is defined from the preferences window, **PrefWind**, and the **CustomViews** window. The window is also built "on-the-fly" by cloning and placing the appropriate controls so that the source code for the window does not have to be modified when the database structure changes. The window is displayed whenever an element record instance is double-clicked in any of the tables in the other windows, such as the **MainWindow** or one of the search windows.

<u>Menus</u>: The **ItemEdit** window has the following menus:

```
    Edit Menu:  Same as MainWindow Edit menu
    Records Menu
        Add New:  Add a new element record
        Delete:  Delete current element record
        Add Relation(s):  Add relations to the element
        Delete Relation(s): Delete selected relations from element
        Inherit Relations:  Inherit relations from another element
    Views Menu
        Old Views:  Display the OldViews window
        Go to Previous View:  Go to the previously displayed record
        Change History:  Display the change history window HistoryWindow
    Query Menu:  Same as MainWindow Query menu
    Reports Menu  (Will be expanded in the future)
        Architecture:  Produce an architecture report
    Help Menu:  Similar to MainWindow help menu
```

<u>Menu Event Procedures</u>: The following are the event procedures associated with each of the menus for the **ItemEdit** window as listed above.

Edit Menu: All commands same as **MainWindow** Edit menu

```
Records Menu, Add New  [mnuNewRecord]
    Verify and save the current record information
    Create a new blank record for data entry

Records Menu, Delete [mnuRecordDelete]
    Query user to delete record
    Delete the record  [DeleteElement]
    Fill in relations grid with next element  [FillElement]

Records Menu, Add Relation(s)  [mnuAddRelation]
    Verify and save the current record information  [CanSave, UpdateElement]
    Display the AddRelations window
```

Records Menu, Delete Relation(s) [mnuDeleteRelation]

```
Check validity of rows to delete
Query user if they want to be queried
For each relation:
    Determine relation link to delete
    Delete the link [RemoveRelation]
Update RelationsGrid
```

Records Menu, Inherit Relations [mnuIEInherit]

```
Verify and save the current record information [CanSave, UpdateElement]
Display the InheritRelations window
```

Views Menu, OldViews [mnuIEOldViews]

```
Verify and save the current record information [CanSave, UpdateElement]
Display the OldViews window
```

Views Menu, Go to Previous View [mnuIEGoto]

```
Verify and save the current record information [CanSave, UpdateElement]
Pop the previous view off the view stack
Display the new record [ShowElement, FillElement]
```

Views Menu, Change History [mnuIEVCH]

```
Verify and save the current record information [CanSave, UpdateElement]
Display the change history window [ShowChangeHistory]
```

Query Menu: Same as **MainWindow** Query window

Reports Menu, Architecture [mnuArchRept]

```
Clear the previous architecture report
Determine all related elements to the current architecture
Display the report results (uses Crystal Reports)
```

Help Menu: Display the appropriate On-Line help topic [Get_Help]

Controls: The following are the controls associated with the **ItemEdit** window. Note that some of the controls are indexed, so that they can be cloned as needed to build the window on-the-fly.

AttributeLabel(Index) Label: Attribute label for any/all attributes
AttributeList(Index) List Box: Attribute list for enumerated attributes
AttributeMemo(Index) Text Box: Scrollable, multi-line attribute value
AttributeValue(Index) Text Box: Single line attribute value box
Related Element List [REList]: Element type to show in **Relations Grid**
RelationsGrid Table: Displays the selected related element records
Previous View Button [IECancel]: Moves to the previously viewed record
First Record Button [IEFirst]: Move to the first record in the table
Previous Record Button [IEPrevious]: Move to the previous record
Next Record Button [IENext]: Move to the next record in the table
Last Record Button [IELast]: Move to the last record in the table
Old Views Button [IEOldViews]: Display the **OldViews** window
Print Button [IEPrint]: Print current record information
Return Button [IEReturn]: Exits the window

Form and Control Event Procedures: The following are the events associated with the **ItemEdit** form and the controls listed above.

Sub Form_Load - Triggered when window is loaded
    Turn on/off "Records" menu based on ReadOnly status

Sub AttributeLabel_DblClick(Index) - When an attribute label is double-clicked
    Display the On-Line help topic for the selected attribute [Get_Help]

Sub REList_Click - When the related element list changes
    Refill the **RelationsGrid** with the related elements [FillIERelGrid]

Sub RelationsGrid_DblClick - When user double-clicks on related element
    Determine validity of element selected
    Verify and save the current record information [CanSave, UpdateElement]
    Display the new element record [ShowElement or FillElement]

Sub IEFirst_Click - When user clicks on the first record button
    Verify and save the current record information [CanSave, UpdateElement]
    Move to the first record in the table
    Refill the **ItemEdit** window [FillElement]

Sub IEPrevious_Click - When user clicks on the previous record button
    Verify and save the current record information [CanSave, UpdateElement]
    Move to the previous record in the table
    Refill the **ItemEdit** window [FillElement]

Sub IENext_Click - When user clicks on the next record button
    Verify and save the current record information [CanSave, UpdateElement]
    Move to the next record in the table
    Refill the **ItemEdit** window [FillElement]

Sub IELast_Click - When user clicks on the last record button
    Verify and save the current record information [CanSave, UpdateElement]
    Move to the last record in the table
    Refill the **ItemEdit** window [FillElement]

Sub IECancel_Click - Same as **Views** Menu, Go to Previous View

Sub IEOldViews_Click - Same as **Views** Menu, Old Views

Sub IEPrint_Click - User clicks on the printer icon (print) button
    Get print destination from user (show **PrintDest** window)
    Print information to printer or to a file [PrintRecord, FileRecord]
    (Note, does not save information first, so they can print without actually making a change to the data)

Sub IEReturn_Click - User clicks on the "Return" button
    Verify and save the current record information [CanSave, UpdateElement]
    Unload and exit the window

4.3.4 RelTable Window (RELTABLE.FRM)

The relations table window, **RelTable**, is similar in design and format to the **MainWindow** form except that the table used by the form is oriented around the relation tables rather than the element tables. In this window, a relation table is selected, and the form's table is filled with all of the element pairs for the selected relation. The window is a resizable window available from the **MainWindow**.

Menus: The **RelTable** window has the following menus:

Edit Menu: Same as the Edit menu of the **MainWindow**
Help Menu: Similar to the **MainWindow** Help menu

Menu Event Procedures:

Edit Menu: Same as Edit menu of the **MainWindow**

Help Menu: Display the appropriate on-line help information [Get_Help]

Controls: The following are controls used by the **RelTable** form:

RelationList List Box: Select the relation table to display
RelationGrid Table: Display the element pairs of the relation
SortButton Button [SortB]: Select the sorting order for the table
OldViews Button: Display the **OldViews** window
Print Button [IEPrint]: Print current table
Return Button [QuitB]: Exit the window
FirstPage Button [Mh3dButnl(0)]: Move to first Virtual Page
PreviousPage Button [Mh3dButnl(1)]: Move to the previous Virtual Page
NextPage Button [Mh3dButnl(2)]: Move to the next Virtual Page
LastPage Button [Mh3dButnl(3)]: Move to the last Virtual Page
PageLabel Text Box: Display current Virtual Page

Form and Control Event Procedures: The following are the events associated with the **MainWindow** form and the controls listed above.

Sub Form_Load - Triggered when the window is loaded
Initialize the **RelationGrid** [InitRelGrid]

Sub Form_Resize - Triggered when window is resized by user or by software
Check for minimum window width
Resize **RelationGrid** table
Align control buttons at bottom of window

Sub RelationList_Click - User clicks on relation list box
Set new current relation table
Refill **RelationGrid** with records from new element [UpdateRelGrid]

Sub RelationGrid_DblClick - User double-clicks on relation grid record
Determine element record clicked
If record deleted, then display warning and ignore event
Otherwise, display the selected record [ShowElement]

Sub SortB_Click - User clicks on the "Sort" button
Sort and redisplay the **RelationGrid** by source or target [UpdateRelGrid]

Sub OldViews_Click - User clicks on the "Old Views" button
Display the **OldViews** window

Sub IEPrint_Click - User clicks on the "Print" button
Print Element Type, Number and Name fields for all rows

Sub QuitB_Click - User clicks on the "Return" button
Unload and exit the **RelTable** window

Sub PageLabel_DblClick - User double-clicks on the virtual page label box
Get virtual page number from user
Set new top virtual page
Refill **RelationGrid** [UpdateRelGrid]

Sub FirstPage_Click [Mh3dButn1(0)] - User clicks "First Page" button
Set top virtual page to 1
Refill **RelationGrid** [UpdateRelGrid]

Sub PreviousPage_Click [Mh3dButn1(1)] - User clicks "Previous Page" button
Decrement top virtual page by 1
Refill **RelationGrid** [UpdateRelGrid]

Sub NextPage_Click [Mh3dButn1(2)] - User clicks "Next Page" button
Increment top virtual page by 1
Refill **RelationGrid** [UpdateRelGrid]

Sub LastPage_Click() [Mh3dButn1(3)] - User clicks "Last Page" button
Set top virtual page to last page
Refill **RelationGrid** [UpdateRelGrid]

4.3.5 OldViews Window (OLDVIEWS.FRM)

The **OldViews** window, or Previous Views window, is used to display all of the element records the user has viewed since launching the program. This keeps a running history of records visited. Double-clicking on a previously viewed record will return to that element record. The window is a resizable window that can be accessed from the **MainWindow, ItemEdit** window, or **RelTable** window.

Menus: The **OldViews** Window contains the standard "Help" and "Edit" menus.

Menu Event Procedures: The "Help" and "Edit" menu event procedures are similar to the **MainWindow** menu event procedures.

Controls: The following controls are used with the **OldViews** window:

ViewsGrid Table: Displays previous views
Print Button [IEPrint]: Print views list
Clear Button [Command2]: Clears the list
Return Button [Command1]: Exits the window

Form and Control Event Procedures: The following are the events associated with the **OldViews** window form and controls listed above.

```
Sub Form_Load - Triggered when form is loaded
    Initialize the ViewsGrid table
    Fill ViewsGrid with all views from previous view stack

Sub Form_Resize - Triggered when form is resized by user or software
    Check for minimum window size
    Resize ViewsGrid table and position buttons

Sub ViewsGrid_DblClick - User double-clicks on an old view element record
    Determine element record selected
    Display selected element record  [ShowElement]

Sub IEPrint_Click - User clicks on the "Print" button
    Print all rows to the printer

Sub Command2_Click - User clicks on the "Clear" button
    Resize and reset view stack size to zero
    Refresh the ViewsGrid table

Sub Command1_Click - User clicks on the "Return" button
    Close the OldViews window
```

4.3.6 SearchWindow (SEARCHWI.FRM)

The **SearchWindow** form was the original window designed to allow the user to do searches on various attributes and relations, in a very flexible way. However, many users have had trouble understanding how to do queries through this window. Subsequently, the **frmSingleQuery** and **MultiSearch** windows were developed (they are discussed below). The **SearchWindow** is displayed when the Multiple Tables command is selected from the **Query** menu from either the **MainWindow** or **ItemEdit** window. The window is resizable.

Menus: The **SearchWindow** has the following menus:

Edit Menu: Same as **MainWindow** Edit menu
Elements Menu
  Change Search Element Type: Select element type for search
Help Menu: Similar to **MainWindow** Help menu

Menu Event Procedures: The following are the event procedures for the menus shown above.

Edit Menu: Same as **MainWindow** Edit menu

```
Elements Menu, Change Search Element Type  [mnuSWCST_Click]
    Get new element type from user through frmElemSelect window
    Reload the SearchWindow

Help Menu:  Display the appropriate on-line help topic  [Get_Help]
```

Controls: The following controls are found on the **SearchWindow**:

HistoryGrid Table: Displays the search history as it is built
AttributeList List Box [AttrList]: Attribute or Relation for search
RelTypeList List Box: Relation type selection box
RelAttrList List Box: Attribute list of related element type
ResultsGrid Table: Displays the results of the search
AND Button [Command1]: Do a search with the conjunction AND
OR Button [Command2]: Do a search with the conjunction OR
Search Button [Command3]: Do the actual search
Reset Button [Command5]: Clear search results
SQL Button [Command6]: Show Structured Query Language (SQL) string
Print Button [IEPrint]: Print the search results
Return Button [Command4]: Exit from the window

Form and Control Event Procedures: The following are the events associated with the **SearchWindow** form and the controls listed above.

```
Sub Form_Load - Triggered when the window is loaded
     Initialize search query strings
     Initialize table headers
     Initialize AttrList and OpList lists

Sub Form_Resize - Triggered when the form is resized by the user or software
     Check for and set minimum window size
     Set location of command buttons, resize HistoryGrid and ResultsGrid

Sub AttrList_Click - User clicks on an attribute list item
     Determine attribute or relation selected
     If attribute selected:
          Set operation list depending on attribute data type
          Display ValueList or ValueText box
          Disable related type and related attribute lists
     If relation selected:
          Enable related type and related attribute lists
          Set up operation list depending on related attribute data type
          Display ValueList or ValueText box

Sub RelTypeList_Click - User clicks on related element type list item
     Determine the related element type selected
     Refill the RelAttrList list appropriately

Sub RelAttrList_Click - User clicks on related element attribute list item
     Determine the related attribute selected
     Fill the OpList with appropriate operations for attribute data type
     Enable the appropriate ValueList and ValueText boxes

Sub ResultsGrid_DblClick - User double-clicks on record in results grid
     Determine the record selected
     Display the selected record  [ShowElement]
```

Sub Command1_Click - User clicks on the "AND" button
If just completed previous search, add "AND" to search string
Otherwise add "AND" to the next search clause [AddToSearch]

Sub Command2_Click - User clicks on the "OR" button
If just completed previous search, add "OR" to search string
Otherwise add "OR" to the next search clause [AddToSearch]

Sub Command3_Click - User clicks on the "Search" button
Do the actual search [AddToSearch]

Sub Command5_Click - User clicks on the "Reset" button
Reset search strings
Clear **ResultsGrid**

Sub Command6_Click - User clicks on the "SQL" button
Display the current search SQL string in a MsgBox

Sub IEPrint_Click - User clicks on the "Print" button
Print search header and criteria
Determine form to use (table or labeled text)
Print all of the result record information

Sub Command4_Click - User clicks on the "Return" button
Unload the **SearchWindow**

4.3.7 frmSingleQuery (FRMSQRY.FRM)

This window was developed to provide a simple interface for searching for records within a single element table. The **frmSingleQuery** window is displayed when the Single Tables command is selected from the **Query** menu from either the **MainWindow** or **ItemEdit** window. This window was originally developed by Nancy Graves.

Menus: The following menus are available in the **frmSingleQuery** window:

File Menu
Show SQL: Show the current SQL string
Print: Print the search results
Exit: Exit from the window
Edit Menu: Same as the **MainWindow** Edit menu
Elements Menu: Same as the **SearchWindow** Elements menu
Sort Menu
By Name: Set sorting order of results to by name
By Number: Set sorting order of results to by number
Help Menu: Similar to the **MainWindow** help menu

Menu Event Procedures: The following are the event procedures associated with the menus above.

File Menu, Show SQL [mnuShowSQL]
Build SQL search string and display to the user

File Menu, Print [mnuPrint]
Create temporary database for search results
Use Crystal Reports to print the search results

File Menu, Exit [mnuExit]
Unload the form and exit the window

Edit Menu: Same as **MainWindow** Edit menu

Elements Menu: Same as **SearchWindow** Elements menu

Sort Menu, By Name [mnuSortName]
Set sorting order flag to by name

Sort Menu, By Number [mnuSortNumber]
Set sorting order flag to by number

Help Menu: Similar to **MainWindow** Help menu

Controls: The following controls are found on the **frmSingleQuery** window

cboOperator(Index) List Box: Lists search operations for each attribute
Do Query Button [cmdExecute]: Executes the query
Reset Query Button [cmdReset]: Resets the search results
Display Field Button [cmdEditRecord]: Display long text value
Return Button [cmdClose]: Close and return from the window
Results Grid Table [Grid1]: Displays the search results

Form and Control Event Procedures: The following are the events associated with the **frmSingleQuery** form and the controls listed above.

Sub Form_Load - Triggered when window is loaded
Initiate window parameters and sizes
Initiate attribute controls [LoadControls]

Sub cboOperator_Click(Index) - User clicks on an operation list item
Set up the **txtInput** or **listIndex** boxes depending on operation selected

Sub cmdClose_Click - User clicks on the "Return" button
Unload the form and exit from window

Sub cmdEditRecord_Click - User clicks on "Display Field" button
Determine record and attribute clicked
Display text value using **frmExpand** window

Sub cmdExecute_Click - User clicks on the "Do Search" button
Create the SQL statement [MakeSQL]
Create snapshot of search results
Fill results grid with results of the search

Sub cmdReset_Click - User clicks on the "Reset Query" button
Clear the search results and search results grid

Sub Grid1_DblClick - User double-clicks on an element record

```
    Determine element record selected
    Display the selected element record  [ShowElement]
```

Additional Routines: The following are additional routines defined within the **frmSingleQuery** form.

Sub LoadControls - Loads the various controls for the window

```
    For each attribute of the current element type:
        Create a label, operator list, and text or list input box
        Fill in the operator list with appropriate operations
```

Function MakeSQL - Returns the current SQL statement

```
    For each operation that is not "All
        Add search clause to the current search strings
```

4.3.8 MultiSearch (MULTSRCH.FRM)

The **MultiSearch** window was developed to be an easier interface for doing a two table query. The window allows the user to select search criteria on the attributes of two element tables, which are related through a valid relation. The search routine then displays the element records in both tables that met the search criteria and the elements in the two tables are related through a selected relationship. The **MultiSearch** window is displayed when the Two Table Search command is selected from the **Query** menu from either the **MainWindow** or **ItemEdit** window. This window was originally developed by Nancy Graves.

Menus: The following menus are found on the **MultiSearch** window:

- **File** Menu
  - Show SQL: Show the current SQL string
  - Print: Print the search results
  - Exit: Exit from the window
- **Edit** Menu: Same as the **MainWindow** Edit menu
- **Sort** Menu
  - **Primary Table** Submenu
    - By Name: Sort primary table results by name
    - By Number: Sort primary results table by number
  - **Related Table** Submenu
    - By Name: Sort related table results by name
    - By Number: Sort related table results by number
- **Help** Menu: Similar to the **MainWindow** help menu

Menu Event Procedures: The following are the event procedures associated with the menu commands above.

File Menu, Show SQL [mnuShowSQL]

```
    Build SQL search string and display to the user
```

File Menu, Print [mnuPrint]

```
    Create temporary database for search results
    Use Crystal Reports to print the search results
```

File Menu, Exit [mnuExit]
Unload the form and exit the window

Edit Menu: Same as **MainWindow** Edit menu

Sort Menu, *Primary Table* Submenu, By Name [mnuSortName]
Set sorting order flag to by name

Sort Menu, *Primary Table* Submenu, By Number [mnuSortNumber]
Set sorting order flag to by number

Sort Menu, *Related Table* Submenu, By Name [mnuSortRByName]
Set sorting order flag to by name

Sort Menu, *Related Table* Submenu, By Number [mnuSortRByNumber]
Set sorting order flag to by number

Help Menu: Similar to **MainWindow** Help menu

Controls: The following are controls found on the **MultiSearch** window:

ElementType(Index) List Box: Element type for primary or related table
AttrList(Index) List box: Attribute list for primary or related table
ValueList(Index) List Box: Possible attribute value for either table
ValueText(Index) Text box: Possible attribute search value for table
AND Button [cmdAND]: AND button for search
OR Button [cmdOR]: OR button for search
Reset Button [cmdReset]: Reset the search results
Return Button [cmdReturn]: Unload and exit the window
Search Button [cmdSearch]: Execute the search
ResultsGrid Table: Displays the search results

Form and Control Event Procedures: The following are the event procedures for the **MultiSrch** form and the controls listed above.

Sub Form_Load - Triggered when window is loaded
Initialize the window controls and **ResultsGrid**

Sub ElementList_Click(Index) - ElementList item is selected
Set the element type based on the Index value

Sub AttrList_Click(Index) - Attribute List item is selected
Determine new attribute type selected
Reset and fill the **OpList(Index)**, **ValueList(Index)**, and **ValueText(Index)**

Sub cmdAND_Click - User clicks the "AND" button
Set the last search operation to "AND"

Sub cmdOR_Click - User clicks the "OR" button
Set the last search operation to "OR"

Sub cmdSearch_Click - User clicks the "Search" button
Create the SQL query string [MakeSQL] and do the search
Fill the **ResultsGrid** with results [FillGrid]

```
Sub cmdReset_Click - User clicks the "Reset" button
     Clear all of the search values and ResultsGrid

Sub cmdReturn_Click - User clicks the "Return" button
     Unload the MultiSearch window

Sub ResultsGrid_DblClick - User double-clicks on an element record result
     Determine the element record selected
     Display the element record selected  [ShowElement]
```

Additional Routines: The following are additional routines defined within the **MultiSearch** form.

```
Sub ClearGrid - Clears the results grid
     Clears and resets the ResultsGrid

Sub FillGrid - Fills the results grid with the results of the search
     Clear the results grid  [ClearGrid]
     Build the SQL query string and do the search
     For each record in the search results, fill in the ResultsGrid

Sub MakeSQL - Builds the SQL search string
     Determine search string for primary search table attributes
     Determine relation used between element types
     Determine search string for related search table
     Build the clauses
```

4.3.9 AddRelations Window (ADDRELAT.FRM)

The **AddRelations** window is used to add relations to an element record. The window is displayed when the Add Relation(s) command is selected from the **Records** menu of the **ItemEdit** window. The user selects the related element type desired, the appropriate relation, and then the records to be related. The window is resizable.

Menus: The **AddRelations** window contains only the **Help** menu.

Menu Event Procedures: The **Help** menu behaves similar to the Help menu of the **MainWindow**.

Controls: The following are the controls on the **AddRelations** window

FirstPage Button [Mh3dButnl(0)]: Move to first Virtual Page
PreviousPage Button [Mh3dButnl(1)]: Move to the previous Virtual Page
NextPage Button [Mh3dButnl(2)]: Move to the next Virtual Page
LastPage Button [Mh3dButnl(3)]: Move to the last Virtual Page
PageLabel Text Box: Display current Virtual Page
Valid Types List Box [VTList]: List of valid related element types
Relation List List Box [RList]: List valid relations for element types
Related Elements List Box [EL2]: Displays records from related table
OK Button [Command2]: Add the selected relation links
Cancel Button [Commandl]: Unload and exit from the window

Form and Control Event Procedures: The following are the event procedures for the form and the controls listed above.

```
Sub Form_Load - Triggered when the window is loaded
    Size and place the controls based on the window size
    Add the valid related element types to the VTList list box

Sub Form_Resize - Triggered when the window is resized by the user or software
    Resize element list table EL2 to fit window and place buttons

Sub VTList_Click - User clicks on Valid Types list item
    Determine the relation(s) between selected element types
    Update the RList with valid relation(s) between elements

Sub RList_Click - User clicks on Relation List item (or by software)
    Fill EL2 table with the records from the selected table
    Reset the virtual mode buttons  [SetAVButtons]

Sub PageLabel_DblClick - User double clicks on the virtual page number box
    Get virtual page number from user
    Set new top virtual page
    Refill window table  [RList_Click]

Sub FirstPage_Click  [Mh3dButn1(0)] - User clicks the "First Page" button
    Set top virtual page to 1
    Refill window table  [RList_Click]

Sub PreviousPage_Click  [Mh3dButn1(1)] - User clicks "Previous Page" button
    Decrement top virtual page by 1
    Refill window table  [RList_Click]

Sub NextPage_Click  [Mh3dButn1(2)] - User clicks the "Next Page" button
    Increment top virtual page by 1
    Refill window table  [RList_Click]

Sub LastPage_Click  [Mh3dButn1(3)] - User clicks the "Last Page" button
    Set top virtual page to last page
    Refill window table  [RList_Click]

Sub Command2_Click - User clicks the "OK" button
    Add the selected relations to the current element  [AddTheRelations]
    Return to the ItemEdit window and refresh grid  [FillIERelGrid]

Sub Command1_Click - User clicks the "Cancel" button
    Unload and exit the window
```

Additional Routines: The routines below are defined within the **AddRelations** window.

```
Sub SetAVButtons - Sets the status of the virtual mode buttons
    Set the visibility and enabled status of the virtual mode buttons
```

4.3.10 InheritRelations Window (INHERITR.FRM)

The **InheritRelations** window is used to inherit relations to an element record based on another element record. The window is displayed when the Inherit Relation(s) command is selected from the **Records** menu of the **ItemEdit** window. The user selects the related element type desired, the appropriate relation, and then the records to be related. The window is nearly identical to the **AddRelations** window. The window is resizable.

Menus: The **InheritRelations** window contains only the **Help** menu.

Menu Event Procedures: The **Help** menu behaves similar to the Help menu of the **MainWindow**.

Controls: The following are the controls on the **InheritRelations** window

FirstPage Button [Mh3dButnl(0)]: Move to first Virtual Page
PreviousPage Button [Mh3dButnl(1)]: Move to the previous Virtual Page
NextPage Button [Mh3dButnl(2)]: Move to the next Virtual Page
LastPage Button [Mh3dButnl(3)]: Move to the last Virtual Page
PageLabel Text Box: Display current Virtual Page
Valid Types List Box [VTList]: List of valid related element types
Relation List List Box [RList]: List valid relations for element types
Related Elements List Box [EL2]: Displays records from related table
OK Button [Command2]: Add the selected relation links
Cancel Button [Commandl]: Unload and exit from the window

Form and Control Event Procedures: The following are the event procedures for the form and the controls listed above.

```
Sub Form_Load - Triggered when the window is loaded
    Size and place the controls based on the window size
    Add the valid related element types to the VTList list box

Sub Form_Resize - Triggered when the window is resized by the user or software
    Resize element list table EL2 to fit window and place buttons

Sub VTList_Click - User clicks on a Valid Type list item
    Determine the relation(s) between selected element types
    Update the RList with valid relation(s) between elements

Sub RList_Click - User clicks on a relation list item (or software)
    Fill EL2 table with the records from the selected table
    Reset the virtual mode buttons [SetAVButtons]

Sub PageLabel_DblClick - User double clicks on the virtual page number box
    Get virtual page number from user
    Set new top virtual page
    Refill window table [RList_Click]

Sub FirstPage_Click [Mh3dButnl(0)] - User clicks the "First Page" button
    Set top virtual page to 1
    Refill window table [RList_Click]
```

```
Sub PreviousPage_Click  [Mh3dButn1(1)] - User clicks "Previous Page" button
     Decrement top virtual page by 1
     Refill window table  [RList_Click]
```

```
Sub NextPage_Click  [Mh3dButn1(2)] - User clicks the "Next Page" button
     Increment top virtual page by 1
     Refill window table  [RList_Click]
```

```
Sub LastPage_Click  [Mh3dButn1(3)] - User clicks the "Last Page" button
     Set top virtual page to last page
     Refill window table  [RList_Click]
```

```
Sub Command2_Click - User clicks the "OK" button
     Inherit selected relations to the current element  [InheritTheRelations]
     Return to the ItemEdit window and refresh grid  [FillIERelGrid]
```

```
Sub Command1_Click - User clicks the "Cancel" button
     Unload and exit the window
```

<u>Additional Routines</u>: The routines below are defined within the **InheritRelations** window.

```
Sub SetAVButtons - Set the virtual mode buttons
     Set the visibility and enabled status of the virtual mode buttons
```

4.3.11 CustomViews Window (CUSTOMVI.FRM)

The **CustomViews** window is used to customize the viewing of attributes and relations within the **ItemEdit** window. The window is displayed when the user clicks on the "Custom..." option for the "Data View" list box on the preferences window, **PrefWind**. The user can then click on the desired attributes to be displayed or not displayed. The user can save the parameters in a external custom view file (.CVF file). The window is scaled to fit the size of the user's screen.

<u>Menus</u>: The following menus are available in the **CustomViews** window:

- **File** Menu
  - <u>Open</u>: Open a custom view file
  - <u>Save</u>: Save settings in current custom view file
  - <u>Save As...</u>: Save settings in a new custom view file
- **Help** Menu: Similar to the **MainWindow** Help menu

<u>Menu Event Procedures</u>: The following are the event procedures associated with the menu items listed above.

```
File Menu, Open  [mnuCVOpen]
     Get .CVF file to open with a "File Open" dialogue box
     Load the custom view parameters  [LoadCustomView]
```

```
File Menu, Save  [mnuCVSave]
     Query user to overwrite .CVF file
     Save current view parameters  [SaveCurrentView]
```

File Menu, Save As...: [mnuCVSaveAs]
    Get new .CVF file to save with a "File Save" dialogue box
    Save the custom view parameters [SaveCurrentView]

Help Menu: Displays the appropriate On-Line help topic [Get_Help]

Controls: The following are the controls used on the **CustomViews** window:

AttributeBox(Index) Check Boxes: Check box to display each attribute
DisplayGridBox Check Box: Option to display relations grid
CancelButton Button: Cancels the view changes and exits the window
OKbutton Button: Save the current view changes and exits the window

Form and Control Event Procedures: The event procedures below are used for the form and the controls listed above.

```
Sub Form_Load - Triggered when the window is loaded
    Set the width of the window to the size of the screen
    Set up the AttributeBox(Index) boxes [DisplayCVAs]

Sub CancelButton_Click - User clicks the "Cancel" button
    Unload all of the AttributeBox(Index) check boxes
    Unload the form

Sub OKButton_Click - User clicks the "OK" button
    Save the current custom view parameters
    Unload all of the AttributeBox(Index) check boxes
    Unload the form
```

Additional Routines: The following are the additional routines defined in the **CustomViews** window.

```
Sub DisplayCVAs - Displays the custom views attributes check boxes
    For all attributes (except Name)
    Clone an AttributeBox control and display
    Display the DisplayGridBox
```

4.3.12 ChangeWindow (CHANGEWI.FRM)

The **ChangeWindow** is used to display all of the changes to the current data file. Note that this is different from the **HistoryWindow** that displays all of the changes since the baseline data. As discussed in the database section, changes are organized into three groups: Elements, Attributes, and Relations. The **ChangeWindow** will display a summary of the records from one of the three tables, and will display the appropriate detailed information window when a change record is double-clicked. The window is displayed when the View Changes command is selected from the **Records** menu of the **MainWindow**. The window is resizable.

Menus: The **ChangeWindow** has the following menus:

Edit Menu: The same as the **MainWindow** Edit menu
Sort Menu
by Element: Sort the table by element type and name
by Date/Time: Sort table by last modified date (descending)
by Change: Sort table by type of change
by Target: Sort table by target element type (RelationChanges)
Help Menu: Display the appropriate on-line help items

Menu Event Procedures: The following are the event procedures associated with the menus above.

Edit Menu: Same as the **MainWindow** Edit menu

Sort Menu, by Element [mnuCWSortElement]
Reset the change table sorting index
Refill the change table [RefillACTable, RefillECTable, RefillRCTable]

Sort Menu, by Date/Time: [mnuCWSortTime]
Reset the change table sorting index to "DateTimeIndex"
Refill the change table [RefillACTable, RefillECTable, RefillRCTable]

Sort Menu, by Change: [mnuCWSortChange]
Reset the change table sorting index to "ChangeIndex"
Refill the change table [RefillECTable or RefillRCTable]

Sort Menu, by Target: [mnuCWSortTarget]
Reset the RelationChanges table sorting index to "TargetIndex"
Refill the change table [RefillRCTable]

Help Menu: Display the appropriate on-line help topics [Get_Help]

Controls: The following controls are defined for the **ChangeWindow**:

Elements Option [Option1]: Display records from ElementChanges table
Relations Option [Option2]: Display records from RelationChanges table
Attributes Option [Option3]: Display records from AttributeChanges
ChangeGrid Table: Main table of the window to display change summaries
IEPrint Button: Print the current change table records
FirstPage Button [Mh3dButn1(0)]: Move to first Virtual Page
PreviousPage Button [Mh3dButn1(1)]: Move to the previous Virtual Page
NextPage Button [Mh3dButn1(2)]: Move to the next Virtual Page
LastPage Button [Mh3dButn1(3)]: Move to the last Virtual Page
PageLabel Text Box: Display current Virtual Page
Return Button [Command1]: Exit from the window

Form and Control Event Procedures: The event procedures below are used by the form and the controls listed above.

Sub Form_Load - Triggered when the window is loaded
Resize the window, **ChangeGrid**, and position button and option controls
Set the correct option button to TRUE [Option1, Option2, or Option3]
(Note that the Option buttons will fill the **ChangeGrid**)

```
Sub Form_Resize - Triggered when the window is resized by the user or software
     Scale the ChangeGrid to fit the window
     Position the option buttons and control buttons

Sub Option1_Click - User selects the "Elements" view option
     Enable and disable appropriate menus
     Set sorting index
     Fill the ChangeGrid table  [RefillECTable]

Sub Option2_Click - User selects the "Relations" view option
     Enable and disable appropriate menus
     Set sorting index
     Fill the ChangeGrid table  [RefillRCTable]

Sub Option3_Click - User selects the "Attributes" view option
     Enable and disable appropriate menus
     Set sorting index
     Fill the ChangeGrid table  [RefillACTable]

Sub ChangeGrid_DblClick - User double-clicks on change history record
     Determine change record clicked
     Display the appropriate window  [ARecord, ERecord, or RRecord]

Sub IEPrint_Click - User clicks the "Print" button
     Determine the printing destination by displaying PrintDest window
     Print the information to file or printer  [FCRs or PCRs]

Sub PageLabel_DblClick - User double-clicks the page label box
     Get virtual page number from user
     Set new top virtual page
     Refill window table  [RefillACTable, RefillECTable, or RefillRCTable]

Sub FirstPage_Click  [Mh3dButn1(0)] - User clicks the "First Page" button
     Set top virtual page to 1
     Refill window table  [RefillACTable, RefillECTable, or RefillRCTable]

Sub PreviousPage_Click  [Mh3dButn1(1)] - User clicks "Previous Page" button
     Decrement top virtual page by 1
     Refill window table  [RefillACTable, RefillECTable, or RefillRCTable]

Sub NextPage_Click  [Mh3dButn1(2)] - User clicks the "Next Page" button
     Increment top virtual page by 1
     Refill window table  [RefillACTable, RefillECTable, or RefillRCTable]

Sub LastPage_Click  [Mh3dButn1(3)] - User clicks the "Last Page" button
     Set top virtual page to last page
     Refill window table  [RefillACTable, RefillECTable, or RefillRCTable]

Sub Command1_Click - User clicks the "Return" button
     Unload and exit from the window
```

Additional Routines: The routines below are also defined in the **ChangeWindow** form.

```
Sub FACS - Prints all of the attribute changes to a file
     For all records in the AttributeChanges table
          Print information to the print file  [FileString]

Sub FCRs - Prints all of the change information to a file
     Open a new print output file
     Call the appropriate printing routine  [FACS, FECS, or FRCS]

Sub FECS - Prints all of the element changes to a file
     For all records in the ElementChanges table
          Print information to the print file  [FileString]

Sub FRCS - Print all of the relation changes to a file
     For all records in the RelationChanges table
          Print information to the print file  [FileString]

Sub NoCWButtons - Disables all of the virtual mode buttons
     Disable and make invisible all of the virtual mode buttons

Sub PACS - Prints all of the attribute changes to the printer
     For all records in the AttributeChanges table
          Print information to the printer  [PrintString]

Sub PCRs - Prints all of the change information to the printer
     Call the appropriate printing routine  [PACS, PECS, or PRCS]

Sub PECS - Prints all of the element changes to the printer
     For all records in the ElementChanges table
          Print information to the printer  [PrintString]

Sub PRCS - Prints all of the relation changes to the printer
     For all records in the RelationChanges table
          Print information to the printer  [PrintString]

Sub RefillACTable - Fills the change grid table with attribute change records
     Set the headers and columns of the ChangeGrid
     For all AttributeChange records to be displayed (all or virtual page)
          Add record to the ChangeGrid table
     Set the virtual mode buttons  [SetCWButtons]

Sub RefillECTable - Fills the change grid table with element change records
     Set the headers and columns of the ChangeGrid
     For all ElementChange records to be displayed (all or virtual page)
          Add record to the ChangeGrid table
     Set the virtual mode buttons  [SetCWButtons]

Sub RefillRCTable - Fills the change grid table with relation change records
     Set the headers and columns of the ChangeGrid
     For all RelationChange records to be displayed (all or virtual page)
          Add record to the ChangeGrid table
     Set the virtual mode buttons  [SetCWButtons]
```

Sub SetCWButtons - Sets the virtual mode buttons
    Depending on the current virtual page, display the appropriate buttons

4.3.13 AttributeChanges Record details window, ARecord (ARECORD.FRM)

The **ARecord** window is used to display the detailed information for an AttributeChanges record. This window is displayed when an AttributeChanges record is double-clicked in the **ChangeGrid** of the **ChangeWindow**. The user may change information in this window only to the Reason/Justification field. All other changes are discarded. The window is very similar in form and function to the **ERecord** and **RRecord** windows.

Menus: The **ARecord** form has the standard **Edit** and **Help** menus.

Menu Event Procedures: The **Edit** and **Help** menu event procedures are similar or identical to those of the **MainWindow**.

Controls: The following controls are used by the **ARecord** window:

IEFirst Button: Move to the first AttributeChange record
IEPrevious Button: Move to the previous AttributeChange record
IENext Button: Move to the next AttributeChange record
IELast Button: Move to the last AttributeChange record
Refresh Button [IEOldViews]: Refresh the data
IEReturn Button: Exit from the window

Form and Control Event Procedures: The event procedures below are used by the **ARecord** form and the controls listed above.

```
Sub Form_Load - Triggered when window is loaded
    Go to the correct AttributeChanges table record
    Enable or disable the record navigation buttons
    Fill in the data for the current record  [FillAStuff]

Sub IEFirst_Click - User clicks on the "First Record" button
    Update the current record  [UpdateAChange]
    Move to the first record in the AttributeChanges table
    Fill in the data for the new record  [FillAStuff]

Sub IEPrevious_Click - User clicks on the "Previous Record" button
    Update the current record  [UpdateAChange]
    Move to the previous record in the AttributeChanges table
    Fill in the data for the new record  [FillAStuff]

Sub IENext_Click - User clicks on the "Next Record" button
    Update the current record  [UpdateAChange]
    Move to the next record in the AttributeChanges table
    Fill in the data for the new record  [FillAStuff]

Sub IELast_Click - User clicks on the "Last Record" button
    Update the current record  [UpdateAChange]
    Move to the last record in the AttributeChanges table
    Fill in the data for the new record  [FillAStuff]
```

Sub IERefresh_Click - User clicks on the "Refresh" button
    Refill the current data for the record [FillAStuff]

Sub IEReturn_Click - User clicks on the "Return" button
    Update the current record [UpdateAChange]
    Unload and exit from the window

Additional Routines: The following are additional routines that are defined within the **ARecord** window.

Sub FillAStuff - Fill in the window with the attribute change values
    Fill the text boxes with the details of the AttributeChange record
    Enable and disable the record navigation buttons

Sub UpdateAChange - Save any change to the attribute change justification
    If Justification was changed, save the new justification value

4.3.14 ElementChanges Record details window, ERecord (ERECORD.FRM)

The **ERecord** window is used to display the detailed information for an ElementChange record. This window is displayed when an ElementChange record is double-clicked in the **ChangeGrid** of the **ChangeWindow**. The user may change information in this window only to the Reason/Justification field. All other changes are discarded. The window is very similar in form and function to the **ARecord** and **RRecord** windows.

Menus: The **ERecord** form has the standard **Edit** and **Help** menus.

Menu Event Procedures: The **Edit** and **Help** menu event procedures are similar or identical to those of the **MainWindow**.

Controls: The following controls are used by the **ERecord** window:

IEFirst Button: Move to the first ElementChange record
IEPrevious Button: Move to the previous ElementChange record
IENext Button: Move to the next ElementChange record
IELast Button: Move to the last ElementChange record
Refresh Button [IEOldViews]: Refresh the data
IEReturn Button: Exit from the window

Form and Control Event Procedures: The event procedures below are used by the **ERecord** form and the controls listed above.

Sub Form_Load - Triggered when the window is loaded
    Go to the correct ElementChanges table record
    Enable or disable the record navigation buttons
    Fill in the data for the current record [FillEStuff]

Sub IEFirst_Click - User clicks on the "First Record" button
    Update the current record [UpdateEChange]
    Move to the first record in the ElementChanges table
    Fill in the data for the new record [FillEStuff]

Sub IEPrevious_Click - User clicks on the "Previous Record" button
Update the current record [UpdateEChange]
Move to the previous record in the ElementChanges table
Fill in the data for the new record [FillEStuff]

Sub IENext_Click - User clicks on the "Next Record" button
Update the current record [UpdateEChange]
Move to the next record in the ElementChanges table
Fill in the data for the new record [FillEStuff]

Sub IELast_Click - User clicks on the "Last Record" button
Update the current record [UpdateEChange]
Move to the last record in the ElementChanges table
Fill in the data for the new record [FillEStuff]

Sub IERefresh_Click - User clicks on the "Refresh" button
Refill the current data for the record [FillEStuff]

Sub IEReturn_Click - User clicks on the "Return" button
Update the current record [UpdateEChange]
Unload and exit from the window

Additional Routines: The following are additional routines which are defined within the **ERecord** window.

Sub FillEStuff - Fills in the element change information
Fill the text boxes with the details of the ElementChange record
Enable and disable the record navigation buttons

Sub UpdateEChange - Updates any change to the element change justification
If Justification was changed, save the new justification value

4.3.15 RelationChanges Record details window, RRecord (RRECORD.FRM)

The **RRecord** window is used to display the detailed information for a RelationChange record. This window is displayed when a RelationChange record is double-clicked in the **ChangeGrid** of the **ChangeWindow**. The user may change information in this window only to the Reason/Justification field. All other changes are discarded. The window is very similar in form and function to the **ARecord** and **ERecord** windows.

Menus: The **RRecord** form has the standard **Edit** and **Help** menus.

Menu Event Procedures: The **Edit** and **Help** menu event procedures are similar or identical to those of the **MainWindow**.

Controls: The following controls are used by the **RRecord** window:

IEFirst Button: Move to the first RelationChange record

IEPrevious Button: Move to the previous RelationChange record

IENext Button: Move to the next RelationChange record

IELast Button: Move to the last RelationChange record

Refresh Button [IEOldViews]: Refresh the data

IEReturn Button: Exit from the window

Form and Control Event Procedures: The event procedures below are used by the **RRecord** form and the controls listed above.

```
Sub Form_Load - Triggered when the window is initially loaded
    Go to the correct RelationChanges table record
    Enable or disable the record navigation buttons
    Fill in the data for the current record  [FillRStuff]

Sub IEFirst_Click - User clicks on the "First Record" button
    Update the current record  [UpdateRChange]
    Move to the first record in the RelationChanges table
    Fill in the data for the new record  [FillRStuff]

Sub IEPrevious_Click - User clicks on the "Previous Record" button
    Update the current record  [UpdateRChange]
    Move to the previous record in the RelationChanges table
    Fill in the data for the new record  [FillRStuff]

Sub IENext_Click - User clicks on the "Next Record" button
    Update the current record  [UpdateRChange]
    Move to the next record in the RelationChanges table
    Fill in the data for the new record  [FillRStuff]

Sub IELast_Click - User clicks on the "Last Record" button
    Update the current record  [UpdateRChange]
    Move to the last record in the RelationChanges table
    Fill in the data for the new record  [FillRStuff]

Sub IERefresh_Click - User clicks on the "Refresh" button
    Refill the current data for the record  [FillRStuff]

Sub IEReturn_Click - User clicks on the "Return" button
    Update the current record  [UpdateRChange]
    Unload and exit from the window
```

Additional Routines: The following are additional routines which are defined within the **RRecord** window.

```
Sub FillRStuff - Fill the window with the relation change information
    Fill the text boxes with the details of the RelationChange record
    Enable and disable the record navigation buttons

Sub UpdateRChange - Updates any change in the relation change justification
    If Justification was changed, save the new justification value
```

4.3.16 HistoryWindow (HISTORYW.FRM)

The **HistoryWindow** is used to display all of the changes since a fixed baseline to a current element record being displayed in the **ItemEdit** window. To do this, the window displays information from the baseline and change history files located on the shared network server (currently \\AP001\SE-TOOLS). The window will allow the user to also display the differences between various versions of text attributes for the element through the **SelectCompare** and **Compare** windows. The window consists of three tables for the element, attribute, and relation changes to the element, plus various buttons and menus. The window is resizable.

Menus: The **HistoryWindow** has the following menus:
Edit Menu: Same as the **MainWindow** Edit menu
Compare Menu: Does a comparison by displaying **SelectCompare** window
Help Menu: Similar to the **MainWindow** Help menu

Menu Event Procedures: The following are the event procedures for the menus listed above.

Edit Menu: Same as the **MainWindow** Edit menu.

Compare Menu Item: [mnuHCompare_Click]
Get comparison to make and make it [GetCompare]

Help Menu: Displays the appropriate on-line help topic [Get_Help]

Controls: The following controls are found on the **HistoryWindow**:

EGrid Table: Displays element changes in history
AGrid Table: Displays attribute changes in history
RGrid Table: Displays relation changes in history
ReturnButton Button: Returns and exits from the window

Form and Control Event Procedures: The following are the events associated with the **HistoryWindow** form and the controls listed above.

Sub Form_Load - Called when form is initially loaded
Initialize window size and grid columns/sizes
Fill in the data for the grids [FillHAGrid, FillHEGrid, FillHRGrid]

Sub Form_Resize - Triggered when form is resized
Resize the tables to fit the window in width and height
Reposition labels and buttons

Sub AGrid_DblClick - User double-clicks on attribute change history record
Display the attribute change information in the **HARecord** window

Sub EGrid_DblClick - User double-clicks on element change history record
Displays the element change information in the **HERecord** window

Sub RGrid_DblClick - User double-clicks on relation change history record
Displays the relation change information in the **HRRecord** window

```
Sub ReturnButton_Click - User clicks the "Return" button
     Close the change history tables and database
     Unload the HistoryWindow
```

Additional Routines: The following are additional routines which are defined within the **HistoryWindow** window.

```
Sub FillHAGrid - Fill the AGrid table with all attribute changes
     For all changes in the AttributeChanges table for the current element
          Add attribute change history to AGrid

Sub FillHEGrid - Fill the EGrid table with all element changes
     For all changes in the ElementChanges table for the current element
          Add element change history to EGrid

Sub FillHRGrid - Fill the RGrid table with all relation changes
     For all changes in the RelationChanges table for the current element
          Add relation change history to RGrid
```

4.3.17 History Attribute Record window (HARECORD.FRM)

The **HARecord** window is used to display the detailed information for an Attribute Change History record. This window is displayed when an Attribute Change History record is double-clicked in the **AGrid** of the **HistoryWindow**. The user may not change any information in this window. The window is very similar in form and function to the **HERecord** and **HRRecord** windows.

Menus: The **HARecord** form has the standard **Edit** and **Help** menus.

Menu Event Procedures: The **Edit** and **Help** menu event procedures are similar or identical to those of the **MainWindow**.

Controls: The following controls are used by the **HARecord** window:

IEFirst Button: Move to the first Attribute Change History record
IEPrevious Button: Move to the previous Attribute Change History record
IENext Button: Move to the next Attribute Change History record
IELast Button: Move to the last Attribute Change History record
IEReturn Button: Exit from the window

Form and Control Event Procedures: The event procedures below are used by the **HARecord** form and the controls listed above.

```
Sub Form_Load - Triggered when window is loaded
     Fill in the data for the current record  [FillHAStuff]

Sub IEFirst_Click - User clicks on the "First Record" button
     Move to the first record in the Attribute Change History table
     Fill in the data for the new record  [FillHAStuff]

Sub IEPrevious_Click - User clicks on the "Previous Record" button
     Move to the previous record in the Attribute Change History table
     Fill in the data for the new record  [FillHAStuff]
```

Sub IENext_Click - User clicks on the "Next Record" button
Move to the next record in the Attribute Change History table
Fill in the data for the new record [FillHAStuff]

Sub IELast_Click - User clicks on the "Last Record" button
Move to the last record in the Attribute Change History table
Fill in the data for the new record [FillHAStuff]

Sub IEReturn_Click - User clicks on the "Return" button
Unload and exit from the window

Additional Routines: The following are additional routines which are defined within the **HARecord** window.

Sub FillHAStuff - Fill in the window with the attribute change history values
Fill the text boxes with the Attribute Change History record
Enable and disable the record navigation buttons

4.3.18 History Element Record window (HERECORD.FRM)

The **HERecord** window is used to display the detailed information for an Element Change History record. This window is displayed when an Element Change History record is double-clicked in the **EGrid** of the **HistoryWindow**. The user may not change any information in this window. The window is very similar in form and function to the **HARecord** and **HRRecord** windows.

Menus: The **HERecord** form has the standard **Edit** and **Help** menus.

Menu Event Procedures: The **Edit** and **Help** menu event procedures are similar or identical to those of the **MainWindow**.

Controls: The following controls are used by the **HERecord** window:

IEFirst Button: Move to the first Element Change History record
IEPrevious Button: Move to the previous Element Change History record
IENext Button: Move to the next Element Change History record
IELast Button: Move to the last Element Change History record
IEReturn Button: Exit from the window

Form and Control Event Procedures: The event procedures below are used by the **HERecord** form and the controls listed above.

Sub Form_Load - Triggered when window is loaded
Fill in the data for the current record [FillHEStuff]

Sub IEFirst_Click - User clicks on the "First Record" button
Move to the first record in the Element Change History table
Fill in the data for the new record [FillHEStuff]

Sub IEPrevious_Click - User clicks on the "Previous Record" button
Move to the previous record in the Element Change History table
Fill in the data for the new record [FillHEStuff]

```
Sub IENext_Click - User clicks on the "Next Record" button
     Move to the next record in the Element Change History table
     Fill in the data for the new record  [FillHEStuff]

Sub IELast_Click - User clicks on the "Last Record" button
     Move to the last record in the Element Change History table
     Fill in the data for the new record  [FillHEStuff]

Sub IEReturn_Click - User clicks on the "Return" button
     Unload and exit from the window
```

Additional Routines: The following are additional routines which are defined within the **HERecord** window.

```
Sub FillHEStuff - Fill in the window with the element change history values
     Fill the text boxes with the Element Change History record
     Enable and disable the record navigation buttons
```

4.3.19 History Relation Record window (HRRECORD.FRM)

The **HRRecord** window is used to display the detailed information for a Relation Change History record. This window is displayed when a Relation Change History record is double-clicked in the **RGrid** of the **HistoryWindow**. The user may not change any information in this window. The window is very similar in form and function to the **HARecord** and **HERecord** windows.

Menus: The **HRRecord** form has the standard **Edit** and **Help** menus.

Menu Event Procedures: The **Edit** and **Help** menu event procedures are similar or identical to those of the **MainWindow**.

Controls: The following controls are used by the **HRRecord** window:

IEFirst Button: Move to the first Relation Change History record
IEPrevious Button: Move to the previous Relation Change History record
IENext Button: Move to the next Relation Change History record
IELast Button: Move to the last Relation Change History record
IEReturn Button: Exit from the window

Form and Control Event Procedures: The event procedures below are used by the **HRRecord** form and the controls listed above.

```
Sub Form_Load - Triggered when window is loaded
     Fill in the data for the current record  [FillHRStuff]

Sub IEFirst_Click - User clicks on the "First Record" button
     Move to the first record in the Relation Change History table
     Fill in the data for the new record  [FillHRStuff]

Sub IEPrevious_Click - User clicks on the "Previous Record" button
     Move to the previous record in the Relation Change History table
     Fill in the data for the new record  [FillHRStuff]
```

Sub IENext_Click - User clicks on the "Next Record" button
Move to the next record in the Relation Change History table
Fill in the data for the new record [FillHRStuff]

Sub IELast_Click - User clicks on the "Last Record" button
Move to the last record in the Relation Change History table
Fill in the data for the new record [FillHRStuff]

Sub IEReturn_Click - User clicks on the "Return" button
Unload and exit from the window

Additional Routines: The following are additional routines which are defined within the **HRRecord** window.

Sub FillHRStuff - Fill in the window with the attribute change history values
Fill the text boxes with the Relation Change History record
Enable and disable the record navigation buttons

4.3.20 SelectCompare Window (SELECTCO.FRM)

The **SelectCompare** window is used to identify which attribute to compare between which historical values, including the baseline and current values. The user selects an attribute, and two historical values to compare. The window will then display a "redline" comparison in the **Compare** window discussed below.

Menus: The **SelectCompare** window contains only the **Help** menu.

Menu Event Procedures: The **Help** menu event procedures are similar to the ones in the **MainWindow**.

Controls: The following are the controls on the **SelectCompare** window:

Attribute List List Box [AttrList]: Attribute for comparison
List1 List: First list of values for comparison
List2 List: Second list of values for comparison
CompareButton Button: Do the actual comparison
ReturnButton Button: Exit from the window

Form and Control Event Procedures: The following are the events associated with the **SelectCompare** window and the controls listed above.

Sub Form_Load - Triggered when window is loaded
Center the window on the screen

Sub AttrList_Click - User selects attribute to be compared
Determine attribute selected
Clear List1 and List2
Add all change history entries to List1 and List2 plus Current

Sub CompareButton_Click - User clicks on the "Compare" button
    Determine change versions selected in List1 and List2
    Find the values for the selected versions
    Compare the values [CompareText]

Sub ReturnButton_Click - User clicks the "Return" button
    Unload the window and exit

4.3.21 Compare Window (COMPARE.FRM)

The **Compare** window is used to display the "redlined" results of a comparison between two historical values of an attribute for an element record. The window displays the old, new, and redlined values. The window is resizable.

Menus: The **Compare** window contains the standard **Edit** and **Help** menus.

Menu Event Procedures: The event procedures for the **Edit** and **Help** menus are the same as or similar to those of the **MainWindow**.

Controls: The following controls are found on the **Compare** window:

PrintButton Button: Prints the results
ReturnButton Button: Exits from the window

Form and Control Event Procedures: The following are the events associated with the **Compare** window and the controls listed above.

Sub Form_Load - Triggered when window is loaded
    Set the window size to the size of the screen

Sub Form_Resize - Triggered when window is resized by the user or software
    Set the width and of the display boxes
    Set the location of the button controls

Sub PrintButton_Click - Prints the contents of the window to the printer
    Print the results to the printer [PrintDeltas]

Sub ReturnButton_Click - Exits the window
    Unload and exit the window

4.4 SUPPORT ROUTINE MODULE DESIGN DESCRIPTIONS

These sections identify the Basic modules (.BAS files) which contain the support routines and functions for the software. Typically, the routines and functions are organized into groups of common functions. A short summary description of each routine is appended to the name. See Appendix A for the location of the routines and argument lists.

4.4.1 Global Variables and Constants (GLOBALS.BAS)

The **GLOBALS.BAS** file identifies the constants and variables that are known globally to all forms and routines in the software. As such, there are no subroutines or functions defined, but definition for record structures, constants, and global variables. The outline below identifies the major ares of definition within this file.

Define maximum array sizes for structure definition arrays
Define Element (and Attribute) array record structure
Define Relation array record structure
Define Enumerated List record structure
Define global program variables for current record/table positions
Define database and table pointers
Define database constants (from DATACONS.TXT)
Define global variables for record views
Define old views and previous views array structures
Define preferences variables and constants
Define custom view description types
Define search routine variables
Define global screen size and miscellaneous constants
Define WinHelp external routine reference
Define data type constants
Define user information variables
Define print destination constants
Define color constants
Define changes view variables
Define FPGrid action constants and functions
Define virtual page constants and variables
Define element type constants
Define change history variables
Define redline method variables

4.4.2 Initialization Routines (INITIAL.BAS)

The initialization routines are used generally to initialize various program parameters. The routines, and their functionality, are listed below.

Function GetDSetRelation - Get a string field from a dynaset
Get a string field value from a Dynaset, return "" if #NULL#

Function GetDSI - Get an integer field from a dynaset
Get an integer field from a Dynaset, return 0 if #NULL#

Function GetElement - Get a string field from an element table
Returns a string field from an element table (superseded by GetString)

Function GetIR - Get an integer field from an element table
Return an integer field from an element table (superseded by GetInt)

Function GetRelation - Get a string field from a relation table
Returns a string field from a relation table (superseded by GetString)

Sub InitProgram - Initialize the program
Initialize program constants (color, default settings, etc.)
Size, position, and initialize the **MainWindow** controls
Initialize the database structure [InitStructure]
Get the user's information [GetUserInfo]
Open initial database (query user if needed) [OpenTheDatabase]
Set the appropriate label nomenclature [SetLabels]
Set the element table indexes
Fill the main window grid [UpdateMainGrid]
Initialize other program variables

Sub InitRelGrid - Initialize the **RelTable** window's **RelationGrid**
Initialize the **RelTable** window controls and **RelationGrid**
Fill the **RelationGrid** [UpdateRelGrid]

Sub RefillMain - Refill the **ElementGrid** of the **MainWindow**
Clear the **MainWindow** **ElementGrid** table
Fill in the <u>Number</u> and <u>Name</u> fields for all rows in the **ElementGrid**

Sub SetRVButtons - Enable or disable the **RelTable** virtual mode buttons
Set the virtual mode buttons for the **RelTable** window

Sub SetVButtons - Enable or disable the **MainWindow** virtual mode buttons
Set the virtual mode buttons for the **MainWindow**

Sub UpdateMainGrid - Refill the **MainWindow** table
Determine size of grid (virtual mode or not) and initialize
Fill in the **ElementGrid** of the **MainWindow** form [RefillMain]
Add the deleted records to the **ElementGrid**
Update the virtual mode buttons [SetVButtons]

Sub UpdateRelGrid - Refill the **RelTable** window table
Determine the size of grid (virtual mode or not) and initialize
Fill in the **RelationGrid** of the **RelTable** form
Add the deleted records to the **RelationGrid**
Update the virtual mode buttons [SetRVButtons]

4.4.3 Database Structure Initialization Routines (INITSTRU.BAS)

The database structure initialization routines found in **INITSTRU.BAS** are used to initialize the database structure definition arrays, and open and close the database. The routines below are used within this file.

Sub Close4Export - Close an "Export to RMSB" file
Close Change tables and database for RMSB export file

Sub CloseTheDatabase - Close the currently open Browser database file
Delete the temporary tables [DeleteTempTables]
Close all tables and the database

Sub InitStructure - Initialize & define database structure definition arrays

```
    Define number of elements, relations, and base attributes
    Initialize element names and nomenclature, and on line help topics
    Initialize base attributes and relations for all elements
    Set up additional attributes for each element type
    Set up additional relations used by each element type
    Initialize relation names and nomenclature, and on line help topics
    Set up valid related element type lists for all relations
    Initialize all of the enumerated attribute lists
```

Sub Open4Export - Open an "Export to RMSB" file

```
    Open change tables and database for exporting
```

Sub OpenTheDatabase - Open a Browser database file

```
    Open database file and determine access level
    Open element tables and set initial Index values
    Open relation tables and set initial Index values
    Open dataversion table
    Open change history tables
    Create the temporary tables  [CreateTempTables]
```

4.4.4 Record Editing Routines (EDITING.BAS)

The record editing routines in the file **EDITING.BAS** are used to do editing of a selected element record. These routines are typically called from the events in the **ItemEdit** window or one of the subordinate windows.

Function AddElement - Add a new element record

```
    For all attributes, add the attribute value
    For all attributes, save the attribute change history
    Add a "Create" to the ElementChanges table  [CreateElement]
    Update main window if necessary  [MGNewColor]
```

Sub AddOldview - add a view to the old views and previous view stacks

```
    Increment stack and place view on the stack
```

Sub AddRel - Add a relation link between elements

```
    Check to make sure the relation doesn't previously exist
    Add the relation and the inverse relation
    Update the RelationChanges tables
    Update attribute values for changed records (date time modified)
```

Function AddTheRelations - Add relations to elements from **AddRelations** window

```
    For all new element records to add the relation
        Add the relation link  [AddRel]
```

Function CanSave - Determine if the record is valid to be saved

```
    (note, this will be modified significantly for Version 4.0)
    For all attributes, check the data types for errors
    If a name, then make sure it is not blank
    If it is a number, make sure it is valid  [IsValidNumberField]
```

```
Sub ChangedAttribute - Change a single attribute
     Look for previous AttributeChanges entry
     If previous, then modify previous, otherwise add new AC record

Sub ColorRList - Fills in the RelationsGrid list in color
     For each related element entry
          Determine color to display and change color

Sub CreateElement - Create a new element record change history entry
     Add "Create" record to ElementChanges table

Sub DeleteElement - Delete an element record from the database
     Delete all related AttributeChanges records
     Delete all related RelationChanges records
     Remove all relation links  [DeleteRelation]
     Remove all inverse relation links
     Delete the entry in the main window if necessary  [MGDelete]
     Find any ElementChanges records and modify, or add new one
     Delete any old or previous views to the element
     Delete the actual element record
     Add deleted "red" row to MainWindow table if necessary

Sub DeleteRelation - Delete a single relation link between elements
     Delete the selected relation
     Update the "color" of the source element
     Update the RelationChanges table with new relation delete record

Sub ElementRename - Rename an element record
     Rename all of the old views in the old views stack
     Change all of the relation table reference names
     Change all of the inverse relation name references
     Change all of the change table reference names
     Add rename change to ElementChanges history table
     Update the main window element grid  [MGDelete, MGInsert, MGNewColor]

Sub ElementRenumber - Change the number attribute of an element record
     Renumber all of the old views in the old views stack
     Change all of the relation table reference numbers
     Change all of the inverse relation number references
     Update the main window element grid  [MGDelete, MGInsert, MGNewColor]

Sub FileRecord - Output element record to a file
     Open the output file and output header information
     Output the element attributes
     Output the element relations
     Close output file

Sub FileString - Prints a string to the printer, with line breaks and spacing
     If string will fit on the line, just print it
     Otherwise, print the string breaking between words at 75 chars wide
```

```
Sub FillElement - Fill in the attributes associated with an element record
    Initialize ItemEdit window and set navigation buttons  [SetButtons]
    Fill in values for all attributes (Loop around a Case statement)
    Add the view to the views list  [AddOldView]
    Set the attribute colors for any changed attributes
    Fill in the relations grid  [ColorRList, FillIERelGrid]

Sub FillFullRG - Fill as full the RelationsGrid of the ItemEdit window
    Initialize size and headers of the RelationsGrid
    Find all related elements and add to RelationsGrid
    Add any deleted or removed elements to the RelationsGrid

Sub FillIERelGrid - Fill for one element type the RelationsGrid
    If fill all elements, then do so and exit  [FillFullRG]
    Initialize size and headers for the RelationsGrid of the ItemEdit window
    Find all related elements of the selected type and add to table
    Add any deleted or removed element links to the table

Sub InheritRels - Inherit relations from other elements
    For all relations of a selected element add to the current element

Function InheritTheRelations - Inherit all relations from InheritRelations
    For each element selected, inherit relations  [InheritRels]

Function IsValidNumberField - Determines if a Number attribute is valid
    If number is null, then return True
    Check each sequence for validity  [IsValidSequence]
    If all sequences valid, then return True, else return false

Function IsValidSequence - Determines if a character sequence is valid
    If empty, then is not a valid sequence.
    If all letters or all numbers, then return True, else return False

Sub PrintAll - Prints out the information for all element records for one type
    (note:  Not currently used in the Browser)
    For all element records for a particular element table
        Print header information
        Print attributes
        Print relations

Sub PrintRecord - Prints out the information for one element record
    Print the header information
    Print the attribute information
    Print the relation information

Sub PrintString - Prints a string to the printer, with line breaks at 75
    If string will fit on the line, just print it
    Otherwise, print the string breaking between words at 75 chars wide
```

Sub RemoveRelation - Removes a relation between elements

```
Remove any previous RelationChanges records
Remove the actual relationship link
Modify the attribute date and time on the source and target elements
Add the relation removal to the RelationChanges table
Delete the row from the RelationsGrid of the ItemEdit window
Add the relation row as a deleted row to the RelationsGrid
If necessary, update the main window  [MGNewColor]
```

Sub SetButtons - Enables/Disables the record navigation buttons on **ItemEdit**

```
Enable or disable buttons according to current record position
```

Sub ShowElement - Displays an element record through the **ItemEdit** window

```
Verify and save current element, if any  [CanSave, UpdateElement]
Fill in the relations list
For all attributes create (clone) a label
Depending on data type, calculate position of the label and attribute
Right justify all labels based on longest label
Create and position all the attribute text, memo, and list boxes
Position the control buttons
Position the RelationGrid and REList list
Fill in the element data [FillElement]
```

Function UpdateElement - Update an element record

```
If read-only, then exit as True
If a new record, then add the new element record and exit
Determine which attributes have changed and save the change
Update the AttributeChanges (and other change tables) as necessary
```

4.4.5 Search Routines (SEARCH.BAS)

These routines are used to support the **SearchWindow** controls and functions. This includes the initialization of searches, doing the actual search, and displaying the results.

Sub AddToSearch - Adds a new clause to the search clause being built

```
Determine the attribute or relation chosen
Determine the related type and attribute (if any)
Determine which operation was selected
Determine the target value for the search
If search value type is invalid, say so and exit
Add any related element types to search clause as needed
Depending on the operation, build the search string accordingly
Update the HistoryGrid of the SearchWindow
If the operation was "" then do the search  [DoTheSearch]
```

Function CheckField - Determines if a value is correct for a given field type

```
For a string type, return True
If integer type, return value of IsNumeric function
For date or time type, return value of IsDate function
Otherwise, return True
```

Sub DoTheSearch - Executes the search of the database

```
Build the SQL search string
Create a QueryDef in the temporary database
Do the actual search (with a snapshot)
If no records found, display results and exit
Otherwise ,fill the ResultsGrid with the name and numbers of records
Delete the QueryDef
```

4.4.6 Data Export Routines (EXPORT.BAS)

The file **EXPORT.BAS** contains the routines to export information out of the Browser in a variety of different formats.

Sub ASCIIChanges - Creates file of changes in ASCII (more readable than .RDT)

```
For all attribute and relation changes for an element
    Print to file the attribute changes
    Print to file the relation changes
```

Sub ASCIIHistory - Print out element history to ASCII text file

```
For all records in the ElementChanges table
    Output change information
```

Sub DBReport - Produces a record count summary report for the current database

```
For all element tables print the record count number
For all relation tables print the record count number
For all change tables print the record count number
```

Sub ExportMatrix - Exports a Functions vs. Requirements matrix (tab delimited)

```
(Note:  Functions are columns, Requirements are rows)
Get output file name from user
For all requirements (rows)
    Clear related function array
    Determine all related functions and put in function array
    Output the row with "X" for all related functions
Tell user when done
```

Sub ExportRMSB - Export a Browser format change file (Access format)

```
Get export filename from user
Copy all records from change tables to new export file
```

Sub InheritReqs - Was a temporary subroutine, currently not used.

Sub RDTChanges - Export the element changes in .RDT format

```
For all attribute and relation changes for an element
    Print to file the attribute changes
    Print to file the relation changes
```

Sub RDTHeader - Print the header information for an .RDT file

```
Print out the header line with RMSB version number
```

Sub RDTHistory - Print out element change history to .RDT file

```
For all elements in the ElementChanges table print a history entry
```

```
Sub RDTPreface - Print the preface section of the .RDT file
    Print out the preface information including date and user name
```

```
Sub ToASCIIText - Export text only report to an ASCII file
    Check output file name
    Output history information  [ASCIIHistory]
    Output change information  [ASCIIChanges]
```

```
Sub ToSuperiorRDT - Export .RDT file as a subordinate
    Open the output .RDT file
    Export header information  [RDTHeader]
    Export preface information  [RDTPreface]
    Export history information  [RDTHistory]
    Export changes information  [RDTChanges]
```

4.4.7 Edit Menu Routines (EDITMENU.BAS)

The routines in **EDITMENU.BAS** provide the ability to interface with the Windows clipboard. This is done only for text, and not for graphics. Generally speaking, for text boxes, the text is moved verbatim to and from the clipboard, and tables are moved to the clipboard as tab-delimited text.

```
Sub AppendToClipboard - Appends selected text to the Windows clipboard
    Get previous clipboard text
    Determine type of ActiveForm and ActiveControl
    Get text from ActiveControl according to type (text box, table, etc.)
    Add new text to previous text and put on clipboard
```

```
Sub CopyToClipboard - Copies text to the clipboard
    Depending on ActiveControl type, get text
    Put text on the clipboard
```

```
Sub CutToClipboard - Cuts text from control to the clipboard
    If not a text box, Beep and exit
    Cut text (delete) from active control and put on clipboard
```

```
Sub PasteFromClipboard - Paste text from the clipboard into control
    If not a text box, beep and exit
    Otherwise replace selected text with clipboard text
```

```
Sub SelectAllText - Select all text (or rows) in current control
    Depending on the ActiveControl type, highlight/select all text items
```

```
Sub ShowClipboard - Display the contents of the clipboard
    Determine clipboard contents
    Display contents in frmExpand window
```

4.4.8 Miscellaneous Utility Routines (UTILITY.BAS)

This file contains many of the generic routines which do not fit into one of the other categories, or are used in many different places in the software. These are typically small functions or routines which perform various "utility" processes.

Sub CreateET - Creates an element type table in the database if does not exist

```
Create the new table definition
Add all of the attributes to the table definition
Add the indexes to the table definition
Add the new table definition to the database
```

Sub CreateRT - Create a relation table in the database if it does not exist

```
Create the new table definition
Add the table fields to the new table
Add the new indexes to the new table
Add the new table definition to the database
```

Sub CreateTempTables - Creates the temporary database tables (for searches)

```
Create the temporary database file XXTEMPXX.MDB
Add the TwoTableQuery results table
Add the Multiplequery results table
Add the ArchReport architecture report table
```

Sub DeleteTempTables - Deletes the temporary database tables

```
Close the temporary database
Delete the temporary database file
```

Function Even - Returns true if an integer is even (false if odd)

```
If X mod 2 = 0, return True else return False
```

Function FileExists - Determines if a file exists

```
Get attributes of file
If error, then return False, else return True
```

Sub FillCVAs - Fills in the Custom Value Attributes

```
For all of the base attributes, add the attribute names
For each of the additional attributes, add the attribute name and type
```

Sub Get Help - Shows an on-line help topic by number

```
Call the built-in WinHelp routine
```

Function GetDateString - Returns a date attribute as a string or empty if null

```
Get the date field from the selected table
If error, return "", else return the string value of the date
```

Function GetInt - Returns a short integer field from a table

```
Get the integer field from the selected table
If error, return zero, else return the value
```

Function GetLongInt - Returns a long integer field from a table

```
Get the long integer field from the selected table
If an error, return zero, else return the value
```

Function GetNewFile - Get a new file name to open from the user

```
    Display a "file open" dialogue box to user
    If canceled, return "", else, return the new file name
```

Function GetString - Get a string (or memo) field from a table

```
    Get the string field value
    If error, return "", else return the value
```

Function GetTheString - Another version of **GetString** to return a string field

```
    Get the string field value
    If error, return "", else return the value
```

Function GetTimeString - Return a time string from a table attribute

```
    Get the time value from the field of the selected table
    If error, return "", else return the time string
```

Function GetUserInfo - Get the user's information from their .INI file

```
    If the file does not exist, get user info  [NewUser window]
    Read all of the input parameters
    If a custom view defined, load it  [FillCVAs, SetCVAs]
```

Function GotoElementRecord - Finds/goes to a specified element record

```
    Depending on the current index, search by name or number and name
    If not found, then display a warning message and return False
    Otherwise return True
```

Function GotoERQNoCase - Goes to an element record and ignores case changes

```
    Depending on the current index, search by name or number and name
    If not found, then return False
    Otherwise return True
```

Function GotoERQuiet - Goes to a specified element record without error msg

```
    Depending on the current index, search by name or number and name
    If not found, then return False
    Otherwise return True
```

Sub IRGColor - Set the color of a RelationsGrid row in the **ItemEdit** window

```
    For each column of the selected row
        Make a picture of the text
        Place picture in the image of the row
```

Sub LoadCustomView - Loads a custom view file

```
    Open the .CVF file
    For all custom attributes, get the value
    Close file and exit
```

Sub MGColor - Sets the color of an ElementGrid row in the **MainWindow**

```
    Select current row into block mode and change the color
```

Sub MGDelete - Deletes an ElementGrid row in the **MainWindow**

```
    Find the appropriate grid row, if not found, exit
    Select the row and delete
```

```
Sub MGInsert - Insert an ElementGrid row in the MainWindow
     Determine new location for insert
     Insert the new row

Sub MGNewColor - Change the color of an ElementGrid row in the MainWindow
     If new color is not blue or new color is blue and old was black then
          Change color of the row  [MGColor]

Sub MoveServer - Tells the user the server has been moved (no longer used)

Function RepairTheDatabase - Repairs the current database file
     Query user to continue
     Close the current database file
     Repair the current database file
     If error, print warning message and return False
     Reopen the current database and return True

Sub SaveCurrentView - Save the current view parameters into a file
     Open the current view file
     Output all of the custom view attribute settings
     Close the custom view file

Sub SaveCustomView - Save a current view into a file
     Open the current view file
     Output all of the custom view attribute settings
     Close the custom view file

Sub SaveUserInfo - Save the user's preferences in the .INI file
     Open the .INI file
     Write all parameters to the .INI file
     Close the .INI file

Sub SetCVAs - Set the "Show" flag for all attribute fields
     For all attributes in the database
     Set the "Show" value for the attribute

Sub SetLabels - Sets all of the program labels based on chosen nomenclature
     For all element types, set the element label
     For all relation types, set the relation labels

Sub SetSortOrder - Sets the sorting order for an element table
     Depending on the sort view, set the element table index

Sub ChnageDatabase - Opens a new database file
     If new database file does not exist, exit
     Close current database  [CloseTheDatabase]
     Unload all windows except main
     Open the new database  [OpenTheDatabase]
     Set the labels  [SetLabels]
     Set indexes  [SetSortOrder]
     Refill MainWindow grid  [UpdateMainGrid]

Sub UpdateUpdate - Updates the Auto-Update program  (No longer used)
```

4.4.9 Redlining routines (REDLINE.BAS)

These routines are used for "redlining" text when doing a comparison between previous and current versions of text attributes. This window is displayed when two historical values are chosen from the **SelectCompare** window. The following are the routines which are defined in this file.

Sub CompareText - Compares and "redlines" two text values

```
    Determine the redline text  [RedLine]
    Fill the window with the input and reline information
```

Function EOSentence - Determines if word separators are an end of sentence

```
    If string in is empty, return False
    If string contains a period, exclamation, or question mark, return True
    Otherwise return False
```

Function FindRenameAttr - Find new attribute value for a renamed element

```
    If no rename event found, return "-NOT FOUND-"
    For each previous rename event seek the element value
    If found, return the attribute value, otherwise return "-NOT FOUND-"
```

Function fPrintString - Prints a string to the printer with indent, 75 chars

```
    If line will fit, print it and return
    Otherwise, keep splitting string at 75 chars or less (at spaces)
    Print leftovers and return
```

Sub GetCompare - Initialize and display the **SelectCompare** window

```
    Set SelectCompare window captions and values
    For each memo or string attribute
        Add the attribute name to the attribute list box
    Display the SelectCompare window
```

Sub PrintDeltas - Prints the "redlined" comparison results to the printer

```
    Print initial label identifying comparison
    Print the old label and old text
    Print the new label and new text
    Print out the delta label and delta text by substituting the
        strikethrough and underline fonts for text markers
```

Function RedJunk - Returns the "redlined" difference between word delimiters

```
    If old junk and new junk are the same, return junk
    If old junk is empty, return inserted new junk
    If new junk is empty, return deleted old junk
    Otherwise, determine differences and return redline
```

Function RedLine - Returns the "redlined" difference between two strings

```
If the two strings are equal, return them
Divide the two strings into "words" and "junk" (anything between words)
For each word in the new string
    If matches current word in old string, increment both counters
    If just changed case of words, delete old and insert new
    Otherwise determine closest matching words in both strings
    Delete old stuff and insert new stuff
Delete any old leftovers, and insert any new leftovers
Return the redline results
```

Sub ShowChangeHistory - Opens the change history file in **HistoryWindow**

```
If baseline file does not exist, print warning message
If change history file does not exist, print warning message and exit
Open the change history database and display the HistoryWindow
```

4.4.10 Additional search query routines (QUERY.BAS)

These are routines to support the **MultiSearch** and **frmSingleQuery** windows. Originally written by Nancy Graves.

Function WhereSeg - Creates a SQL string for a search operation

```
Check for blanks (if exist, exit with null)
Check for valid value types
Create SQL string depending on search operation
```

This page intentionally left blank.

5.0 REFERENCES

WHC-CM-3-10, *Software Practices*, Revision 0, Appendix H, "System Design Description," Westinghouse Hanford Company, January 31, 1993.

WHC-SD-WM-CSCM-034, *TWRS Systems Engineering Software Configuration Management Plan*, Draft, dated September 1996.

This page intentionally left blank.

APPENDIX A

SUBROUTINE AND FUNCTION LIST

The table below identifies all of the routines and functions used within the RMSB software. The routines are ordered first by the name of the routine, and then by the file name where the routine is found. For more information on each of the routines, refer to the appropriate file section previously mentioned, or the actual source code.

File Name	Type	Routine Name
editing.bas	Function	AddElement () As Integer
editing.bas	Sub	AddOldView (t As Integer, n As String, num As String)
editing.bas	Sub	AddRel (WhichRel As Integer, SType As Integer, SName As String, SNumber As String, RType As Integer, RName As String, RNumber As String)
editing.bas	Function	AddTheRelations () As Integer
search.bas	Sub	AddToSearch (TheOp As String)
historyw.frm	Sub	AGrid_DblClick ()
editmenu.bas	Sub	AppendToClipboard ()
export.bas	Sub	ASCIIChanges ()
export.bas	Sub	ASCIIHistory ()
itemedit.frm	Sub	AttributeLabel_DblClick (Index As Integer)
itemedit.frm	Sub	AttributeList Click (Index As Integer)
itemedit.frm	Sub	AttributeMemo_Change (Index As Integer)
itemedit.frm	Sub	AttributeValue Change (Index As Integer)
searchwi.frm	Sub	AttrList_Click ()
selectco.frm	Sub	AttrList Click ()
multsrch.frm	Sub	AttrList Click (Index As Integer)
multsrch.frm	Sub	BPValueText_Change (Index As Integer)
multsrch.frm	Sub	BRValueText_Change (Index As Integer)
customvi.frm	Sub	CancelButton Click ()
editing.bas	Function	CanSave () As Integer
frmsqry.frm	Sub	cboOperator Click (Index As Integer)
utility.bas	Sub	ChangeDatabase (TFN As String)
editing.bas	Sub	ChangedAttribute (EType As Integer, WATT As Integer, NewValue As String)
changewi.frm	Sub	ChangeGrid DblClick (Col As Long, Row As Long)
search.bas	Function	CheckField (FieldType As Integer, Fieldstring As String) As Integer
multsrch.frm	Sub	ClearGrid ()
initstru.bas	Sub	Close4Export ()
merge.bas	Sub	CloseMergeFile ()
initstru.bas	Sub	CloseTheDatabase ()

File Name	Type	Routine Name
multsrch.frm	Sub	cmdAND Click ()
frmsqry.frm	Sub	cmdClose_Click ()
frmsqry.frm	Sub	cmdEditRecord Click ()
frmsqry.frm	Sub	cmdExecute Click ()
frmelsel.frm	Sub	cmdOK_Click ()
frmexpan.frm	Sub	cmdOK Click ()
listrela.frm	Sub	cmdOK_Click ()
multsrch.frm	Sub	cmdOR_Click ()
frmsqry.frm	Sub	cmdReset Click ()
multsrch.frm	Sub	cmdReset_Click ()
multsrch.frm	Sub	cmdReturn Click ()
multsrch.frm	Sub	cmdSearch Click ()
editing.bas	Sub	ColorRList (ET As Integer, EN As String)
addrelat.frm	Sub	Command1 Click ()
changewi.frm	Sub	Command1_Click ()
frmelsel.frm	Sub	Command1 Click ()
getnewpa.frm	Sub	Command1 Click ()
getreaso.frm	Sub	Command1_Click ()
inheritr.frm	Sub	Command1_Click ()
mainwind.frm	Sub	Command1 Click ()
newuser.frm	Sub	Command1_Click ()
oldviews.frm	Sub	Command1 Click ()
prefwind.frm	Sub	Command1_Click ()
printdes.frm	Sub	Command1 Click ()
searchwi.frm	Sub	Command1 Click ()
welcome.frm	Sub	Command1_Click ()
addrelat.frm	Sub	Command2 Click ()
getnewpa.frm	Sub	Command2 Click ()
getreaso.frm	Sub	Command2_Click ()
inheritr.frm	Sub	Command2 Click ()
mainwind.frm	Sub	Command2_Click ()
newuser.frm	Sub	Command2_Click ()
oldviews.frm	Sub	Command2 Click ()
prefwind.frm	Sub	Command2_Click ()
printdes.frm	Sub	Command2 Click ()
searchwi.frm	Sub	Command2 Click ()
welcome.frm	Sub	Command2_Click ()

File Name	Type	Routine Name
mainwind.frm	Sub	Command3 Click ()
printdes.frm	Sub	Command3_Click ()
searchwi.frm	Sub	Command3_Click ()
mainwind.frm	Sub	Command4 Click ()
searchwi.frm	Sub	Command4_Click ()
mainwind.frm	Sub	Command5 Click ()
searchwi.frm	Sub	Command5_Click ()
searchwi.frm	Sub	Command6_Click ()
selectco.frm	Sub	CompareButton Click ()
redline.bas	Sub	CompareText (OldText As String, OldTitle As String, NewText As String, NewTitle As String, CompareLabel As String)
editmenu.bas	Sub	CopyToClipboard ()
editing.bas	Sub	CreateElement (EType As Integer, newname As String)
utility.bas	Sub	CreateET (i As Integer)
utility.bas	Sub	CreateRT (i As Integer)
utility.bas	Sub	CreateTempTables ()
editmenu.bas	Sub	CutToClipboard ()
merge.bas	Sub	DandRElements (AttributeChanges As Table)
export.bas	Sub	DBReport (TFN As String)
merge.bas	Sub	DelACs (StartRec As Integer, EndRec As Integer, QueryUser As Integer)
merge.bas	Sub	DelECs (StartRec As Integer, EndRec As Integer, QueryUser As Integer, AttributeChanges As Table)
editing.bas	Sub	DeleteElement (EType As Integer, TheName As String, TheNum As String)
editing.bas	Sub	DeleteRelation (TheRel As Integer, SType As Integer, SName As String, SNumber As String, RType As Integer, RName As String, RNumber As String)
utility.bas	Sub	DeleteTempTables ()
merge.bas	Sub	DelRCs (StartRec As Integer, EndRec As Integer, QueryUser As Integer)
customvi.frm	Sub	DisplayCVAs ()
search.bas	Sub	DoTheSearch ()
historyw.frm	Sub	EGrid DblClick ()
mainwind.frm	Sub	ElementGrid DblClick (Col As Long, Row As Long)
mainwind.frm	Sub	ElementList Click ()
editing.bas	Sub	ElementRename (EType As Integer, OldName As String, OldNumber As String)
editing.bas	Sub	ElementRenumber (EType As Integer, OldNumber As String)
multsrch.frm	Sub	ElementType_Click (Index As Integer)
redline.bas	Function	EOSentence (InString As String) As Integer
utility.bas	Function	Even (x As Integer) As Integer
export.bas	Sub	ExportMatrix (TheFileName As String)
export.bas	Sub	ExportRMSB ()

File Name	Type	Routine Name
changewi.frm	Sub	FACS ()
changewi.frm	Sub	FCRs (NewFileName As String)
changewi.frm	Sub	FECS ()
utility.bas	Function	FileExists (TheFile As String) As Integer
editing.bas	Sub	FileRecord (TheFileName As String)
editing.bas	Sub	FileString (thestring As String, fl As Integer, Indent As Integer)
arecord.frm	Sub	FillAStuff ()
utility.bas	Sub	FillCVAs ()
editing.bas	Sub	FillElement (EType As Integer)
erecord.frm	Sub	FillESTuff ()
editing.bas	Sub	FillFullRG (EType As Integer)
multsrch.frm	Sub	FillGrid ()
historyw.frm	Sub	FillHAGrid ()
harecord.frm	Sub	FillHAStuff ()
historyw.frm	Sub	FillHEGrid ()
herecord.frm	Sub	FillHESTuff ()
historyw.frm	Sub	FillHRGrid ()
hrrecord.frm	Sub	FillHRStuff ()
editing.bas	Sub	FillIERelGrid (EType As Integer)
rrecord.frm	Sub	FillRStuff ()
merge.bas	Function	FindElement (EType As Integer, EName As String) As String
redline.bas	Function	FindRenameAttr (T As Table, WA As Integer) As String
listrela.frm	Sub	Form_Activate ()
mainwind.frm	Sub	Form LinkError (LinkErr As Integer)
mainwind.frm	Sub	Form LinkExecute (CmdStr As String, Cancel As Integer)
addrelat.frm	Sub	Form_Load ()
arecord.frm	Sub	Form_Load ()
changewi.frm	Sub	Form_Load ()
compare.frm	Sub	Form_Load ()
customvi.frm	Sub	Form_Load ()
erecord.frm	Sub	Form_Load ()
frmelsel.frm	Sub	Form_Load ()
frmexpan.frm	Sub	Form_Load ()
frmsqry.frm	Sub	Form_Load ()
getnewpa.frm	Sub	Form_Load ()
getreaso.frm	Sub	Form_Load ()
harecord.frm	Sub	Form_Load ()

File Name	Type	Routine Name
herecord.frm	Sub	Form_Load ()
historyw.frm	Sub	Form_Load ()
hrrecord.frm	Sub	Form_Load ()
inheritr.frm	Sub	Form_Load ()
itemedit.frm	Sub	Form_Load ()
listrela.frm	Sub	Form_Load ()
mainwind.frm	Sub	Form_Load ()
multsrch.frm	Sub	Form_Load ()
newuser.frm	Sub	Form_Load ()
oldviews.frm	Sub	Form_Load ()
prefwind.frm	Sub	Form_Load ()
printdes.frm	Sub	Form_Load ()
reltable.frm	Sub	Form_Load ()
rrecord.frm	Sub	Form_Load ()
searchwi.frm	Sub	Form_Load ()
selectco.frm	Sub	Form_Load ()
welcome.frm	Sub	Form_Load ()
addrelat.frm	Sub	Form_Resize ()
changewi.frm	Sub	Form_Resize ()
compare.frm	Sub	Form_Resize ()
historyw.frm	Sub	Form_Resize ()
inheritr.frm	Sub	Form_Resize ()
mainwind.frm	Sub	Form_Resize ()
oldviews.frm	Sub	Form_Resize ()
reltable.frm	Sub	Form_Resize ()
searchwi.frm	Sub	Form_Resize ()
redline.bas	Function	fPrintString (thestring As String, fl As Integer, Indent As Integer) As Integer
changewi.frm	Sub	FRCS ()
redline.bas	Sub	GetCompare ()
utility.bas	Function	GetDateString (TheTable As Table, thefield As String) As String
initial.bas	Function	GetDSetRelation (TheDS As Dynaset, FieldName As String) As String
initial.bas	Function	GetDSI (TheDS As Dynaset, FieldName As String) As Integer
initial.bas	Function	GetElement (theindex As Integer, FieldName As String) As String
utility.bas	Function	GetInt (TheTable As Table, thefield As String) As Integer
initial.bas	Function	GetIR (theindex As Integer, FieldName As String) As Integer
utility.bas	Function	GetLongInt (TheTable As Table, thefield As String) As Long
utility.bas	Function	GetNewFile (Title As String, Ext As String, Filter As String) As String

File Name	Type	Routine Name
initial.bas	Function	GetRelation (theindex As Integer, FieldName As String) As String
utility.bas	Function	GetString (TheTable As Table, thefield As String) As String
utility.bas	Function	GetTheString (TheTable As Table, EType As Integer, TheAttribute As Integer) As String
utility.bas	Function	GetTimeString (TheTable As Table, thefield As String) As String
utility.bas	Sub	GetUserInfo ()
utility.bas	Sub	Get_Help (HelpTopic As Long)
utility.bas	Function	GotoElementRecord (t As Integer, n As String, num As String) As Integer
utility.bas	Function	GotoERQNoCase (t As Integer, n As String, num As String)
utility.bas	Function	GotoERQuiet (t As Integer, n As String, num As String) As Integer
frmsqry.frm	Sub	Grid1_DblClick ()
itemedit.frm	Sub	HelpIEMenus_Click ()
merge.bas	Sub	IDCAttributes (AttributeChanges As Table)
merge.bas	Sub	IDCCreate ()
merge.bas	Sub	IDCDelete ()
merge.bas	Sub	IDCElements ()
merge.bas	Sub	IDConflicts (AttributeChanges As Table)
merge.bas	Sub	IDCRelations ()
merge.bas	Sub	IDCRename ()
itemedit.frm	Sub	IECancel_Click ()
arecord.frm	Sub	IEFirst_Click ()
erecord.frm	Sub	IEFirst_Click ()
harecord.frm	Sub	IEFirst_Click ()
herecord.frm	Sub	IEFirst_Click ()
hrrecord.frm	Sub	IEFirst_Click ()
itemedit.frm	Sub	IEFirst_Click ()
rrecord.frm	Sub	IEFirst_Click ()
arecord.frm	Sub	IELast_Click ()
erecord.frm	Sub	IELast_Click ()
harecord.frm	Sub	IELast_Click ()
herecord.frm	Sub	IELast_Click ()
hrrecord.frm	Sub	IELast_Click ()
itemedit.frm	Sub	IELast_Click ()
rrecord.frm	Sub	IELast_Click ()
arecord.frm	Sub	IENext_Click ()
erecord.frm	Sub	IENext_Click ()
harecord.frm	Sub	IENext_Click ()
herecord.frm	Sub	IENext_Click ()

File Name	Type	Routine Name
hrrecord.frm	Sub	IENext_Click ()
itemedit.frm	Sub	IENext_Click ()
rrecord.frm	Sub	IENext_Click ()
arecord.frm	Sub	IEOldViews_Click ()
erecord.frm	Sub	IEOldViews_Click ()
itemedit.frm	Sub	IEOldViews_Click ()
rrecord.frm	Sub	IEOldViews_Click ()
arecord.frm	Sub	IEPrevious_Click ()
erecord.frm	Sub	IEPrevious_Click ()
harecord.frm	Sub	IEPrevious_Click ()
herecord.frm	Sub	IEPrevious_Click ()
hrrecord.frm	Sub	IEPrevious_Click ()
itemedit.frm	Sub	IEPrevious_Click ()
rrecord.frm	Sub	IEPrevious_Click ()
changewi.frm	Sub	IEPrint_Click ()
itemedit.frm	Sub	IEPrint_Click ()
mainwind.frm	Sub	IEPrint_Click ()
oldviews.frm	Sub	IEPrint_Click ()
reltable.frm	Sub	IEPrint_Click ()
searchwi.frm	Sub	IEPrint_Click ()
arecord.frm	Sub	IEReturn_Click ()
erecord.frm	Sub	IEReturn_Click ()
harecord.frm	Sub	IEReturn_Click ()
herecord.frm	Sub	IEReturn_Click ()
hrrecord.frm	Sub	IEReturn_Click ()
itemedit.frm	Sub	IEReturn_Click ()
rrecord.frm	Sub	IEReturn_Click ()
merge.bas	Sub	IncAttributes ()
merge.bas	Sub	IncConflicts ()
merge.bas	Sub	IncElements ()
merge.bas	Sub	IncorpAttributes (QueryUser As Integer)
merge.bas	Sub	IncorpCreate (QueryUser As Integer)
merge.bas	Sub	IncorpDelete (QueryUser As Integer)
merge.bas	Sub	IncorpElements (QueryUser As Integer)
merge.bas	Sub	IncorporateAll (QueryUser As Integer)
merge.bas	Sub	IncorpRelations (QueryUser As Integer)
merge.bas	Sub	IncorpRename (QueryUser As Integer)

File Name	Type	Routine Name
merge.bas	Sub	IncRelations ()
editing.bas	Sub	InheritRels (WhichRel As Integer, SType As Integer, SName As String, SNumber As String, RType As Integer, RName As String, RNumber As String, TheRType As Integer)
export.bas	Sub	InheritReqs (TheFileName As String)
editing.bas	Function	InheritTheRelations ()
initial.bas	Sub	InitProgram ()
initial.bas	Sub	InitRelGrid ()
initstru.bas	Sub	InitStructure ()
utility.bas	Sub	IRGColor (TheRow As Integer, TheColor As Long)
editing.bas	Function	IsValidNumberField (TheNumber As String) As Integer
editing.bas	Function	IsValidSequence (seq As String) As Integer
arecord.frm	Sub	JField_Change ()
erecord.frm	Sub	JField_Change ()
harecord.frm	Sub	JField_Change ()
herecord.frm	Sub	JField_Change ()
hrrecord.frm	Sub	JField_Change ()
rrecord.frm	Sub	JField_Change ()
itemedit.frm	Sub	List1_Click ()
prefwind.frm	Sub	List1_Click ()
frmsqry.frm	Sub	LoadControls ()
utility.bas	Sub	LoadCustomView (TFN As String)
multsrch.frm	Sub	MakeSQL ()
frmsqry.frm	Function	MakeSQL () As Integer
merge.bas	Sub	MergeAppendAll ()
merge.bas	Sub	MergeAttributes (Query As Integer, AttributeChanges As Table)
merge.bas	Sub	MergeCreate (Query As Integer, AttributeChanges As Table)
merge.bas	Sub	MergeDelete (Query As Integer, AttributeChanges As Table)
merge.bas	Sub	MergeElements (Query As Integer)
merge.bas	Sub	MergeFiles (Query As Integer, AttributeChanges As Table)
merge.bas	Sub	MergeRelations (Query As Integer)
merge.bas	Sub	MergeRename (Query As Integer, AttributeChanges As Table)
utility.bas	Sub	MGColor (TheRow As Integer, TheColor As Long)
utility.bas	Sub	MGDelete (whichname As String)
utility.bas	Sub	MGInsert (whichname As String, whichnum As String)
utility.bas	Sub	MGNewColor (whichname As String, NewColor As Long)
addrelat.frm	Sub	Mh3dButn1_Click (Index As Integer)
changewi.frm	Sub	Mh3dButn1_Click (Index As Integer)

File Name	Type	Routine Name
inheritr.frm	Sub	Mh3dButn1_Click (Index As Integer)
mainwind.frm	Sub	Mh3dButn1_Click (Index As Integer)
reltable.frm	Sub	Mh3dButn1_Click (Index As Integer)
mainwind.frm	Sub	mnuAbout_Click ()
arecord.frm	Sub	mnuACCon_Click ()
arecord.frm	Sub	mnuACCopy_Click ()
arecord.frm	Sub	mnuACCut_Click ()
arecord.frm	Sub	mnuACDC_Click ()
arecord.frm	Sub	mnuACEM_Click ()
arecord.frm	Sub	mnuACGI_Click ()
arecord.frm	Sub	mnuACPaste_Click ()
arecord.frm	Sub	mnuACRAppend_Click ()
arecord.frm	Sub	mnuACSAll_Click ()
arecord.frm	Sub	mnuACTW_Click ()
arecord.frm	Sub	mnuACUH_Click ()
itemedit.frm	Sub	mnuAddRelation_Click ()
itemedit.frm	Sub	mnuArchRept_Click ()
changewi.frm	Sub	mnuCCCon_Click ()
changewi.frm	Sub	mnuCCDAppend_Click ()
changewi.frm	Sub	mnuCCEM_Click ()
changewi.frm	Sub	mnuCCGI_Click ()
changewi.frm	Sub	mnuCCTW_Click ()
changewi.frm	Sub	mnuCCUH_Click ()
reltable.frm	Sub	mnuCGI_Click ()
historyw.frm	Sub	mnuCHWAppend_Click ()
historyw.frm	Sub	mnuCHWCopy_Click ()
historyw.frm	Sub	mnuCHWCut_Click ()
historyw.frm	Sub	mnuCHWDC_Click ()
historyw.frm	Sub	mnuCHWHelpCon_Click ()
historyw.frm	Sub	mnuCHWHelpGI_Click ()
historyw.frm	Sub	mnuCHWHelpTW_Click ()
historyw.frm	Sub	mnuCHWHelpUH_Click ()
historyw.frm	Sub	mnuCHWPaste_Click ()
historyw.frm	Sub	mnuCHWSA_Click ()
itemedit.frm	Sub	mnuCopy_Click ()
customvi.frm	Sub	mnuCVCon_Click ()
customvi.frm	Sub	mnuCVGInfo_Click ()

File Name	Type	Routine Name
customvi.frm	Sub	mnuCVOpen Click ()
customvi.frm	Sub	mnuCVSaveAs_Click ()
customvi.frm	Sub	mnuCVSave_Click ()
customvi.frm	Sub	mnuCVUHelp Click ()
changewi.frm	Sub	mnuCWCopy_Click ()
changewi.frm	Sub	mnuCWCut Click ()
changewi.frm	Sub	mnuCWDC_Click ()
changewi.frm	Sub	mnuCWPaste Click ()
changewi.frm	Sub	mnuCWSelectAll Click ()
changewi.frm	Sub	mnuCWSortChange_Click ()
changewi.frm	Sub	mnuCWSortElement_Click ()
changewi.frm	Sub	mnuCWSortTarget Click ()
changewi.frm	Sub	mnuCWSortTime_Click ()
mainwind.frm	Sub	mnuDataVersion Click ()
mainwind.frm	Sub	mnuDBReport_Click ()
itemedit.frm	Sub	mnuDeleteRelation_Click ()
erecord.frm	Sub	mnuECCon Click ()
erecord.frm	Sub	mnuECCopy_Click ()
erecord.frm	Sub	mnuECCut_Click ()
erecord.frm	Sub	mnuECDC Click ()
erecord.frm	Sub	mnuECEM_Click ()
erecord.frm	Sub	mnuECGI Click ()
erecord.frm	Sub	mnuECPaste_Click ()
erecord.frm	Sub	mnuECRAppend Click ()
erecord.frm	Sub	mnuECSAll Click ()
erecord.frm	Sub	mnuECTW_Click ()
erecord.frm	Sub	mnuECUH Click ()
frmsqry.frm	Sub	mnuExit Click ()
mainwind.frm	Sub	mnuExit_Click ()
multsrch.frm	Sub	mnuExit Click ()
mainwind.frm	Sub	mnuExportRMSB_Click ()
mainwind.frm	Sub	mnuFECtoText_Click ()
mainwind.frm	Sub	mnuFileOpen Click ()
mainwind.frm	Sub	mnuFRMatrix_Click ()
frmsqry.frm	Sub	mnuFSQCSET Click ()
harecord.frm	Sub	mnuHACCon Click ()
harecord.frm	Sub	mnuHACCopy_Click ()

File Name	Type	Routine Name
harecord.frm	Sub	mnuHACCut Click ()
harecord.frm	Sub	mnuHACDC_Click ()
harecord.frm	Sub	mnuHACEM Click ()
harecord.frm	Sub	mnuHACGI Click ()
harecord.frm	Sub	mnuHACPaste_Click ()
harecord.frm	Sub	mnuHACRAppend Click ()
harecord.frm	Sub	mnuHACSAll_Click ()
harecord.frm	Sub	mnuHACTW_Click ()
harecord.frm	Sub	mnuHACUH Click ()
historyw.frm	Sub	mnuHCompare_Click ()
herecord.frm	Sub	mnuHECCon_Click ()
herecord.frm	Sub	mnuHECCopy Click ()
herecord.frm	Sub	mnuHECCut_Click ()
herecord.frm	Sub	mnuHECDC Click ()
herecord.frm	Sub	mnuHECEM_Click ()
herecord.frm	Sub	mnuHECGI Click ()
herecord.frm	Sub	mnuHECPaste Click ()
herecord.frm	Sub	mnuHECRAppend_Click ()
herecord.frm	Sub	mnuHECSAll Click ()
herecord.frm	Sub	mnuHECTW Click ()
herecord.frm	Sub	mnuHECUH_Click ()
mainwind.frm	Sub	mnuHelpMWButtons Click ()
mainwind.frm	Sub	mnuHelpMWMenus_Click ()
addrelat.frm	Sub	mnuHelp Click ()
inheritr.frm	Sub	mnuHelp Click ()
hrrecord.frm	Sub	mnuHRCCon_Click ()
hrrecord.frm	Sub	mnuHRCCopy Click ()
hrrecord.frm	Sub	mnuHRCCut Click ()
hrrecord.frm	Sub	mnuHRCDC_Click ()
hrrecord.frm	Sub	mnuHRCEM Click ()
hrrecord.frm	Sub	mnuHRCGI_Click ()
hrrecord.frm	Sub	mnuHRCPaste Click ()
hrrecord.frm	Sub	mnuHRCRAppend Click ()
hrrecord.frm	Sub	mnuHRCSAll_Click ()
hrrecord.frm	Sub	mnuHRCTW_Click ()
hrrecord.frm	Sub	mnuHRCUH Click ()
itemedit.frm	Sub	mnuIEAppend_Click ()

File Name	Type	Routine Name
itemedit.frm	Sub	mnuIEAtts Click ()
itemedit.frm	Sub	mnuIEB_Click ()
itemedit.frm	Sub	mnuIECopy Click ()
itemedit.frm	Sub	mnuIECut Click ()
itemedit.frm	Sub	mnuIEC_Click ()
itemedit.frm	Sub	mnuIEDC Click ()
itemedit.frm	Sub	mnuIEEls_Click ()
itemedit.frm	Sub	mnuIEEM_Click ()
itemedit.frm	Sub	mnuIEGI Click ()
itemedit.frm	Sub	mnuIEGoto_Click ()
itemedit.frm	Sub	mnuIEInherit_Click ()
itemedit.frm	Sub	mnuIEMQuery Click ()
itemedit.frm	Sub	mnuIEOldViews_Click ()
itemedit.frm	Sub	mnuIEPaste Click ()
itemedit.frm	Sub	mnuIEQSingle_Click ()
itemedit.frm	Sub	mnuIERels Click ()
itemedit.frm	Sub	mnuIESelectAll Click ()
itemedit.frm	Sub	mnuIEThisAttr_Click ()
itemedit.frm	Sub	mnuIEThis_Click ()
itemedit.frm	Sub	mnuIETwoTable Click ()
itemedit.frm	Sub	mnuIETW_Click ()
itemedit.frm	Sub	mnuIEUH Click ()
itemedit.frm	Sub	mnuIEVCH_Click ()
mainwind.frm	Sub	mnuMainCopy Click ()
mainwind.frm	Sub	mnuMainCut Click ()
mainwind.frm	Sub	mnuMainDBStruct_Click ()
mainwind.frm	Sub	mnuMainDelete_Click ()
mainwind.frm	Sub	mnuMainErase Click ()
mainwind.frm	Sub	mnuMainNew_Click ()
mainwind.frm	Sub	mnuMainPaste Click ()
mainwind.frm	Sub	mnuMainSelectAll_Click ()
mainwind.frm	Sub	mnuMainThisElement Click ()
mainwind.frm	Sub	mnuMWAppend Click ()
mainwind.frm	Sub	mnuMWC_Click ()
mainwind.frm	Sub	mnuMWDC Click ()
mainwind.frm	Sub	mnuMWEM Click ()
mainwind.frm	Sub	mnuMWG_Click ()

File Name	Type	Routine Name
mainwind.frm	Sub	mnuMWMW Click ()
mainwind.frm	Sub	mnuMWQMultiple_Click ()
mainwind.frm	Sub	mnuMWQSingle Click ()
mainwind.frm	Sub	mnuMWUH Click ()
mainwind.frm	Sub	mnuMWVH_Click ()
itemedit.frm	Sub	mnuNewRecord Click ()
oldviews.frm	Sub	mnuOVAppend_Click ()
oldviews.frm	Sub	mnuOVCopy Click ()
oldviews.frm	Sub	mnuOVCut Click ()
oldviews.frm	Sub	mnuOVC_Click ()
oldviews.frm	Sub	mnuOVEM_Click ()
oldviews.frm	Sub	mnuOVG Click ()
oldviews.frm	Sub	mnuOVPaste_Click ()
oldviews.frm	Sub	mnuOVSelectAll Click ()
oldviews.frm	Sub	mnuOVTW_Click ()
oldviews.frm	Sub	mnuOVUH_Click ()
oldviews.frm	Sub	mnuPEVDC Click ()
prefwind.frm	Sub	mnuPrefHelp_Click ()
mainwind.frm	Sub	mnuPrefs_Click ()
frmsqry.frm	Sub	mnuPrint Click ()
multsrch.frm	Sub	mnuPrint_Click ()
frmsqry.frm	Sub	mnuQCopy Click ()
frmsqry.frm	Sub	mnuQCut_Click ()
frmsqry.frm	Sub	mnuQC_Click ()
frmsqry.frm	Sub	mnuQEM Click ()
frmsqry.frm	Sub	mnuQGI_Click ()
frmsqry.frm	Sub	mnuQPaste_Click ()
frmsqry.frm	Sub	mnuQSelectAll Click ()
frmsqry.frm	Sub	mnuQSW_Click ()
frmsqry.frm	Sub	mnuQUH Click ()
rrecord.frm	Sub	mnuRCCon_Click ()
rrecord.frm	Sub	mnuRCCopy Click ()
rrecord.frm	Sub	mnuRCCut Click ()
rrecord.frm	Sub	mnuRCDC Click ()
rrecord.frm	Sub	mnuRCEM Click ()
rrecord.frm	Sub	mnuRCGI Click ()
rrecord.frm	Sub	mnuRCPaste_Click ()

File Name	Type	Routine Name
rrecord.frm	Sub	mnuRCRAppend_Click ()
rrecord.frm	Sub	mnuRCSAll_Click ()
rrecord.frm	Sub	mnuRCTW_Click ()
rrecord.frm	Sub	mnuRCUH_Click ()
itemedit.frm	Sub	mnuRecordDelete_Click ()
compare.frm	Sub	mnuRedAppend_Click ()
compare.frm	Sub	mnuRedCopy_Click ()
compare.frm	Sub	mnuRedCut_Click ()
compare.frm	Sub	mnuRedDC_Click ()
compare.frm	Sub	mnuRedHelpCon_Click ()
compare.frm	Sub	mnuRedHelpGI_Click ()
compare.frm	Sub	mnuRedHelpTW_Click ()
compare.frm	Sub	mnuRedHelpUH_Click ()
compare.frm	Sub	mnuRedPaste_Click ()
compare.frm	Sub	mnuRedSA_Click ()
reltable.frm	Sub	mnuRelRels_Click ()
reltable.frm	Sub	mnuRelThisRel_Click ()
mainwind.frm	Sub	mnuRepair_Click ()
reltable.frm	Sub	mnuRTAppend_Click ()
reltable.frm	Sub	mnuRTCopy_Click ()
reltable.frm	Sub	mnuRTCut_Click ()
reltable.frm	Sub	mnuRTC_Click ()
reltable.frm	Sub	mnuRTDC_Click ()
reltable.frm	Sub	mnuRTEM_Click ()
reltable.frm	Sub	mnuRTPaste_Click ()
reltable.frm	Sub	mnuRTSelectAll_Click ()
reltable.frm	Sub	mnuRTTW_Click ()
reltable.frm	Sub	mnuRTUH_Click ()
selectco.frm	Sub	mnuSCWHelpCon_Click ()
selectco.frm	Sub	mnuSCWHelpGI_Click ()
selectco.frm	Sub	mnuSCWHelpTW_Click ()
selectco.frm	Sub	mnuSCWHelpUH_Click ()
searchwi.frm	Sub	mnuSearchCopy_Click ()
mainwind.frm	Sub	mnuShowChanges_Click ()
frmsqry.frm	Sub	mnuShowSQL_Click ()
multsrch.frm	Sub	mnuShowSQL_Click ()
frmsqry.frm	Sub	mnuSortName_Click ()

File Name	Type	Routine Name
multsrch.frm	Sub	mnuSortName_Click ()
frmsqry.frm	Sub	mnuSortNumber_Click ()
multsrch.frm	Sub	mnuSortNumber_Click ()
multsrch.frm	Sub	mnuSortRByName_Click ()
multsrch.frm	Sub	mnuSortRByNumber_Click ()
printdes.frm	Sub	mnuSPDH_Click ()
frmsqry.frm	Sub	mnuSQWDC_Click ()
frmsqry.frm	Sub	mnuSQYAppend_Click ()
searchwi.frm	Sub	mnuSrchDC_Click ()
mainwind.frm	Sub	mnuSupRDT_Click ()
searchwi.frm	Sub	mnuSWAppend_Click ()
multsrch.frm	Sub	mnuSWCopy_Click ()
searchwi.frm	Sub	mnuSWCopy_Click ()
searchwi.frm	Sub	mnuSWCST_Click ()
multsrch.frm	Sub	mnuSWCut_Click ()
searchwi.frm	Sub	mnuSWCut_Click ()
multsrch.frm	Sub	mnuSWC_Click ()
searchwi.frm	Sub	mnuSWC_Click ()
multsrch.frm	Sub	mnuSWEM_Click ()
searchwi.frm	Sub	mnuSWEM_Click ()
multsrch.frm	Sub	mnuSWGI_Click ()
searchwi.frm	Sub	mnuSWGI_Click ()
multsrch.frm	Sub	mnuSWPaste_Click ()
searchwi.frm	Sub	mnuSWPaste_Click ()
searchwi.frm	Sub	mnuSWSB_Click ()
searchwi.frm	Sub	mnuSWSC_Click ()
multsrch.frm	Sub	mnuSWSelectAll_Click ()
searchwi.frm	Sub	mnuSWSelectAll_Click ()
searchwi.frm	Sub	mnuSWSW_Click ()
multsrch.frm	Sub	mnuSWUH_Click ()
searchwi.frm	Sub	mnuSWUH_Click ()
multsrch.frm	Sub	mnuTTSAppend_Click ()
multsrch.frm	Sub	mnuTTSDC_Click ()
multsrch.frm	Sub	mnuTTSW_Click ()
mainwind.frm	Sub	mnuTwoTable_Click ()
newuser.frm	Sub	mnuUserHelp_Click ()
utility.bas	Sub	MoveServer ()

File Name	Type	Routine Name
printdes.frm	Sub	NewFileName Change ()
changewi.frm	Sub	NoCWButtons ()
customvi.frm	Sub	OKButton Click ()
reltable.frm	Sub	OldViews Click ()
initstru.bas	Sub	Open4Export (TFN As String)
merge.bas	Sub	OpenLog2 ()
merge.bas	Sub	OpenLogFile ()
merge.bas	Sub	OpenMergeFile ()
initstru.bas	Sub	OpenTheDatabase (TFN As String)
multsrch.frm	Sub	OpList Click (Index As Integer)
changewi.frm	Sub	Option1 Click ()
changewi.frm	Sub	Option2 Click ()
changewi.frm	Sub	Option3 Click ()
changewi.frm	Sub	PACS ()
addrelat.frm	Sub	PageLabel Click ()
inheritr.frm	Sub	PageLabel Click ()
changewi.frm	Sub	PageLabel DblClick ()
mainwind.frm	Sub	PageLabel DblClick ()
reltable.frm	Sub	PageLabel DblClick ()
editmenu.bas	Sub	PasteFromClipboard ()
changewi.frm	Sub	PCRs ()
changewi.frm	Sub	PECS ()
changewi.frm	Sub	PRCS ()
editing.bas	Sub	PrintAll ()
compare.frm	Sub	PrintButton Click ()
redline.bas	Sub	PrintDeltas (TheText As String)
editing.bas	Sub	PrintRecord ()
editing.bas	Sub	PrintString (thestring As String, fl As Integer, Indent As Integer)
mainwind.frm	Sub	QuitButton Click ()
reltable.frm	Sub	QuitB Click ()
export.bas	Sub	RDTChanges ()
export.bas	Sub	RDTHeader ()
export.bas	Sub	RDTHistory ()
export.bas	Sub	RDTPreface ()
redline.bas	Function	RedJunk (NewJunk As String, OldJunk As String) As String
redline.bas	Function	RedLine (OldString As String, NewString As String) As String
changewi.frm	Sub	RefillACTable ()

File Name	Type	Routine Name
changewi.frm	Sub	RefillECTable ()
initial.bas	Sub	RefillMain (NumRows As Integer)
changewi.frm	Sub	RefillRCTable ()
reltable.frm	Sub	RelationGrid_DblClick (Col As Long, Row As Long)
reltable.frm	Sub	RelationList_Click ()
mainwind.frm	Sub	RelationsB_Click ()
itemedit.frm	Sub	RelationsGrid_DblClick () 'Col As Long, Row As Long)
searchwi.frm	Sub	RelAttrList_Click ()
itemedit.frm	Sub	REList_Click ()
searchwi.frm	Sub	RelTypeList_Click ()
editing.bas	Sub	RemoveRelation (TheRel As Integer, SType As Integer, SName As String, SNumber As String, RType As Integer, RName As String, RNumber As String)
utility.bas	Function	RepairTheDatabase () As Integer
multsrch.frm	Sub	ResultsGrid_DblClick ()
searchwi.frm	Sub	ResultsGrid_DblClick ()
compare.frm	Sub	ReturnButton_Click ()
historyw.frm	Sub	ReturnButton_Click ()
selectco.frm	Sub	ReturnButton_Click ()
historyw.frm	Sub	RGrid_DblClick ()
addrelat.frm	Sub	RList_Click ()
inheritr.frm	Sub	RList_Click ()
utility.bas	Sub	SaveCurrentView (TFN As String)
utility.bas	Sub	SaveCustomView (TFN As String)
utility.bas	Sub	SaveUserInfo ()
mainwind.frm	Sub	SearchButton_Click ()
editmenu.bas	Sub	SelectAllText ()
addrelat.frm	Sub	SetAVButtons ()
inheritr.frm	Sub	SetAVButtons ()
editing.bas	Sub	SetButtons (EType As Integer)
utility.bas	Sub	SetCVAs ()
changewi.frm	Sub	SetCWButtons ()
utility.bas	Sub	SetLabels (NewView As Integer)
initial.bas	Sub	SetRVButtons ()
utility.bas	Sub	SetSortOrder (i As Integer, NewOrder As Integer)
initial.bas	Sub	SetVButtons ()
redline.bas	Sub	ShowChangeHistory (TheType As Integer, TheName As String)
editmenu.bas	Sub	ShowClipboard ()
editing.bas	Sub	ShowElement (EType As Integer, ename As String, ENumber As String)

File Name	Type	Routine Name
mainwind.frm	Sub	SortButton Click ()
reltable.frm	Sub	SortB_Click ()
export.bas	Sub	ToASCIIText (TheFileName As String)
export.bas	Sub	ToSuperiorRDT (TheFileName As String)
arecord.frm	Sub	UpdateAChange ()
erecord.frm	Sub	UpdateEChange ()
editing.bas	Function	UpdateElement () As Integer
initial.bas	Sub	UpdateMainGrid ()
rrecord.frm	Sub	UpdateRChange ()
initial.bas	Sub	UpdateRelGrid ()
utility.bas	Sub	UpdateUpdate ()
multsrch.frm	Sub	ValueList Change (Index As Integer)
multsrch.frm	Sub	ValueText Change (Index As Integer)
oldviews.frm	Sub	ViewsGrid DblClick ()
prefwind.frm	Sub	VMode_Click ()
frmsqry.frm	Sub	vsMoveChild Change ()
addrelat.frm	Sub	VTList Click ()
inheritr.frm	Sub	VTList_Click ()
query.bas	Function	WhereSeg (AttrName As String, AttrType As Integer, OpType As Integer, value As String, value1 As String) As String

DISTRIBUTION SHEET

To Distribution	From SE Risk Management & Modeling	Page 1 of 1
		Date 9/26/96
Project Title/Work Order Requirements Management System Browser Software Design Description		EDT No. 618229
		ECN No.

Name	MSIN	Text With All Attach.	Text Only	Attach./ Appendix Only	EDT/ECN Only
<u>WHC</u>					
N. G. Awadalla	H6-35	X			
D. D. Frank	H6-35	X			
B. C. Gneiting	H6-35	X			
N. J. Graves	H6-35	X			
L. G. Peck	H6-35	X			
D. F. Salsman	H6-35	X			
J. M. Vann	H6-33	X			
<u>RL</u>					
D. J. Francis	K6-51	X			
<u>Other</u>					
Central Files (Original)	A3-88	X			
Log Book (D. D. Frank)	H6-35	X			