

**FRAMES Internet Database
Tools: Database Owner's Tool
(DOT), Data Extraction Tool
(DET), and Database Client
Editor (DCE)**

January 2001

Prepared for
U.S. Army Research and Development Center
Waterways Experiment Station
U.S. Army Corps of Engineers
Vicksburg, Mississippi 39180
under Contract DE-AC06-76RL01830

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.** 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 Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY

operated by

BATTELLE

for the

UNITED STATES DEPARTMENT OF ENERGY

under Contract DE-ACO6-76RLO1830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;

ph: (865) 576-8401

fax: (865) 576-5728

email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service,
U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161

ph: (800) 553-6847

fax: (703) 605-6900

email: orders@ntis.fedworld.gov

online ordering: <http://www.ntis.gov/ordering.htm>



This document was printed on recycled paper.

**FRAMES Internet Database Tools:
Database Owner's Tool (DOT), Data
Extraction Tool (DET), and Database
Client Editor (DCE)**

M.A. Pelton
R.Y. Taira
K.E. Dorow
F.C. Rutz
N.A. Reitz
G. Whelan

January 2001

Prepared for
U.S. Army Research and Development Center
Waterways Experiment Station
U.S. Army Corps of Engineers
Vicksburg, Mississippi 39180
under Contract DE-AC06-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Summary

The U.S. Army Research and Development Center, Waterways Experiment Station (WES), U.S. Corps of Engineers develops tools to help analysts assess the impacts of anthropogenic activities in the environment. As such, WES is developing the Army Risk Assessment Modeling System (ARAMS) to provide the Army with the capability to perform human and ecologically based risk/hazard assessment associated with past-practice and current activities at military installations. The intent of the system is to provide a platform from which a variety of assessments can be performed, using screening-level tools, science-support tools, site-specific databases, physicochemical databases, visualization, and Conceptual Site Model (CSM) guidance. The system is envisioned to help a risk analyst visualize an assessment from source, through multiple environmental media (e.g., groundwater, surface water, air, and land), to sensitive receptors of concern (e.g., humans and wildlife). Concurrently, WES is also sponsoring the development of the science-support-based Land-Based Management System (LMS), which has many of the same goals of ARAMS, yet is taking a more detailed approach to each of the components of an integrated system. The linkage of ARAMS and LMS is an ultimate goal of WES, such that the components in each system can be used by the other system to support assessment activities.

To help use current and existing state-of-the-art multimedia tools and to gain an appreciation for current technological advances, the Pacific Northwest National Laboratory (PNNL) is supporting WES and ARAMS by modifying and updating the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES) for inclusion as a component in ARAMS. FRAMES is a Windows-based software platform that provides an interactive user interface and, more importantly, specifications to allow a variety of DOS and Windows-based environmental codes to be integrated within a single framework. As a new component in ARAMS and FRAMES, PNNL has developed the FRAMES Internet Database Tools, which are software products that enable the FRAMES system to access data from disparate MS Access 97 or Access 2000 databases (i.e., databases with different structures and formats). These tools add a powerful functionality to the FRAMES system because they allow users to utilize databases that are potentially large and complicated without having to reformat them to fit the system. Additionally, the databases accessed can be located on any computer that is accessible via the Internet. The tools help users map their database structures to the systems structure so their data can be used. No modifications to the databases themselves are required.

The FRAMES Internet Database Tools are comprised of three components: the Database Owner's Tool (DOT), the Data Extraction Tool (DET), and the Database Client Editor (DCE). The DOT allows the database owner to specify the mapping between parameters needed by models in FRAMES and their locations in their database. The mapping specified is used to create the database extraction plan. The DET is the tool that actually implements the database extraction plan that is generated by the DOT. It is responsible for sending the data query from FRAMES to the database, extracting the pertinent data, and returning it to FRAMES for use by the requesting model. The DCE allows the user to specify what data are being requested from the database. The DCE makes calls to the DET and DOT to obtain the requested data and provide it for the requesting model.

This document provides requirements, design, data-file specifications, test plan, and Quality Assurance/Quality Control (QA/QC) protocol for the FRAMES Internet Database Tools. The requirements identify the attributes of the system. The design describes how the system will be structured to meet those requirements. The specification presents the specific modifications to FRAMES to meet the requirements

and design. The test plan ensures that the basic functionality listed in the requirements (black box testing) actually functions as designed, and QA/QC ensures that the software meets the client's needs.

Acknowledgments

The authors thank Mr. Wayne Cosby of the Pacific Northwest National Laboratory (PNNL) for editing this document. This work was supported by WES, as part of their ARAMS program. PNNL is operated for the U.S. Department of Energy by Battelle under Contract DE-AC06-76RLO 1830.

Acronyms and Abbreviations

ARAMS	Army Risk Assessment Modeling System
BBF	Body Burden File
CCC	Criterion Continuous Concentration
CSM	Conceptual Site Model, simplified description of the environmental problem to be modeled
DCE	Database Client Editor
DES	DESCRIPTION file, containing all necessary information to link a module into FRAMES
DET	Data Extraction Tool
DIC	Dictionary (file)
DOS	Disk Operation System, basic operation system on the computer
DOT	Database Owner's Tool
EBF	Ecological Benchmark File
ED	Effective Dose
EHQ	Ecological Hazard Quotient
EPA	U.S. Environmental Protection Agency
ERED	Environmental Residue-Effects Database
FRAMES	Framework for Risk Analysis in Multimedia Environmental Systems
FRANCO	FRequency ANalysis of COncentration
FUI	Framework User Interface
GID	Global Input Data File, contains all of the input data to run modules
LC	Lethal Concentration
LD	Lethal Dose
LMS	Land-based Management System
MUI	Model-Specific User Interface
NOAEL	No Observable Adverse Effects Level
NOEL	No Observable Effects Level
OCRWM	Office of Civilian Radioactive Waste Management
ODBC	Open Database Connectivity
PNNL	Pacific Northwest National Laboratory
QA/QC	Quality Assurance and Quality Control, processes that ensure the quality of the product
WCF	Water Concentration File, containing time-varying constituent concentrations
WEAP	Wildlife Ecological Assessment Program

WES

Waterways Experiment Station

Contents

Summary iii
Acknowledgments	v
Acronyms and Abbreviations	vii
1.0 Introduction	1.1
1.1 Objective	1.1
1.2 Definitions	1.1
2.0 Requirements	2.1
2.1 DOT	2.1
2.1.1 System Requirements	2.1
2.1.2 Input Requirements	2.1
2.1.3 Output Requirements	2.2
2.2 Database Extraction Tool (DET)	2.2
2.2.1 General Requirements	2.2
2.3 DCE	2.2
2.3.1 System Requirements	2.2
2.3.2 Input Requirements	2.2
2.3.3 Output Requirements	2.3
3.0 Design	3.1
3.1 DOT	3.1
3.1.1 System Design	3.1
3.1.2 Input Design	3.2
3.1.3 Output Design	3.4
3.2 DET	3.17
3.2.1 System Design	3.17
3.2.2 Input Design	3.18
3.2.3 Output Design	3.18
3.3 DCE	3.18
3.3.1 System Design	3.18
3.3.2 Input Design	3.18
3.3.3 Output Design	3.19
4.0 Specifications	4.1
4.1 DOT	4.1
4.1.1 Input Specifications	4.1

4.1.1.1 Dictionary File	4.1
4.1.2 Output Specifications	4.2
4.1.2.1 DOT Database	4.2
4.2 DET	4.5
4.2.1 Input Specifications	4.5
4.2.1.1 Format of Data Query	4.5
4.2.1.2 Data Query Example	4.5
4.2.2 Output Specifications	4.5
4.3 DCE	4.6
4.3.1 Input Specifications	4.6
4.3.1.1 DES File for DCE	4.6
4.3.2 Output Specifications	4.6
5.0 Testing Approach and Results	5.1
5.1 DOT	5.1
5.1.1 Summary of Requirements	5.1
5.1.2 TEST CASES	5.4
5.1.2.1 DOT_01	5.4
5.1.2.2 DOT_02	5.6
5.1.2.3 DOT_03	5.8
5.1.2.4 DOT_04	5.10
5.1.2.5 DOT_05	5.12
5.1.2.6 DOT_06	5.13
5.2 DET	5.15
5.2.1 Summary of Requirements	5.15
5.2.2 TEST CASES	5.16
5.2.2.1 DET_01	5.16
5.2.2.2 DET_02	5.17
5.2.2.3 DET_03	5.18
5.3 DCE	5.19
5.3.1 Summary of Requirements	5.19
5.3.2 TEST CASE	5.20
5.3.2.1 DCE_01	5.20
6.0 Quality Assurance Program	6.1
7.0 References	7.1
Appendix A: Ecological Benchmark File (EBF)	A.1

Figures

3.1.	System Overview	3.2
3.2.	Screen and Data-Flow Diagram	3.3
3.3.	Welcome Screen: This Is the First Screen That Appears When the URL for this Software Is Accessed	3.4
3.4.	Dictionary Parsing Tool: After the “Dictionary Parsing Tool” Is Selected in the Left Panel, a Screen Appears Requesting to Add or Delete a Dictionary	3.5
3.5.	New Dictionary Information: The Data Required to Connect to and Parse a Dictionary on the Local Network	3.6
3.6.	Dictionary Deleting Tool: After selecting “Delete a Dictionary,” this Screen Requests Which Dictionary to Delete	3.7
3.7.	Select Query Database: After the “Database Owners Tool” Is Selected in the Left Panel, a Screen Appears Requesting a Database to Connect to or Delete	3.8
3.8.	New Query Database: The Data Required to Connect to a Database on the Local Network	3.9
3.9.	Extraction Plan—Parameter Selection: The Extraction Plan That Will Be Built Is for the Parameter(s) Selected Here	3.10
3.10.	Extraction Plan—Table Selection: Database Tables to Use in the Query Are Selected Here	3.11
3.11.	Extraction Plan—Field Selection: Fields in the Table Are Selected and That Will Be the Data Returned from the Database	3.12
3.12.	Extraction Plan—Match Parameter with Database Field: Match Parameters with Fields in the Database	3.13
3.13.	Extraction Plan—Filter Selection: Creating the “Where Clause” for the Query	3.14
3.14.	Extraction Plan—Join Selection: Describing How the Two Tables Will Be Joined	3.15
3.15.	Extraction Plan—Results (Providing a Key Value): an Actual Value must Be Provided for Any Key Values Required	3.16
3.16.	Extraction Plan—Results (Display of Results): Displays the Data Found in the Tables That Match the Query	3.17
3.17.	DCE Initial Screen	3.20
3.18.	Chemical Aliases	3.21
3.19.	Lifeform Selection	3.22
3.20.	Lifeform Aliases	3.23
3.21.	Viewing and Editing a Parameter	3.24
6.1.	Ensuring Quality in the Environmental Software Development Process	6.2
6.2.	Quality Assurance Implementation Checklist for the FRAMES Technology Software System	6.4

Tables

4.1. Database	4.2
4.2. Dictionary	4.2
4.3. Extraction Plan	4.2
4.4. Extraction Plan Fields	4.3
4.5. Extraction Plan Join	4.3
4.6. Extraction Plan Static Filter	4.3
4.7. Extraction Plan Tables	4.3
4.8. Fields	4.3
4.9. KeyFilter	4.4
4.10. Parameter	4.4
4.11. Parameter Keys	4.4
4.12. Tables	4.5
5.1. Fundamental Requirements for Testing the DOT	5.1
5.2. Relationship Between Test Cases and DOT Fundamental Requirements	5.3
5.3. Expected Results DOT_02	5.7
5.4. Expected Results DOT_03	5.8
5.5. Filter Selection Data For DOT_03	5.9
5.6. Expected Results for Query Without “Unique” Selection	5.10
5.7. Expected Results For Query With ‘unique’ Selection	5.11
5.9. Fundamental Requirements for Testing the DET	5.15
5.10. Relationship Between Test Cases and DET Fundamental Requirements	5.15
5.11. Expected Results For Query with One Key Value	5.16
5.12. Expected Results For Query With Five Key Values	5.17
5.13. Expected Results for Five Key Query With Only Three Keys Specified	5.18
5.14. Fundamental Requirements for Testing the DCE	5.19
6.1. Relationship of Pacific Northwest National Laboratory (PNNL) Environmental Software Development Process to Quality Assurance Requirements	6.3

1.0 Introduction

1.1 Objective

Tools are being developed that will enable the FRAMES system to access data from MS Access 97 or Access 2000 databases regardless of table structure, formats, or even location. These tools should allow users to utilize databases that are potentially large and complicated without having to reformat them to fit the system. The FRAMES Internet Database Tools are comprised of three components: the Database Owner's Tool (DOT), the Data Extraction Tool (DET), and the Database Client Editor (DCE). The DOT allows the database owner to specify the mapping between parameters needed by models in FRAMES and their locations in their database. The mapping specified is used to create the database extraction plan. The DET is the tool that actually implements the database extraction plan that is generated by the DOT. It is responsible for sending the data query from FRAMES to the database, extracting the pertinent data, and returning it to FRAMES for use by the requesting model. The DCE allows the user to specify what data are being requested from the database. The DCE makes calls to the DET and DOT to obtain the requested data and provide it for the requesting model.

1.2 Definitions

- Requirement – Characteristics and behaviors that a piece of software must possess to function adequately for its intended purpose. A good requirement is also testable. Sometimes it is called an attribute.
- Design – A design is a comprehensive description of how a piece of software will function (i.e., how it will meet its requirements).
- Specifications – Specifications are a detailed description of an interface to a computer program or set of subroutines such that another programmer could develop a program that would use the subroutines properly.
- Model/Code – Loosely defined herein to represent the software product for simulating an aspect associated with the release, fate & transport, exposure, intake, or risk/hazard of chemicals released into the environment; however, a model/code can simulate any phenomena and is not limited to hazardous-waste-site assessments.
- Module – Model/Code and accompanying 1) pre- and post-processors for communicating with other models, databases, frameworks, etc. and 2) model-specific user interface (MUI).
- System or Framework – Loosely defines a grouping of models, modules, databases, processors, or a combination.
- Testable – Refers to the property of having the capability to examine and interrogate, such that a clear and concise conclusion can be drawn.

2.0 Requirements

The FRAMES Internet Database Tools package is comprised of three separate software products, which are the DOT, DET, and DCE. Each software product has its specific requirements that it must meet independent of the other components. For this reason, the requirements for each product will be listed separately in this section. Requirements are characteristics and behaviors that a piece of software must possess to function adequately for its intended purpose.

2.1 DOT

2.1.1 System Requirements

The DOT shall meet the following requirements when integrated into the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES) architecture:

- Connect to any MS ACCESS 97 or ACCESS 2000 database.
- Be accessible on a server via the Internet.
- Ask for user login credentials for security.
- Provide a web-page interface for creating stored execution plans.
- Provide a web-page interface for creating and storing an open database connectivity (ODBC) connection information for the source database. Provide a web-page interface for creating and storing Dictionary (DIC) connection information for FRAMES datasets.
- Provide the capability to test and modify extraction plans.
- Be usable by database owners and assist them in mapping the data from the source database into FRAMES dataset(s) outlined by the DIC file.
- Store database scheme (structure), DIC scheme, and extraction plans.
- Be capable of joining data from separate tables within a database when searching for requested data.
- Be capable of filtering out non-unique data when the option is selected through the DOT user interface.
- Operate within FRAMES under Windows-95 or higher.

2.1.2 Input Requirements

The DOT shall meet the following input requirements:

- Allow the user to enter data necessary to specify the name and location of a database.
- Allow the user choose the FRAMES dataset DIC to map to (Note: No checking is performed to ensure that the DIC is in the correct format).
- Read a DIC file to determine parameter list.
- Read a DIC file to determine parameter correlations.
- Allow the user to specify how parameters are mapped.

2.1.3 Output Requirements

The DOT shall meet the following outputs requirements:

- Create an extraction plan(s) in the form of an SQL statement with replaceable parameters (keys).
- Create an extraction plan(s) capable of retrieving a list.
- Create an extraction plan(s) capable of retrieving related data from multiple tables.
- Create stored extraction plan that may have one or more key values associated with it.
- Provide error messages when DIC or MDB files are not found.

2.2 Database Extraction Tool (DET)

2.2.1 General Requirements

- Connect to any MS ACCESS 97 or ACCESS 2000 database.
- Execute an extraction plan to retrieve data from the associated database.
- Format extracted data so that it is usable by FRAMES.
- Use key values that will be provided in requests for data and will act as filters for the extracted data. Keys are substituted in for the replaceable parameters by the DET when an extraction is executed.
- Operate within FRAMES under Windows-95 or higher.

2.3 DCE

2.3.1 System Requirements

DCE shall meet the following system requirements:

- Operate within FRAMES under Windows-95 or higher.
- Issue extraction queries to DET via HTTP.
- Extract all data from Eco Benchmark database via DET based on chemical and species alias selections.
- Display extracted data in user interface.
- Include a DES file for use in FRAMES.

2.3.2 Input Requirements

DCE shall meet the following input requirements:

- Read a Global Input Data (GID) file, specifically the CON section of a GID file, to determine chemicals of concern.
- Read the dataset dictionary file (.DIC) to determine dataset parameters to retrieve.
- Retrieve a list of lifeforms from the Eco Benchmark database via DET and display for user selection.

- Retrieve a list of chemicals from the Eco Benchmark database via DET and display for user alias selection.
- Allow the user to alias one lifeform to another.
- Allow the user to alias one chemical to another.
- Allow the user to edit the data retrieved.

2.3.3 Output Requirements

DCE shall meet the following output requirements:

- Write an ECO GID section that contains the names and values of the extracted parameters.
- Create an error file on start up.
- Delete the error if there are no errors.

3.0 Design

Design elements are strategies for meeting requirements. This section reviews the design associated with meeting the requirements outlined in Section 2.0. The design for each component will be described separately in this section.

3.1 DOT

3.1.1 System Design

The following specifies the hardware and software required to host the FRAMES DOT software:

Hardware:

- PC-server or high-end workstation LAN connection with a network path to the computer holding the ERED database (could be on the same server).

Software:

- Windows NT 4.0 Server or Workstation.
- Windows NT Option Pack (IIS 4.0 Web Server with ASP).
- Microsoft SQL Server 7.0 Database (Standard or Desktop) or the Microsoft Database Engine (MSDE—MSSQL7.0 compliant) with Enterprise Manager installed.

Setup:

- IIS
 - a) In IIS, create a virtual web called FRAMES—configure for BASIC AUTHENTICATION and use either your network domain or the web server domain for authentication.
 - b) Copy all the .asp files into the directory that holds the virtual web created above.
- MSSQL Server 7.0
 - a) In the Query Analyzer, run the SP_ATTACH_DB stored procedure to attach the FramesStructures database files (.mdf and .ldf files).
 - b) Set the “sa” password to FRAMES2000.
- ERED Database Server
 - a) Make sure that the account that you will be using to get into the DOT has full permissions for the file share that contains the folder in which the ERED access database resides (if you used the web server domain for the authentication in the IIS setup, you will also need to add an account with the same name and password as the web server to this server’s domain).

The system overview is given in Figure 3.1.

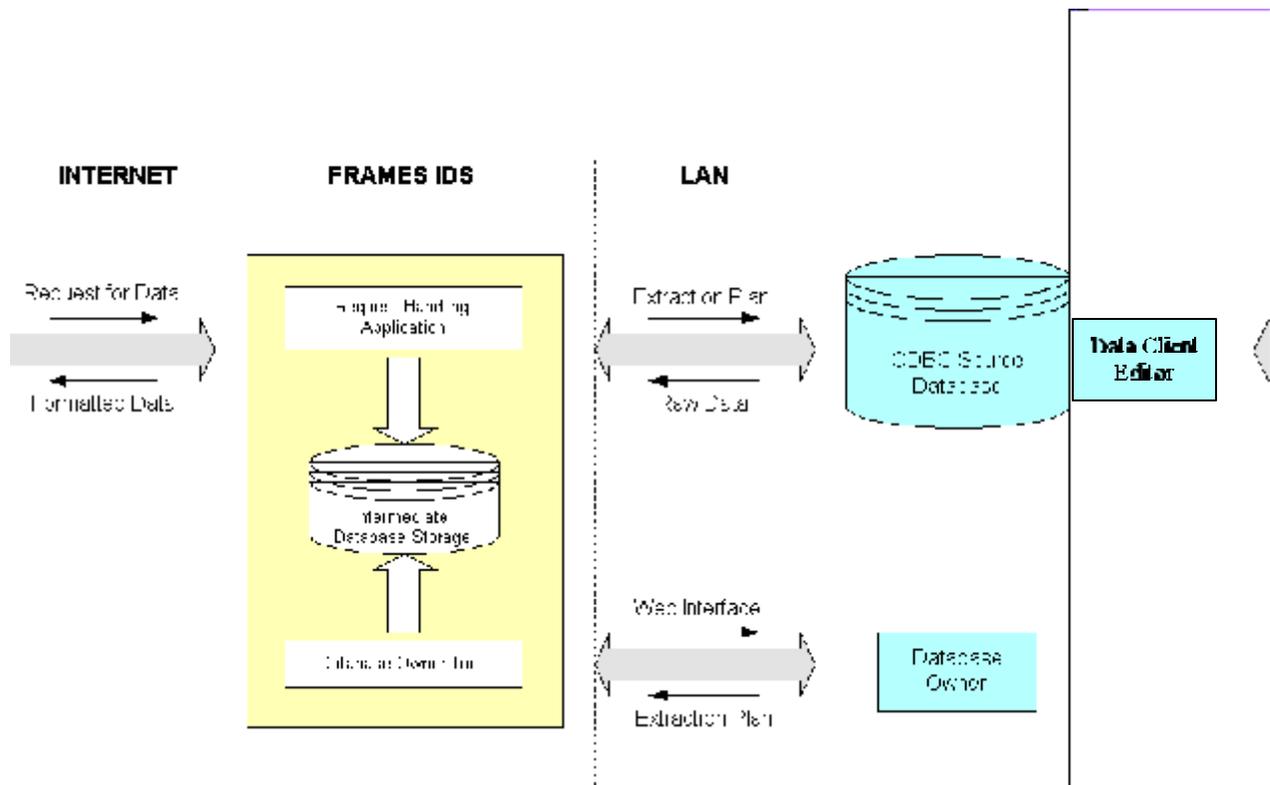


Figure 3.1. System Overview

3.1.2 Input Design

DOT will read the data dictionary, format described in Appendix A, to determine the key(s) needed for extraction. The keys will be placed starting in the eighth column in the DIC. All keys starting with "Num" or "aka" should be ignored by the DOT and DET. An extraction plan should be developed for all others. Any parameter in the DIC that starts with "Num" or "aka" need not have an extraction plan. It will be the job of the DCE to fill in the "Num" and "aka" parameters with the appropriate values.

The input data flow will follow the data-flow diagram in Figure 3.2. The input screens will also follow this diagram as they are virtually the same.

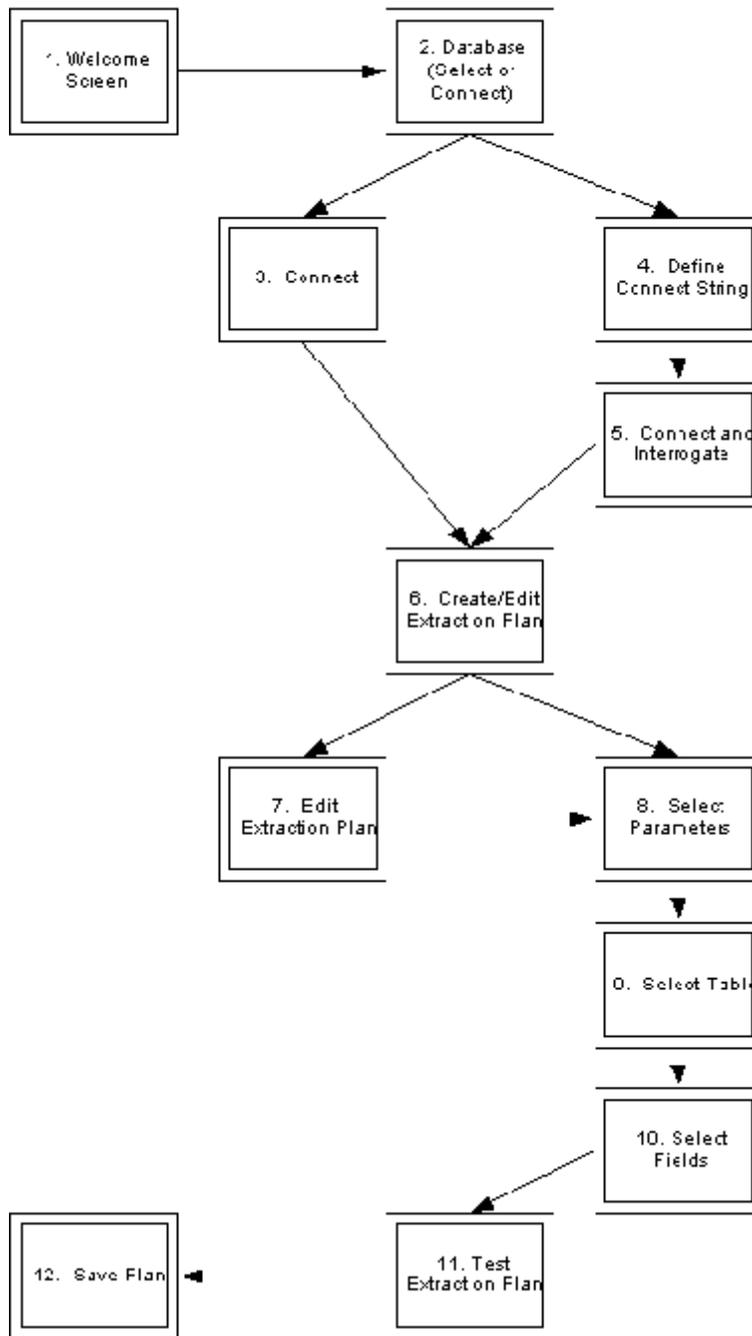


Figure 3.2. Screen and Data-Flow Diagram

3.1.3 Output Design

The DOT pages will be interactive and look like something like Figures 3.3 through 3.16. In most cases, the input from the previous page will be displayed on the current page. This will let the users know the state of the application. The DOT will follow the data-flow diagram in Figure 3.2 to determine which screen to



display. The caption for each figure explains the use of each individual page.

Figure 3.3. Welcome Screen: This Is the First Screen That Appears When the URL for this Software Is Accessed

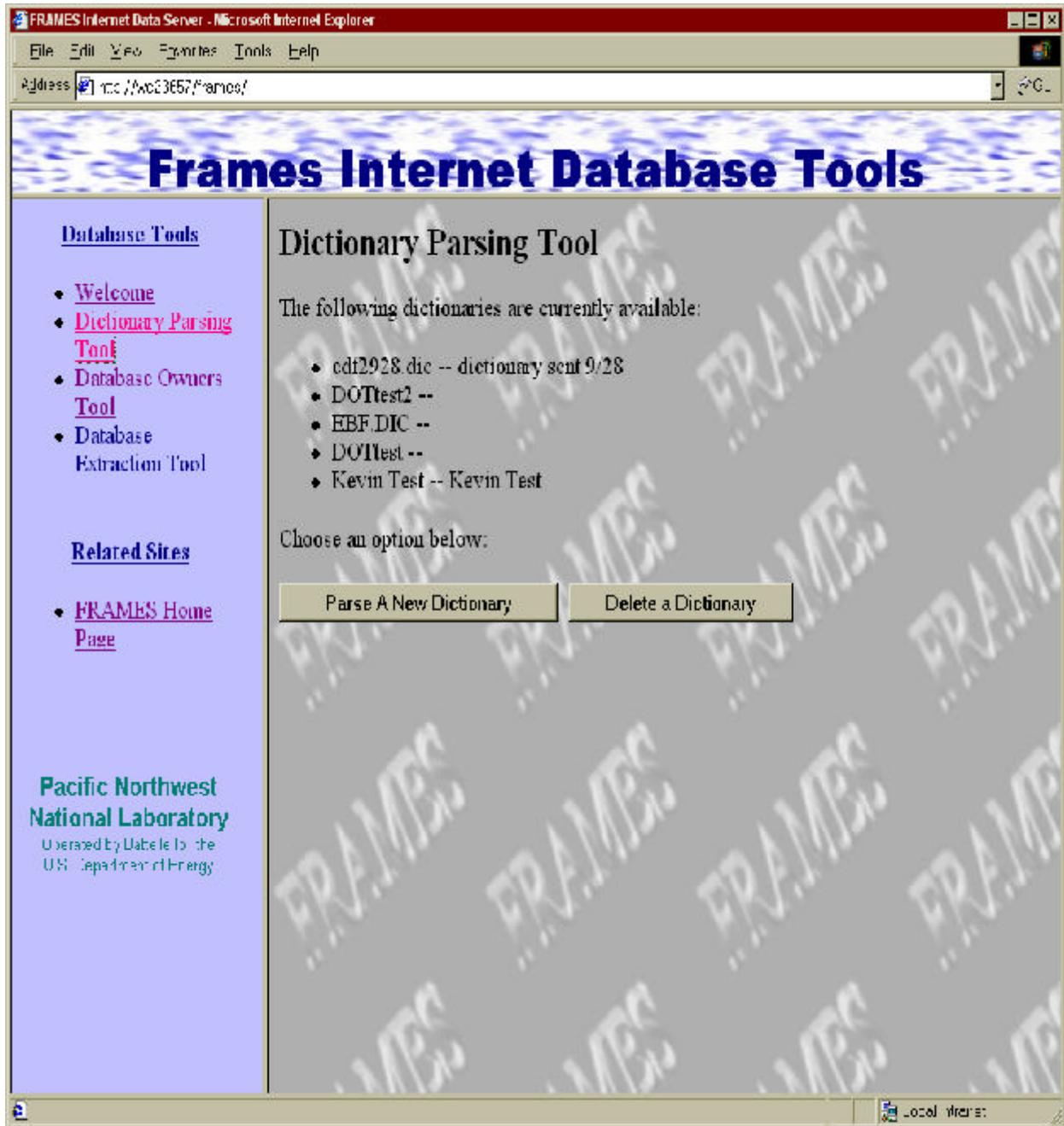


Figure 3.4. Dictionary Parsing Tool: After the “Dictionary Parsing Tool” Is Selected in the Left Panel, a Screen Appears Requesting to Add or Delete a Dictionary

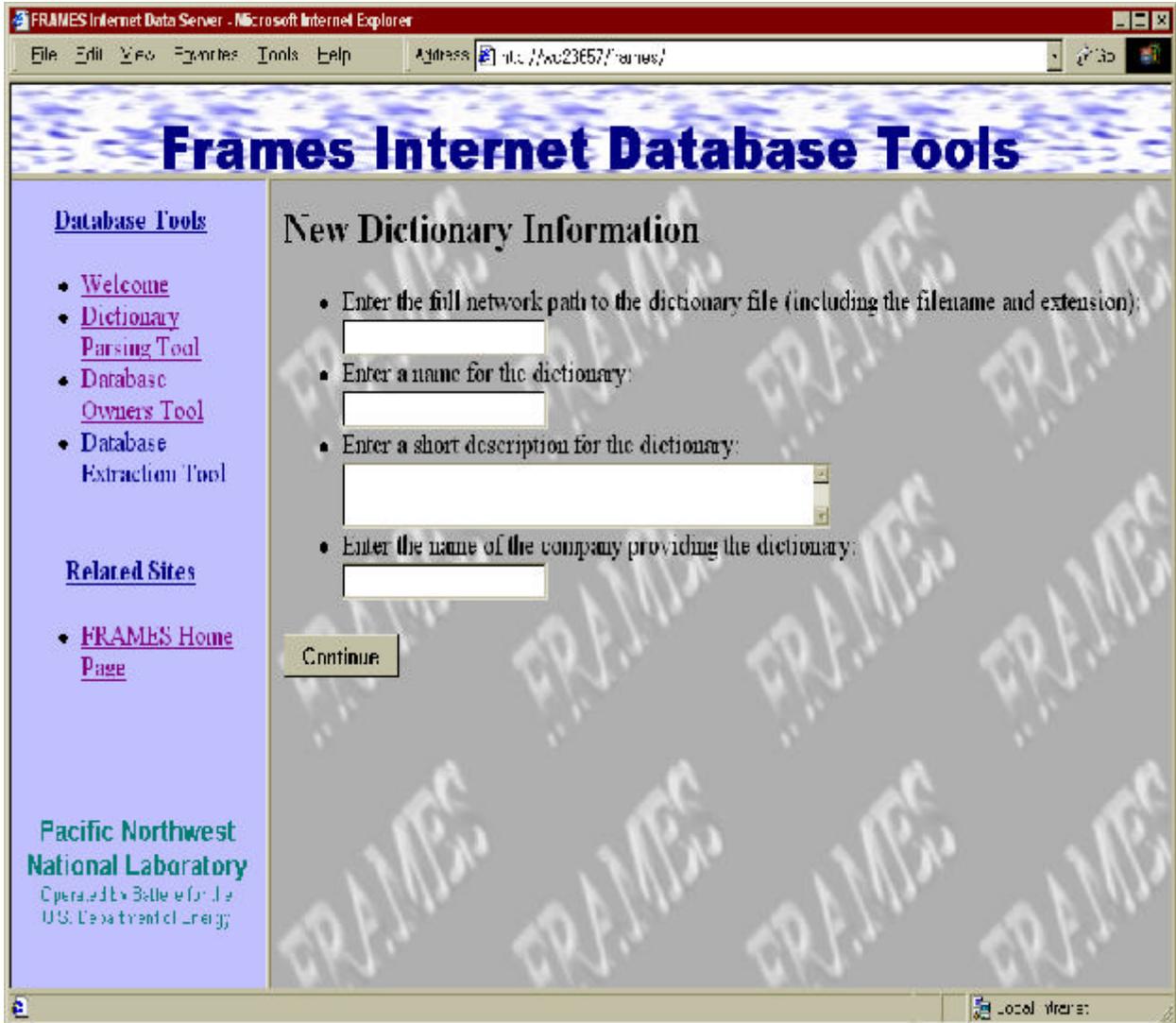


Figure 3.5. New Dictionary Information: The Data Required to Connect to and Parse a Dictionary on the Local Network

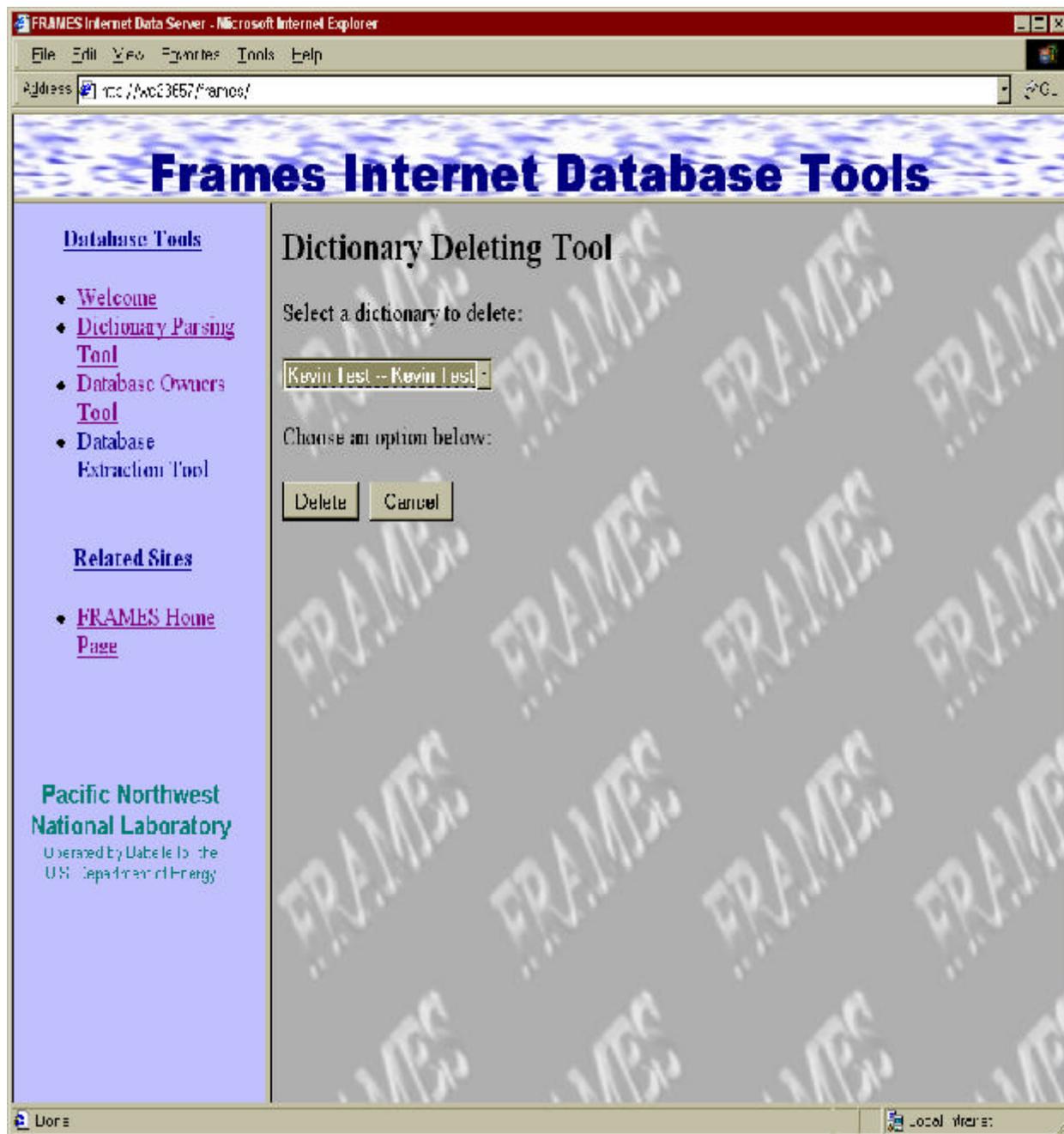


Figure 3.6. Dictionary Deleting Tool: After selecting “Delete a Dictionary,” this Screen Requests Which Dictionary to Delete

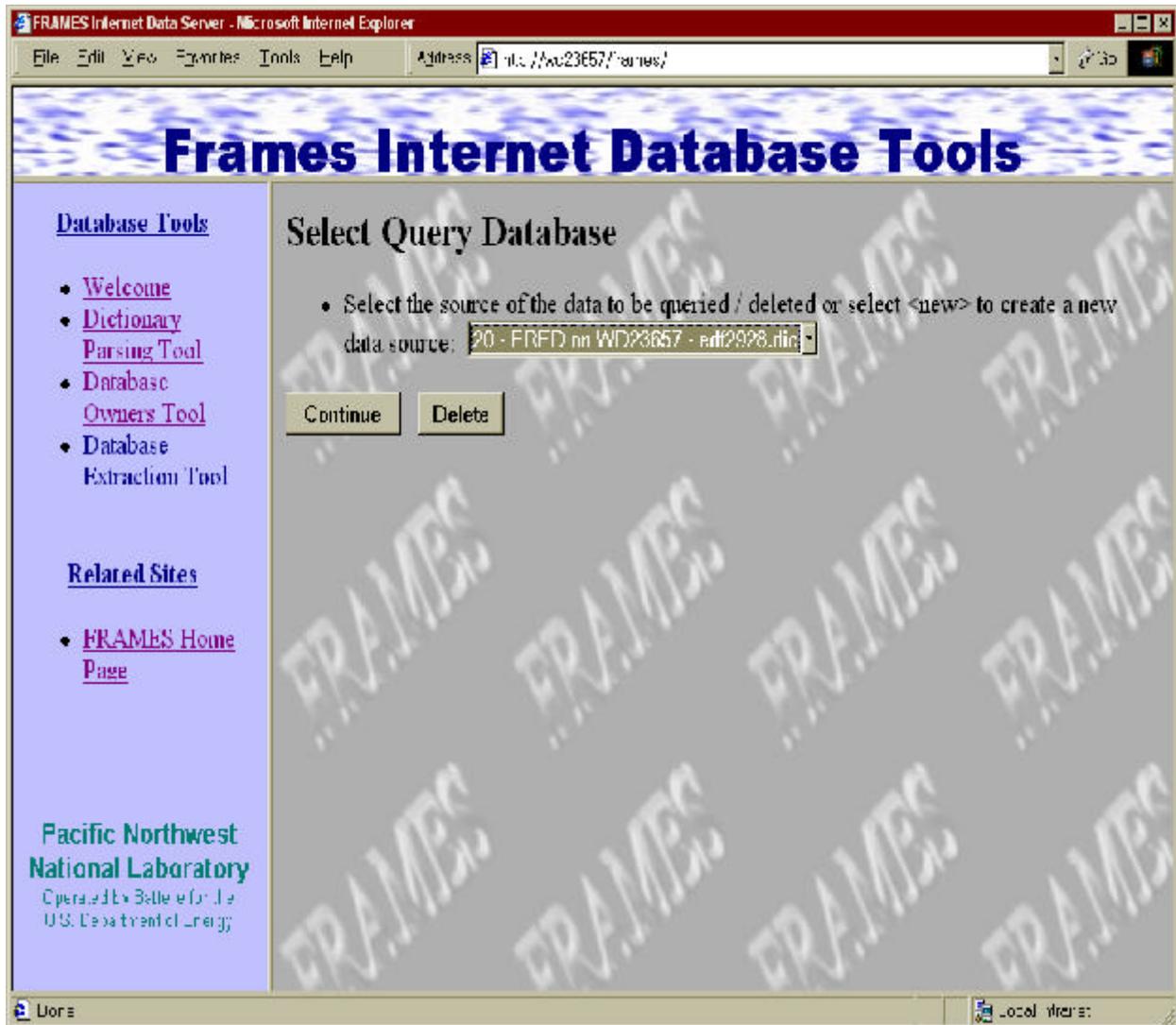


Figure 3.7. Select Query Database: After the “Database Owners Tool” Is Selected in the Left Panel, a Screen Appears Requesting a Database to Connect to or Delete

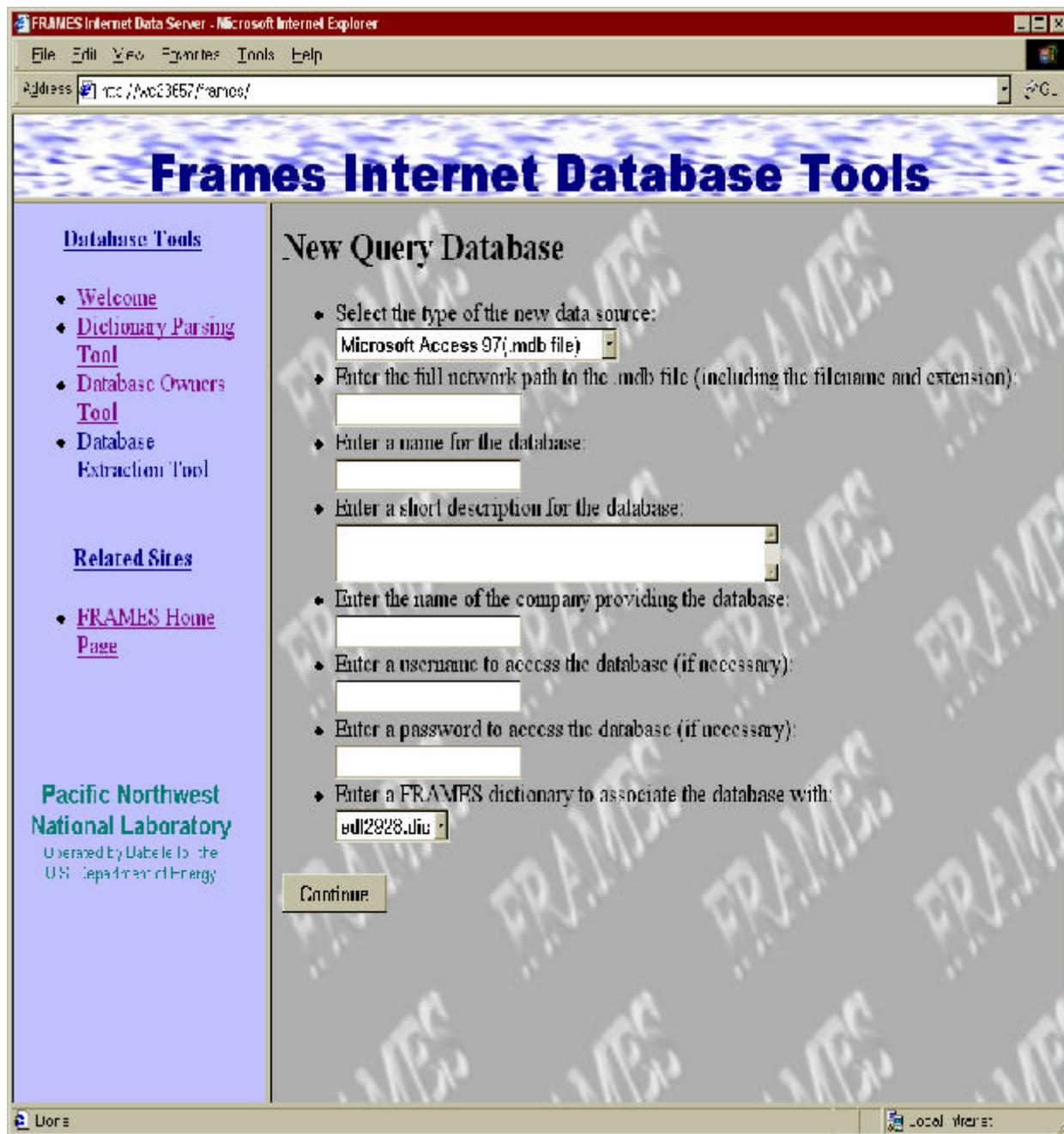


Figure 3.8. New Query Database: The Data Required to Connect to a Database on the Local Network

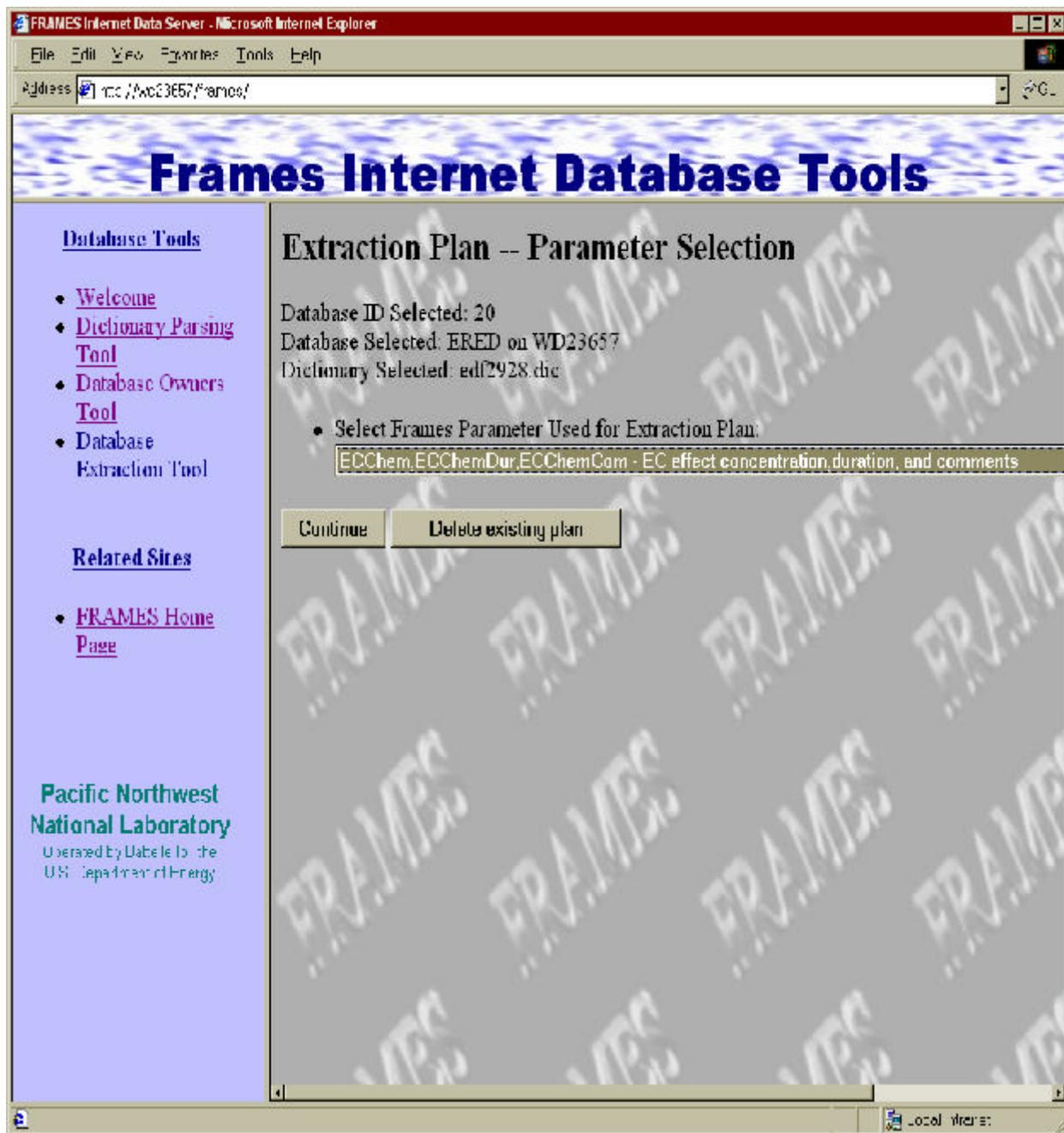


Figure 3.9. Extraction Plan—Parameter Selection: The Extraction Plan That Will Be Built Is for the Parameter(s) Selected Here

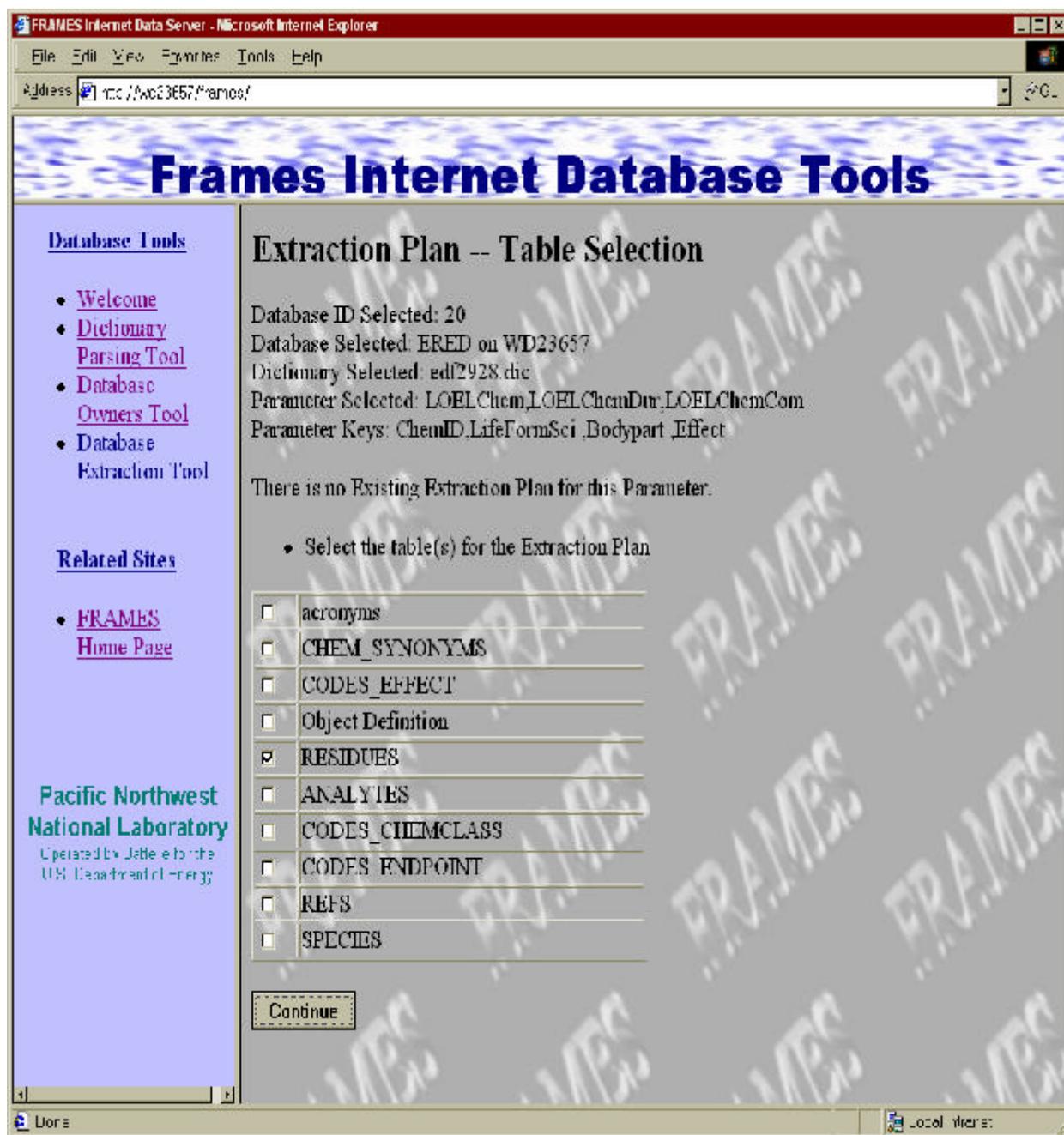


Figure 3.10. Extraction Plan—Table Selection: Database Tables to Use in the Query Are Selected Here

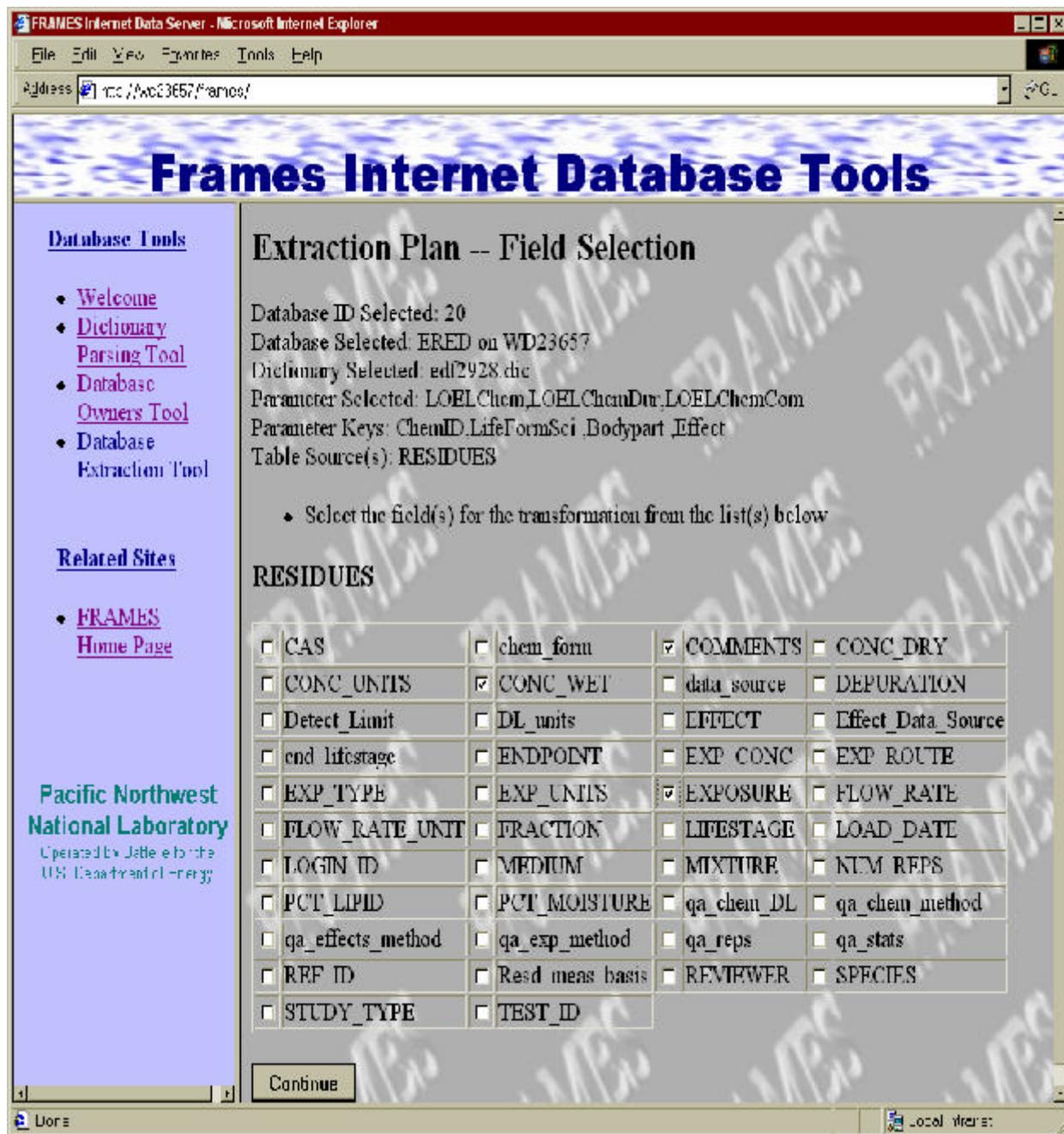


Figure 3.11. Extraction Plan—Field Selection: Fields in the Table Are Selected and That Will Be the Data Returned from the Database

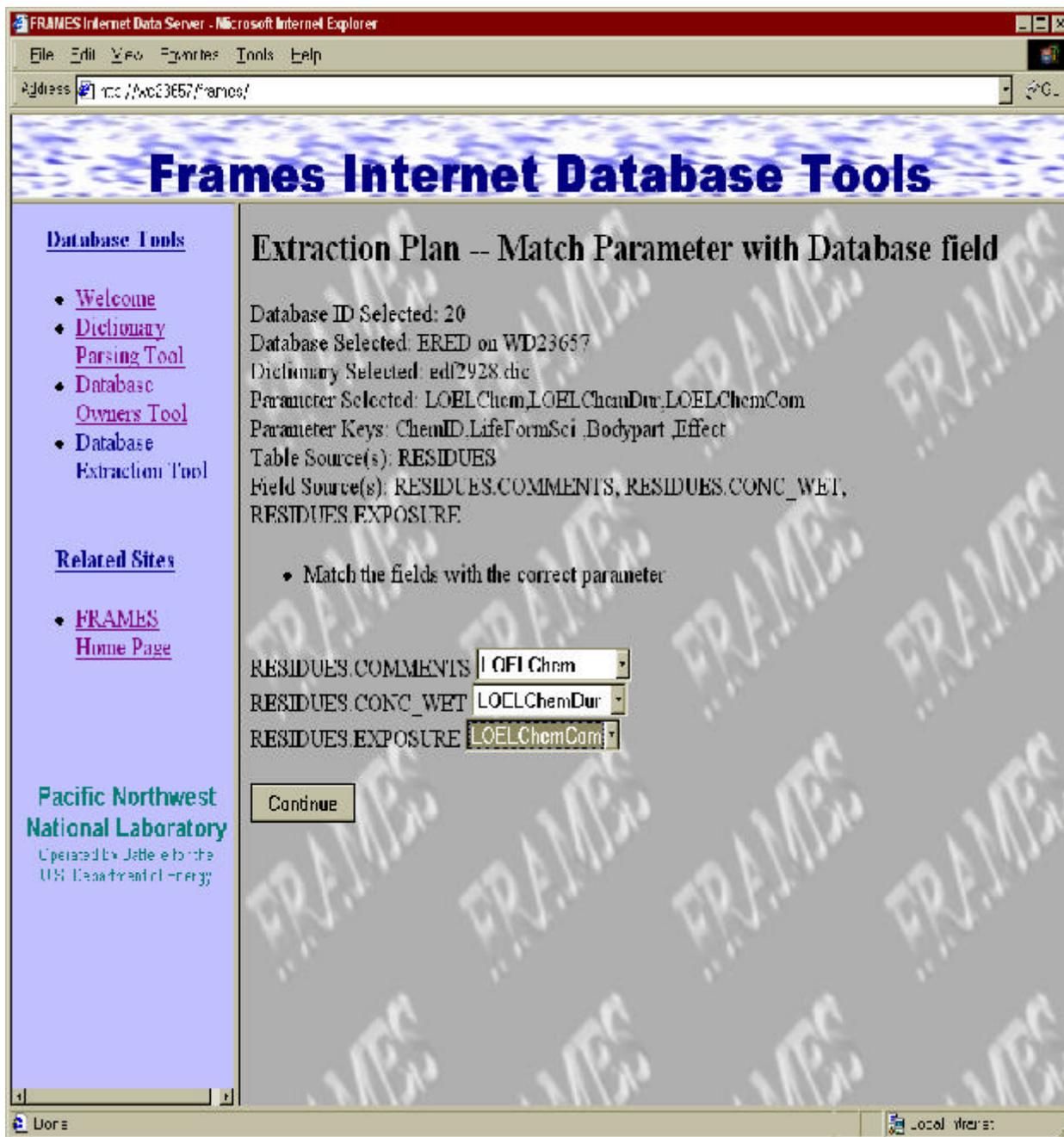


Figure 3.12. Extraction Plan—Match Parameter with Database Field: Match Parameters with Fields in the Database



Figure 3.13. Extraction Plan—Filter Selection: Creating the “Where Clause” for the Query

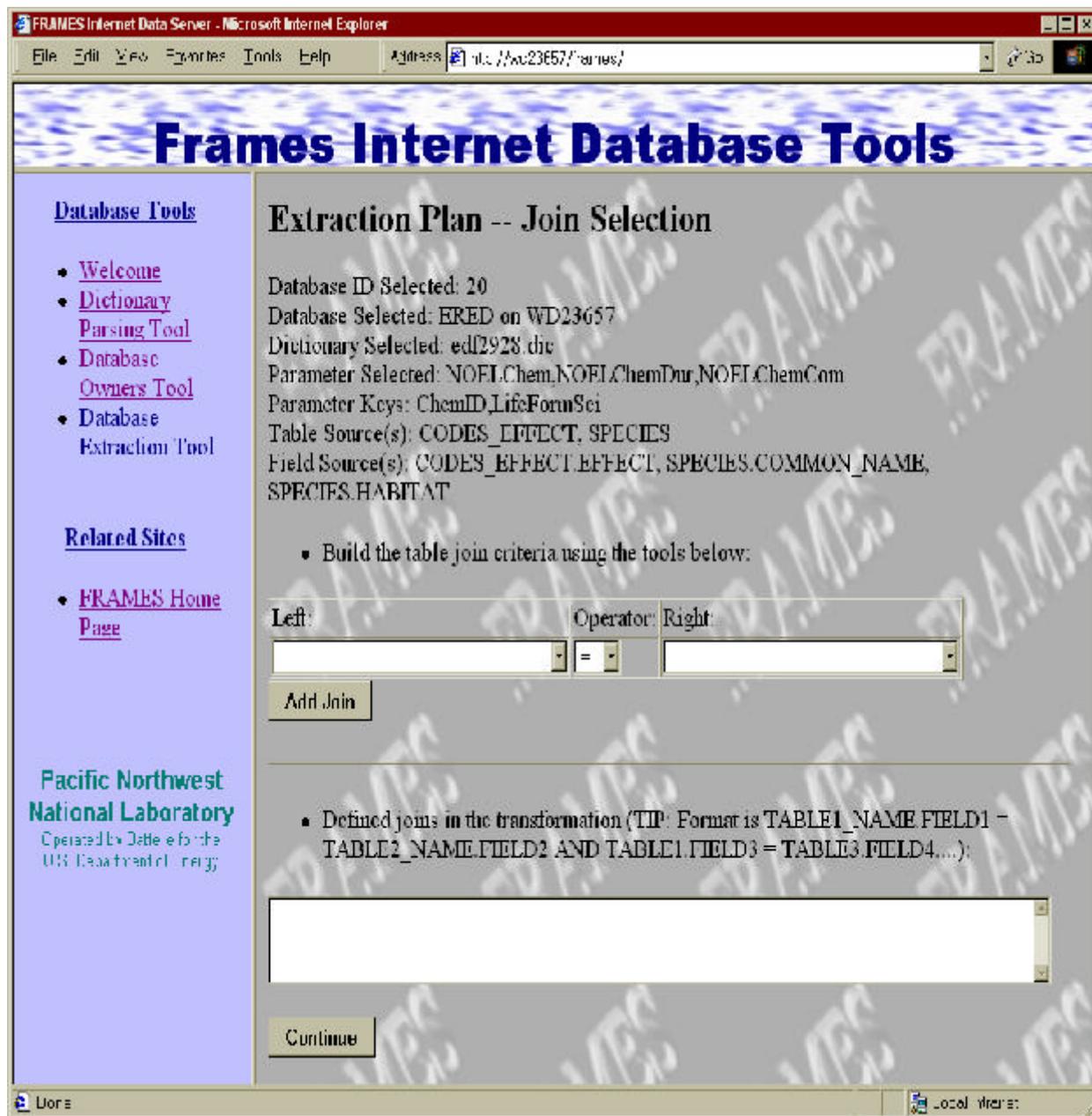


Figure 3.14. Extraction Plan—Join Selection: Describing How the Two Tables Will Be Joined



Figure 3.15. Extraction Plan—Results (Providing a Key Value): an Actual Value must Be Provided for Any Key Values Required



Figure 3.16. Extraction Plan—Results (Display of Results): Displays the Data Found in the Tables That Match the Query

3.2 DET

The following subsections give a gross overview of the DET design. The workings of web servers and active web pages are not discussed. It is assumed that the person who implements this design will have a complete understanding of web servers and active server pages.

3.2.1 System Design

The DET will be the program using active server pages. The DET will be accessed via the Internet and will access the DOT database to determine the result of a data query from a valid client. A valid client is defined as one that provides the correct user name and password using NT authentication. All access to the DET is through the TCP/IP protocols.

3.2.2 Input Design

The input of the DET will be that of the DOT database and the data-query string. The DOT database is to be accessed in the same manner as outlined in Section 3.1. The data query string is to be passed to the DET by the web server via a command line. The format of the commandline/data query is outlined in Section 4.2.1

3.2.3 Output Design

The output of the DET will be sent via a temporary web page over the Internet to the requesting TCP/IP address. The format of the response will not contain any HTML and will only contain the information requested in ASCII text. The format of the response is outlined in Section 4.2.2

3.3 DCE

The DCE is to be used for any DIC file that has LifeFormSci and ChemID for primary keys, (see Appendix A). The DCE is a general-purpose UI for viewing and editing the responses returned from queries to the DCE.

3.3.1 System Design

Then establish a link with the DET using TCP/IP protocols. The link will be secured by using a user name and password. The user will be prompted for this information upon start up of every instance of the UI. The password will be entered as stars on the screen while the user types the actual password.

The UI will have look and feel consistent with that of Windows95+. Input will be driven by the usual point and click methods standard to windows.

3.3.2 Input Design

The list of parameters to query for, display, and edit will be dynamic. This list will come from an ASCII file. The format of the file for the parameters is outlined in Appendix A. The file will be read, and the list of parameter descriptions will be placed on the screen in an outline control on the right hand side of the screen. The outline control will also contain the two primary keys, LifeFomSci and ChemID. The outline screen will look like Figure 3.16.

While the "Chemical Aliases" line of the outline control is highlighted, the selected chemicals will be displayed on the left in a listbox and are determined by the GID file's CON section. A dropdown list shall appear above the ChemId list box for use in aliasing.

While the "Lifeform Selection" line of the outline control is highlighted, all lifeforms will be displayed on the left in a checklist box. All lifeforms will be obtained from the database to be queried. The user will be allowed to select any number of lifeforms. A button will be placed below this listbox to allow a user to provide his/her own lifeForm name.

While the “Lifeform Aliases” line of the outline control is highlighted, the selected Lifeforms will be displayed on the left in a listbox and are selected by the user as mentioned above. A dropdown list shall appear above the lifeforms list box for use in aliasing.

While the “View and Editing” line of the outline control is highlighted, the right-hand side of the screen will display a button to retrieve the data. This button will be enabled and disabled depending on the state of the chemical and lifeform selections. After retrieval, the button is disabled until a change is detected in the chemical and lifeform selections. The button will also be disabled if there is no alias to a user-added lifeform.

While on any child of the “Viewing and Editing” leaf of the outline control, the right-hand side of the screen will display the values of parameters selected. The UI will be constructed such that up to six indices can be used to determine what values to display.

3.3.3 Output Design

The file output of the DCE will be that of a GID file section; the section is outlined in Appendix A. The DCE will write all parameters and their respective values to the GID file. It will also write to the GID file any parameters that are needed to perform the aliasing functionality.

The DCE screens will look like Figures 3.17 through 3.21.

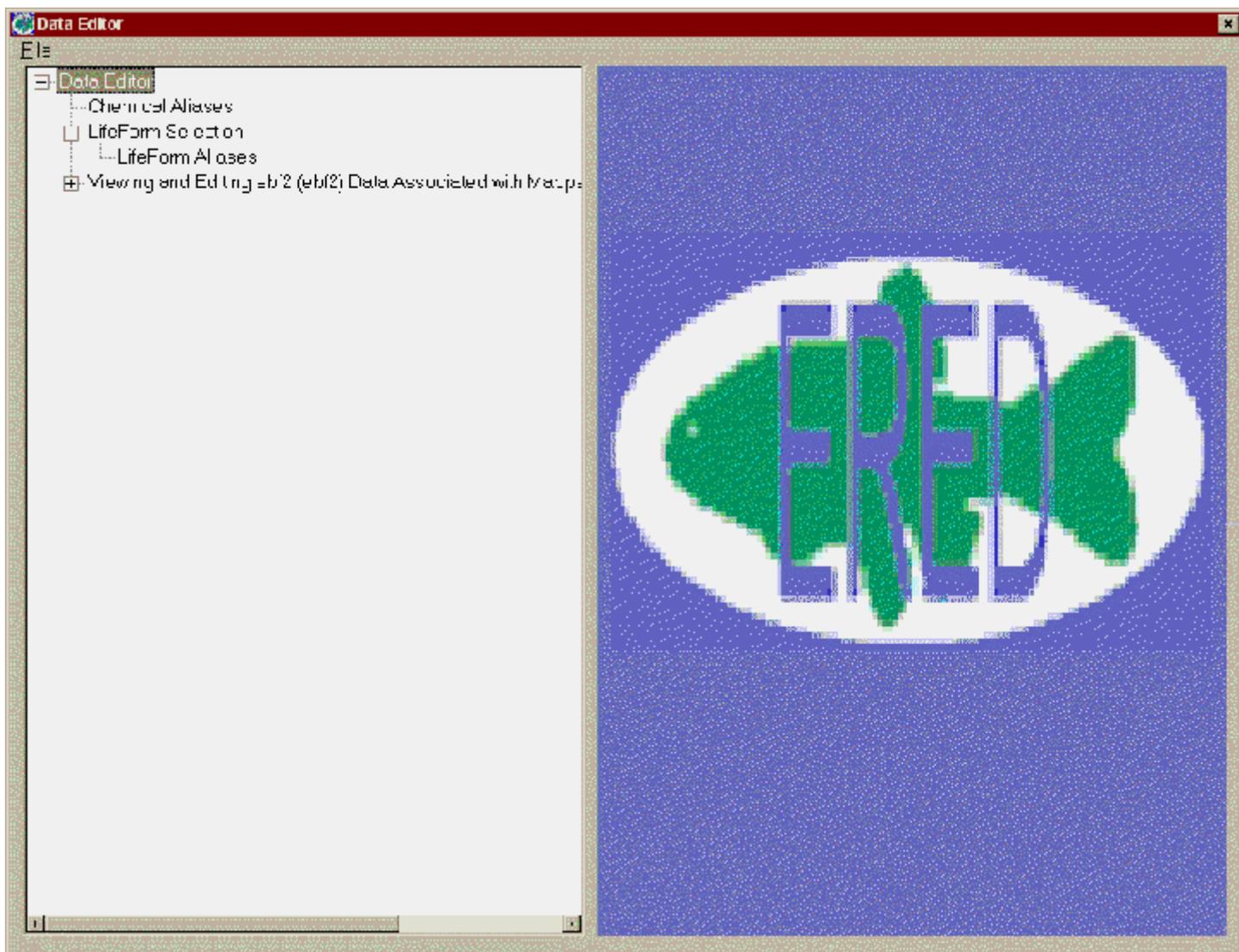


Figure 3.17 DCE Initial Screen

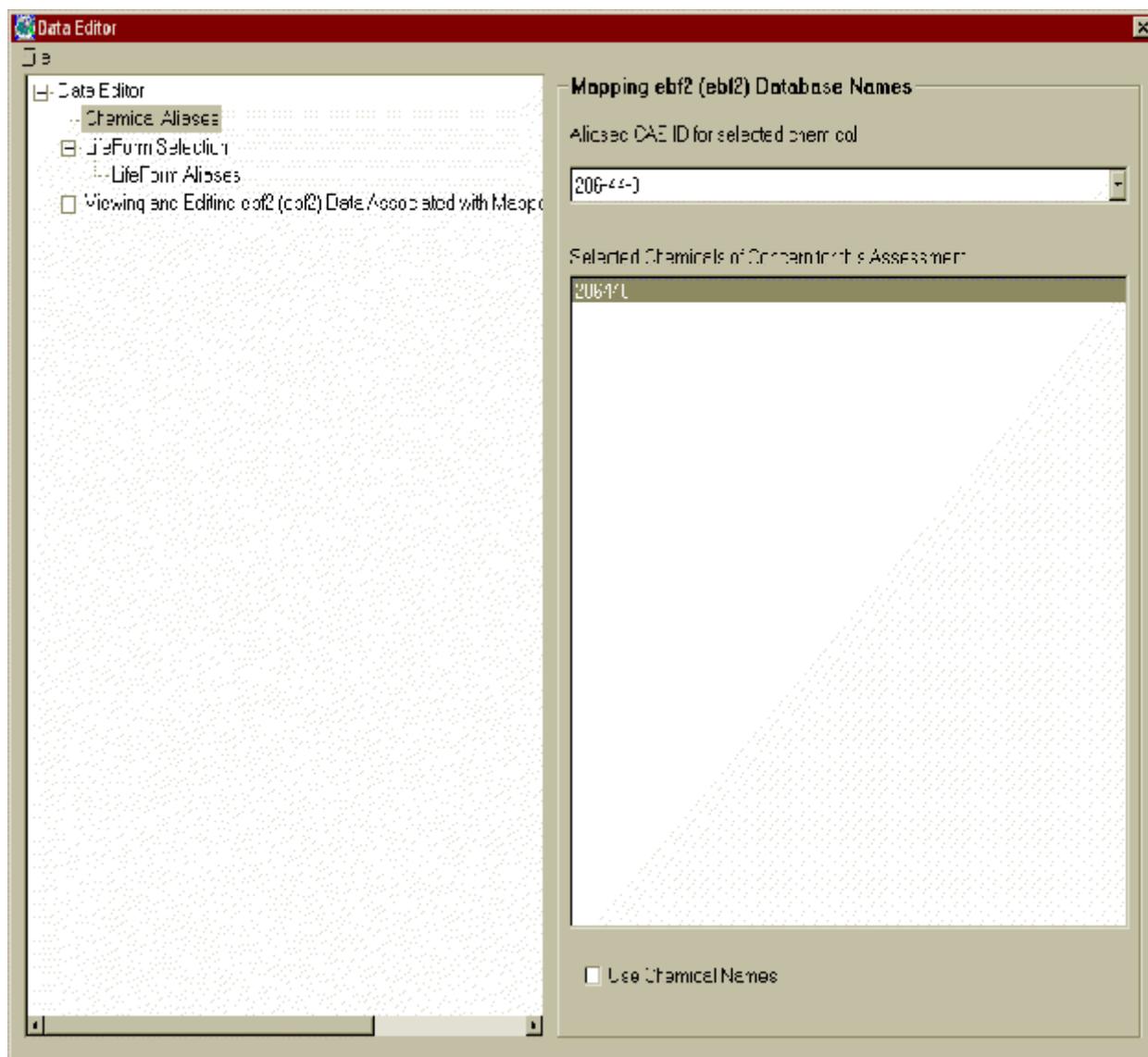


Figure 3.18. Chemical Aliases

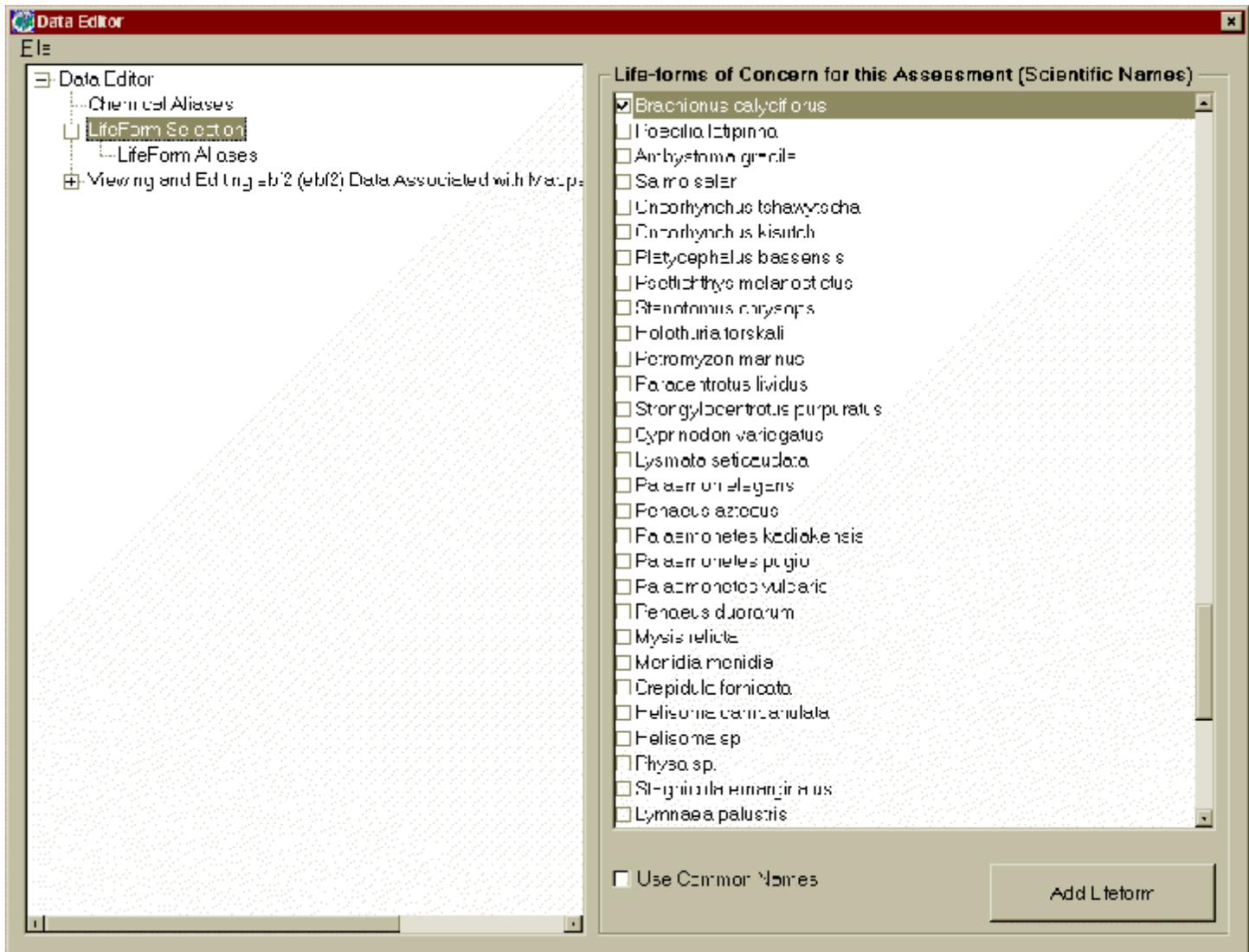


Figure 3.19. Lifeform Selection

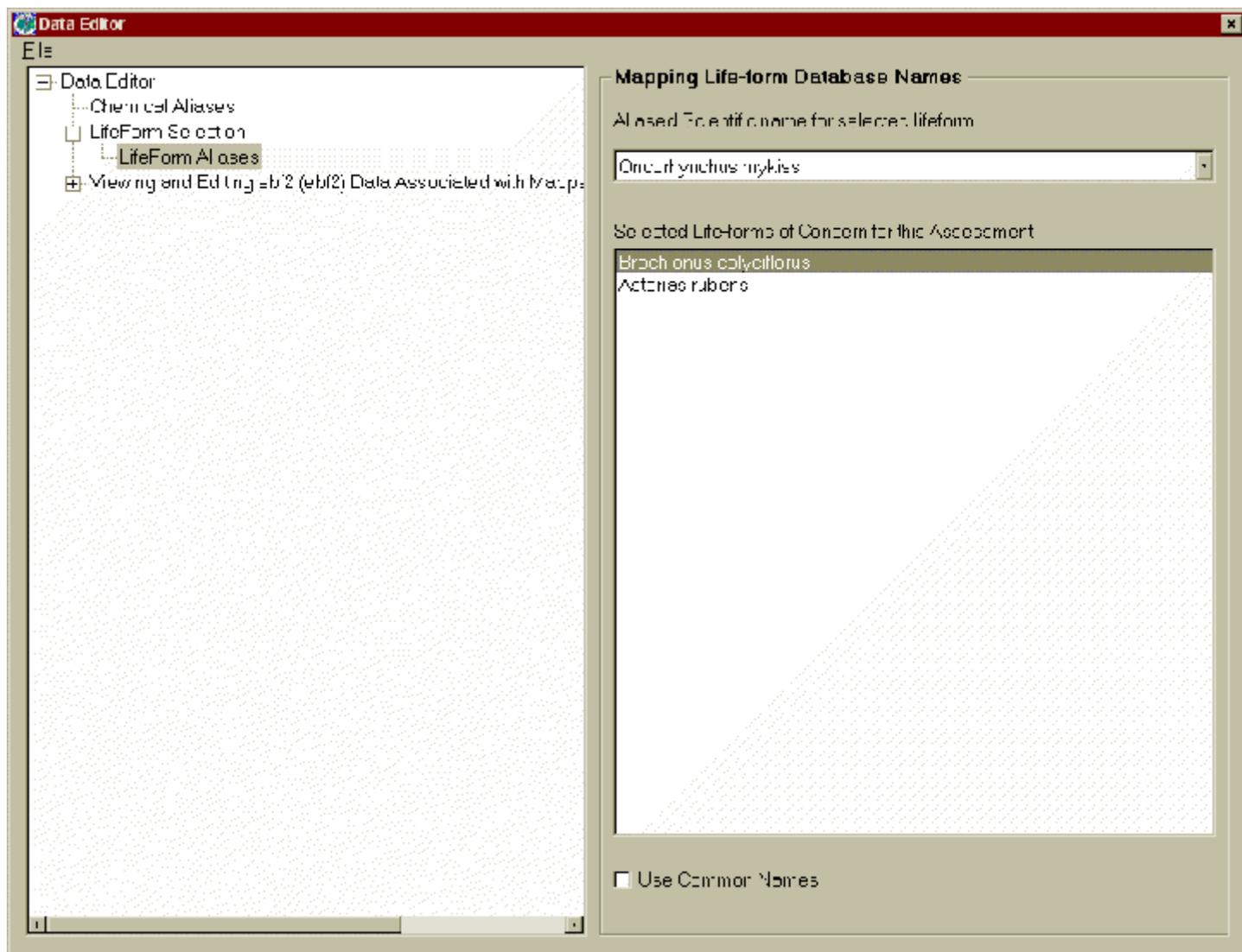


Figure 3.20. Lifeform Aliases

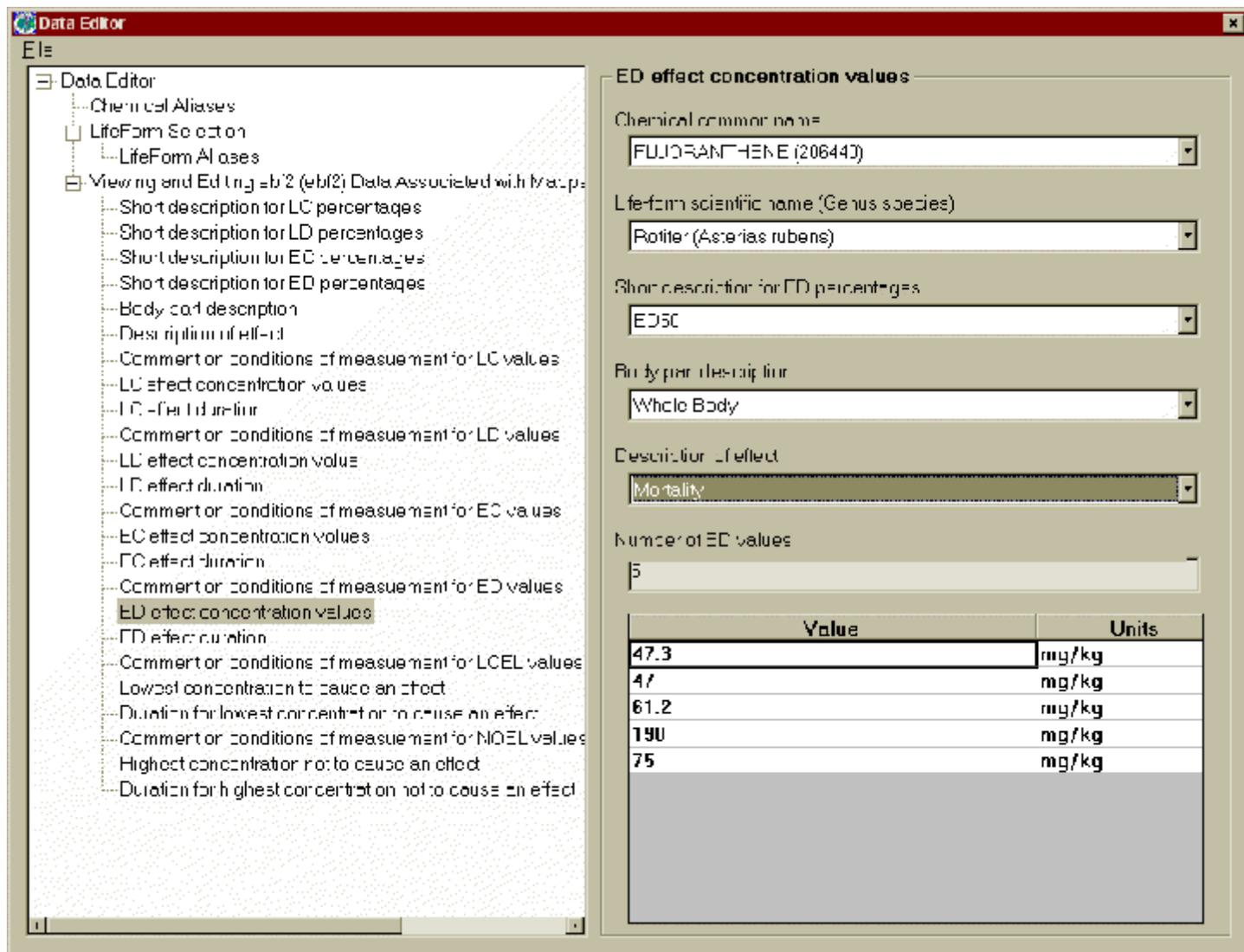


Figure 3.21. Viewing and Editing a Parameter

4.0 Specifications

Specifications provide the details that describe how the design will be implemented. This section reviews the specifications associated with implementing the design outlined in Section 3.0. The specifications for each component will be described separately in this section.

4.1 DOT

4.1.1 Input Specifications

The DOT will import and store FRAMES dictionary files. The format of a dictionary file is discussed in Section 4.1.1.1. The DOT will also import and store Access97 and Access2000 database schemes. Other ODBC-compliant data sources will be included later.

4.1.1.1 Dictionary File

The data-dictionary file consists of three types of information:

- 1) the number of variable lines
- 2) a field header line
- 3) the variable data-description lines.

A data-dictionary file is a comma-delimited ASCII text file that contains the data about the data (sometimes referred to as meta data) that describes a particular dataset. The first line of the dictionary file contains a number N, followed by the name of the dataset. N is defined as the number of parameter declarations plus table declarations minus one. The second line of the file contains the label for each column of data in the file. The column definitions are as follows:

Column Definition

```
=====
```

“Code”	Variable Name - (String20)
“Dimensions”	Dimensionality - (Integer)
“Data Type”	Type of data - (String) [“Logical”],[“Integer”],[“Real”],[“String”]
“Min”	Minimum value/length for variable - (Double)
“Max”	Maximum value/length for variable - (Double)
“Units”	Units of variable - (String20)
“Description”	Description of variable - (String80)
“Key1”	First indexing key of variable - (String20)
“Key2”	Second indexing key of variable - (String20)
...	More indexing key variables if needed - (String20)

Following the column header line, there are N-1 number of declaration lines. A parameter declaration line consists of the columns described above. Each line contains all of the description data needed for a parameter declaration. All parameter keys must have a declaration and must also be one dimensional or less. The table declarations must be made after all parameter declarations have been made. Each table declaration will consist of the word “~Table,” followed by the number of parameters in the table, and then followed by the parameters associated with the table.

The exact contents of the dictionary file are present in Appendix A

4.1.2 Output Specifications

The output of the DOT will be the interactive screens described in Section 3.1 and the database that tool populates. The DOT database is outlined in the following section.

4.1.2.1 DOT Database

The DOT will place all extraction plans, imported database schema, and DIC information in the tables defined in Tables 4.1–4.12. The DOT will use the software outlined in Section 3.1 to access these tables. The tables are in no particular order.

Table 4.1. Database

Name	Type	Length	Description
DatabaseID	int	4	A number identifying the imported database
DatabaseName	char	50	The name of the imported database
Description	char	200	A description of the imported database
Source	char	50	The name of the database provider
ConnectionString	nvarchar	200	ODBC connection string
DictionaryID	int	4	The number identifying which dictionary to use

Table 4.2. Dictionary

Name	Type	Length	Description
DicID	int	4	A number identifying the imported dictionary
DicName	char	50	The name of the imported dictionary
Description	char	200	A description of the imported dictionary
Source	char	50	The name of the dictionary provider
ConnectionString	nvarchar	200	Connection string

Table 4.3. Extraction Plan

Name	Type	Length	Description
ExtPlanID	int	4	A number identifying the extraction plan
ParameterID	int	4	A number identifying the related parameter
ExtPlanSQL	varchar	4000	The SQL extraction plan

ExtPlanDesc	varchar	4000	The description of the extraction plan
ExtPlanCDate	datetime	8	The date and time of the extraction plan
DatabaseID	int	4	A number identifying the related database

Table 4.4. Extraction Plan Fields

Name	Type	Length	Description
EPFieldID	int	4	A number identifying the related extraction-plan field
ExtPlanID	int	4	A number identifying the related extraction plan
FieldID	int	4	A number identifying the related field

Table 4.5. Extraction Plan Join

Name	Type	Length	Description
EPJoinID	int	4	A number identifying the extractionplan join clause
ExtPlanID	int	4	A number identifying the related extraction plan
JoinClause	nvarchar	1000	String specifying the join clause

Table 4.6. Extraction Plan Static Filter

Name	Type	Length	Description
EPStaticFilter ID	int	4	A number identifying the extraction-plan static filter
ExtPlanID	int	4	A number identifying the related extraction plan
StaticFilter	nvarchar	300	String specifying the static filter

Table 4.7. Extraction Plan Tables

Name	Type	Length	Description
ExtPlanTableID	int	4	A number identifying the extraction-plan table
ExtPlanID	int	4	A number identifying the related extraction plan
TableID	int	4	A number identifying the table

Table 4.8. Fields

Name	Type	Length	Description
FieldID	int	4	A number identifying the field

Name	Type	Length	Description
TableID	int	4	A number identifying the related table
DatabaseID	int	4	A number identifying the related database
TableName	nvarchar	50	The name of the table from where this field was defined
FieldName	nvarchar	50	The name of the field
FieldType	nvarchar	50	The type of the field
FieldLength	nvarchar	50	The length of the field
FieldDesc	nvarchar	250	A short description of the field

Table 4.9. KeyFilter

Name	Type	Length	Description
KeyFilterID	int	4	A number identifying the key filter
ExtPlanID	int	4	A number identifying the related extraction plan
ParameterID	int	4	A number identifying the related parameter
FieldID	char	50	A number identifying the related field
Operator	char	50	The operator of the filter

Table 4.10. Parameter

Name	Type	Length	Description
ParameterID	int	4	A number identifying the parameter
ParameterName	vchar	70	The name of the parameter
ParameterDesc	vchar	1000	A short description of the parameter
DicID	int	4	A number identifying the related dictionary
Visible	vchar	1	Show this parameter for making of extraction plan

Table 4.11. Parameter Keys

Name	Type	Length	Description
KeyID	int	4	A number identifying the key
KeyName	char	50	The name of the parameter key
ParameterID	int	4	A number identifying the related parameter

KeyDesc	char	200	A short description of the key
---------	------	-----	--------------------------------

Table 4.12. Tables

Name	Type	Length	Description
TableID	int	4	A number identifying the table
DatabaseID	int	4	A number identifying the related database
TableName	nvarchar	50	The name of the table
Description	ntext	16	A short description of the key
RecordCount	int	4	The count of records in the table

4.2 DET

4.2.1 Input Specifications

There are two inputs to the DCE, the DOT database and the Data Query string. The input to the DCE is a Data Query string passed to it via the Internet. The specification of the DOT database is described in Section 4.1.2. The specification of the Data Query string is described in Section 4.2.1.1.

4.2.1.1 Format of Data Query

A data item request has the form of:

URL?[querystring]

where URL is the domain name of the server and querystring is equal to a combination of the following:

DD = followed by the name of the data dictionary

DB = followed by the database number

Step = followed by the action to take (extract)

Parameter = followed by the name of the parameter to be retrieved

Keycount = followed by the number of keys

Key# = followed by the nth keyname

Val# = followed by the nth key value

4.2.1.2 Data Query Example

URL/extractdata.asp?step=extract?DD=edf.dic&DB=4&parm=deathrate&keycount=2&key1=chemical&key2=lifeform&val1=SR90&val2=frog&

4.2.2 Output Specifications

The DET has only one output and that is the response to a data query. The response is returned via the Internet and should only contain text, no HTML. The format of a response to a data query should list the parameter or table values with a count. The record fields should be quoted, comma-separated strings. A line following the count should contain the list of field names separated by commas. The field names should correspond with the names found in the dictionary. For instance if the query was looking for Parameter="LifeFormSci," then the field name should be "LifeFormSci." All records should be separated by line feeds. The format should look something like the following:

```
6
EDChem,EDChemDur,BodyPart,Effect
"47.3","4","Whole Body","ED50","Mortality"
"47","4","Whole Body","ED50","Mortality"
"61.2","4","Whole Body","ED50","Mortality"
"190","4","Whole Body","ED50","Mortality"
"75","4","Whole Body","ED50","Mortality"
"1.9","4","Whole Body","ED50","Physiological"
```

4.3 DCE

4.3.1 Input Specifications

There are two types of input files, the GID file, and the Eco Benchmarks dictionary file. The GID file contains Contaminant list information in a section named "FUI." The format of a GID file and the contents of the FUI section are laid out in <http://mepas.pnl.gov:2080/earth/earth.htm> and earlier documented in Whelan et al. (1997). The DCE will read an Eco Benchmarks dictionary file of the format outlined in Section 4.1.1.1. The exact contents of that file are listed in Appendix A.

4.3.1.1 DES File for DCE

The UI executable entry in the DES file will contain the URL for the desired database. For example, the first two lines of the DES file should look something like this:

```
mf, Version 2.0
Ecology, ERED Database, dataedit.exe http://wes.mil.gov/ered.asp ebf.dic 4
```

The rest of the file will be as stated in the FRAME specifications document. The command line for the DCE will be as follows:

```
[Executable] [URL] [FUI File] [Run File] [Site Index] [Model Index] [Model ID]
```

For example:

```
C:\frames\dataedit.exe http://mil.gov/ered.asp edf.dic 4 C:\frames\scenario C:\frames\~tmp 1 1 ecol
```

The DCE will strip the number off the model ID and append .dic and prepend the DCE executable path to obtain the .DIC file that contains the primary key and dictionary information.

4.3.2 Output Specifications

The DCE will output Eco Benchmarks data in the form of a GID file section. The GID file format is outlined in <http://mepas.pnl.gov:2080/earth/earth.htm> and earlier documented in Whelan et al. (1997). The exact contents of the Gid file section are listed in Appendix A.

5.0 Testing Approach and Results

As noted previously, the FRAMES Internet Database Tools are programs that will enable the FRAMES system to access data from MS Access 97 or Access 2000 databases regardless of table structure, formats, and location. The FRAMES Internet Database Tools are comprised of three components: the DOT, the DET, and the DCE. Each component has specific requirements that it must meet independent of the other components. For this reason, each component was tested separately

This section describes the testing conducted for the DOT, DET, and DCE. It includes summaries of the requirements on which testing was based and descriptions of test cases, and it lists the results of their implementation. One test case for the DCE was generated through the FRAMES User Interface (FUI). The GID file is included with the test package. The GID file generated through independent testing should match this file.

5.1 DOT

The DOT allows the database owner to specify the mapping between parameters needed by models in FRAMES and their locations in their database. The mapping specified is used to create the database extraction plan.

5.1.1 Summary of Requirements

Requirements for the DOT are described in Section 2.0. of this document. These requirements were reworded, as necessary, to create a list of fundamental requirements that are suitable for testing. Table 5.1 lists these requirements. To ensure that the DOT meets the requirements listed in Table 5.1, test cases were developed to check performance. Table 5.2 shows the relationship between these requirements and the test cases described in Section 5.2.

Table 5.1. Fundamental Requirements for Testing the DOT

Requirement Number	Requirement
1	The DOT shall connect to any MS Access 97 or Access 2000 database.
2	The DOT shall be accessible on a server via the Internet.
3	The DOT shall ask for user login credentials for security.
4	The DOT shall provide a web-page interface for creating stored execution plans.
5	The DOT shall provide a web-page interface for creating and storing ODBC connection information for the source database.
6	The DOT shall provide a web-page interface for creating and storing DIC connection information for FRAMES datasets.
7	The DOT shall provide the capability to test and modify extraction plans.
8	The DOT shall be used by the database owners and will assist them in mapping the data from the source database into FRAMES dataset(s) outlined by the DIC file.
9	The DOT shall be capable of joining data from separate tables within a database when searching for requested data.

Requirement Number	Requirement
10	The DOT shall be capable of filtering out non-unique data when the option is selected through the DOT user interface.
11	The DOT shall allow the user to enter data necessary to specify the name and location of a database.
12	The DOT shall allow the user to choose the FRAMES dataset DIC to map to (Note: No checking is performed to ensure that the DIC is in the correct format).
13	The DOT shall read a DIC file to determine parameter list.
14	The DOT shall read a DIC file to determine parameter correlations.
15	The DOT shall allow the user to specify how parameters are mapped.
16	The DOT shall create an extraction plan(s) in the form of an SQL statement with replaceable parameters (keys).
17	The DOT shall create an extraction plan(s) capable of retrieving a list.
18	The DOT shall create an extraction plan(s) capable of retrieving related data from multiple tables.
19	The DOT shall create a stored extraction plan that may have one or more key values associated with it.
20	The DOT shall provide error messages when DIC or MDB files are not found.
21	The DOT must operate within FRAMES under Windows-95 or higher.

Table 5.2. Relationship Between Test Cases and DOT Fundamental Requirements

		Test Case Name (DOT_xx)					
		01	02	03	04	05	06
Requirements	1	X					
	2	X					
	3	X					
	4	X					
	5	X					
	6	X					
	7	X					
	8	X					
	9			X			
	10				X		
	11	X					
	12	X					
	13	X					
	14	X					
	15	X					
	16						X
	17	X					
	18			X			
	19		X	X			
	20					X	
	21	X					

5.1.2 TEST CASES

5.1.2.1 DOT_01

5.1.2.1.1 Description and Rationale

This test case addresses the capability of the DOT to allow the user to map a DIC file and a database and its parameters. The parameter being mapped will have one key parameter associated with it. This test case also addresses the basic functionality of the DOT; therefore, several of the other requirements are addressed in this case. The requirements addressed are listed below. These requirement numbers are referenced in the results section (Note: the requirement numbers used below do not correspond to the requirement numbers listed in Table 5.1).

- 1) The DOT shall connect to any MS Access 97 or Access 2000 database. The DOT shall be accessible on a server via the Internet.
- 2) The DOT shall ask for user login credentials for security.
- 3) The DOT shall provide a web-page interface for creating stored execution plans.
- 4) The DOT shall provide a web-page interface for creating and storing ODBC-connection information for the source database.
- 5) The DOT shall provide a web-page interface for creating and storing DIC connection information for FRAMES datasets.
- 6) The DOT shall provide the capability to test and modify extraction plans.
- 7) The DOT shall be used by the database owners and will assist them in mapping the data from the source database into FRAMES dataset(s) outlined by the DIC file.
- 8) The DOT shall allow the user to enter data necessary to specify the name and location of a database.
- 9) The DOT shall allow the user choose the FRAMES dataset DIC to map to (Note: No checking is performed to ensure that the DIC is in the correct format.)
- 10) The DOT shall read a DIC file to determine parameter list
- 11) The DOT shall read a DIC file to determine parameter correlations.
- 12) The DOT shall allow the user to specify how parameters are mapped.
- 13) The DOT shall create an extraction plan(s) capable of retrieving a list.
- 14) The DOT must operate within FRAMES under Windows-95 or higher.

5.1.2.1.2 Input Data

This test case requires that you use the dictionary file “EBF.DIC” and the database “DOTDB01.MDB.” Both of these files should be located on a computer that can be mapped and accessed through the Internet. All other input will be described in the procedure section below.

5.1.2.1.3 Expected Results

It is expected that the DOT will execute without error and will meet all of the requirements listed in Section 5.2.1.1.

5.1.2.1.4 Procedure

Place the DOT server on a computer that can be accessed by the computer that will be used to carry out this test plan. Then bring up a browser (i.e., Netscape, Internet Explorer, etc.) and type in the URL to map the computer and folder location. If the DOT server is located on a computer that is behind a firewall, which is recommended, you will be prompted to enter a user ID and password. Enter those and click on the “Ok” button. The web page user interface should appear. The following sections will describe how to map a DIC file and a database.

To map a DIC file:

To the left of the screen there should be a list of choices entitled “Database Tools.” Click on the Dictionary Parsing Tool. On the screen to the right, click on the “Parse A New Dictionary” button. You will be prompted to fill in the network location, name, short description, and your company name. Enter the full network path and dictionary file name. The dictionary file name will be “EBF.DIC.” For the second box, the name box, enter in the name DOTtest. The short description and company name boxes are not required information and can be left blank. Click the “Continue” button. A screen informing you that the dictionary was parsed successfully should appear. The screen will list the contents of the dictionary file. You can check that the contents are correct by opening up the actual dictionary file using a file editor such as Notepad or Wordpad. This completes the process for mapping a dictionary file.

To map a database and create an extraction plan for a parameter:

To the left of the screen, there should be a list of choices entitled “Database Tools.” Click on the Database Owners Tool. On the screen to the right, use the list box to select “<new>” and then click on the “Continue” button. You will then be prompted to enter in the database information. In the first box, select “Microsoft Access 2000 (.mdb file).” In the next box, enter the full network path and the name of the database, which is “DOTdb01.mdb.” In the next box, enter “Test01.” The next four boxes can be left blank. In the last box, use the list box to select “DOTtest,” which is the DIC file that you just mapped. Then click on the “Continue” button.

The “Extraction Plan - Parameter Selection” page should appear on the screen. Use the list box to select “Effect - Description of effect” and then click on the “Continue” button.

The “Extraction Plan - Table Selection” page should appear. Click on the box to the left of “Data3” to select this as the table for data to be retrieved from. Then click on the “Continue” button.

The “Extraction Plan - Field Selection” page should appear. Click on the box to the left of “T3FieldOne” to select this as the field data to be retrieved from. Then click on the “Continue” button.

The “Extraction Plan - Match Parameter with Database Field” page should appear. “Bodypart” should appear in the list box. This indicates that the field “T3Fieldone” in the table “Data3” will be mapped to the “Effect” parameter in the DIC file. Then click on the “Continue” button.

The “Extraction Plan - Filter Selection” page should appear. Use the list box provided to select “Data3.T3Fieldone” for the Field and use the other list box to select “>=” for the Operator. Then click on the “Continue” button.

The “Extraction Plan - Results” page should appear. You will need to fill in a key value for the Effect. Type “22” in the text box. Then click on the “Test Extraction Plan” button to test the extraction plan entered. The results of the query run should appear. The entries for the “T3Fieldone” in Table “Data3” will be mapped to the “Effect” parameter. A table should appear with “Effect” as the label and the numbers, 22, 33, and 44 listed below it. Click on the button “Save Extraction Plan” to save the extraction plan. A message indicating that the save was successful should appear. This completes the process for mapping a database and creating an extraction plan. The browser can now be exited to end the test case.

5.1.2.1.5 Results

- 1) The DOT connected to the Access 2000 database.
- 2) The DOT was accessible on a server via the Internet.
- 3) The DOT asked for user login credentials for security.
- 4) The DOT provided a web-page interface for creating stored execution plans.
- 5) The DOT provided a web-page interface for creating and storing ODBC-connection information for the source database.
- 6) The DOT provided a web-page interface for creating and storing DIC-connection information for FRAMES datasets.
- 7) The DOT provided the capability to test and modify extraction plans.
- 8) The DOT was used by the database owner and assisted in mapping the data from the source database into FRAMES dataset(s) outlined by the DIC file.
- 9) The DOT allowed the user to enter data necessary to specify the name and location of a database.
- 10) The DOT allowed the user to choose the FRAMES dataset DIC to map to (Note: No checking is performed to ensure that the DIC is in the correct format).
- 11) The DOT read a DIC file to determine parameter list.
- 12) The DOT read a DIC file to determine parameter correlations.
- 13) The DOT allowed the user to specify how parameters are mapped.
- 14) The DOT created an extraction plan(s) capable of retrieving a list.
- 15) The DOT operated within FRAMES under Windows-95 or higher.

5.1.2.2 DOT_02

5.1.2.2.1 Description and Rationale

This test case addresses the capability of the DOT to create and store an extraction plan that may have one or more key values associated with it. The DIC mapping used in test case DOT_01 will be used, but a separate extraction plan will be created for a parameter that has three key values.

5.1.2.2.2 Input Data

This test case requires that you use the dictionary file “EBF.DIC” and the database “DOTDB01.MDB.” Both of these files should be located on a computer that can be mapped and accessed through the Internet. All other input will be described in the procedure section below.

5.1.2.2.3 Expected Results

It is expected that the DOT will correctly store and execute an extraction plan for a parameter that has three key values associated with it.

The results should be presented in the format shown in Table 5.3:

Table 5.3. Expected Results DOT_02

LCChem	LCChemDur	LCChemCom
103	data3	203
104	data4	204

5.1.2.2.4 Procedure

Place the DOT server on a computer that can be accessed by the computer that will be used to carry out this test plan. Then bring up a browser (i.e., Netscape, Internet Explorer, etc.) and type in the URL to map the computer and folder location. If the DOT server is located on a computer that is behind a firewall, which is recommended, you will be prompted to enter a user ID and password. Enter those and click on the “Ok” button. The web page user interface should appear.

To create an extraction plan for the parameter:

To the left of the screen, there should be a list of choices entitled “Database Tools.” Click on the Database Owners Tool. On the screen to the right, use the list box to select the database that was mapped for test case DOT_01 and then click on the “Continue” button.

The “Extraction Plan - Parameter Selection” page should appear on the screen. Use the list box to select “LCChem, LCChemDur, LCChemCom - LC effect concentration values, LC effect duration, Comment on conditions of measurement for LC values” and then click on the “Continue” button.

The “Extraction Plan - Table Selection” page should appear. Click on the box to the left of “Data2” to select this as the table for data to be retrieved from. Then click on the “Continue” button.

The “Extraction Plan - Field Selection” page should appear. Click on the boxes to the left of “T2Field1,” “T2Field2,” and “T2Field3” to select these as the field data to be retrieved from. Then click on the “Continue” button.

The “Extraction Plan - Match Parameter with Database Field” page should appear; for Data2.T2Field1, select “LCChemCom”; for Data2.T2Field2, select “LCChemDur”; and for Data2.T2Field3, select “LCChem.” This indicates the three keys to use to map the DIC parameters to the database parameters. Then click on the “Continue” button.

The “Extraction Plan - Filter Selection” page should appear. For the key value “LifeFormSci,” select the field “Data2.ID” and the operator “>”; for the key value “ChemID,” select the field “Data2.T2Field4” and the operator “=”; for the key value “LCDescript,” select the field “Data2.T2Field5” and the operator “<=.” Then click on the “Continue” button.

The “Extraction Plan - Results” page should appear. Enter the following three key values in the three boxes: 1, 303, T2F5G. Click on the “Test Extraction Plan” button to test the extraction plan entered. The results of the query run should appear. A table should appear that looks similar to the one listed in Section 5.1.2.2.3. Click on the button “Save Extraction Plan” to save the extraction plan. A message indicating that the save was successful should appear. This completes the process for mapping a database and creating an extraction plan. The browser can now be exited to end the test case.

5.1.2.2.5 Results

The DOT allowed the user to enter in a parameter mapping with three key values, ran a test to ensure that the extraction plan worked properly, and saved the extraction plan.

5.1.2.3 DOT_03

5.1.2.3.1 Description and Rationale

This test case addresses the capability of the DOT to create and store an extraction plan that may have one or more key values associated with it. The DIC mapping used in test case DOT_01 will be used, but a separate extraction plan will be created for a parameter that has five key values. This test case also addresses the capability of the DOT to create and store an extraction plan that includes parameters from separate tables. The DOT should be capable of joining fields from separate tables to return the data requested by the extraction plan.

5.1.2.3.2 Input Data

This test case requires that you use the dictionary file “EBF.DIC” and the database “DOTDB01.MDB.” Both of these files should be located on a computer that can be mapped and accessed through the Internet. All other input will be described in the procedure section below.

5.1.2.3.3 Expected Results

It is expected that the DOT will correctly store and execute an extraction plan that includes parameters from two separated tables and join them to return the data requested by the extraction plan. It is also expected that the DOT will correctly store and execute an extraction plan for a parameter that has five key values associated with it.

The results should be presented in the format shown in Table 5.4:

Table 5.4. Expected Results DOT_03

ECChem	ECChemCom	ECChemDuration
data55	data5	55

5.1.2.3.4 Procedure

Place the DOT server on a computer that can be accessed by the computer that will be used to carry out this test plan. Then bring up a browser (i.e., Netscape, Internet Explorer, etc.) and type in the URL to map the computer and folder location. If the DOT server is located on a computer that is behind a firewall, which is recommended, you will be prompted to enter a user ID and password. Enter those and click on the “Ok” button. The web page user interface should appear.

To create an extraction plan for the parameter:

To the left of the screen there should be a list of choices entitled “Database Tools.” Click on the Database Owners Tool. On the screen to the right, use the list box to select the database that was mapped for test case DOT_01 and then click on the “Continue” button.

The “Extraction Plan - Parameter Selection” page should appear on the screen. Use the list box to select “ECChem, ECChemDur, ECChemCom - EC effect concentration values, EC effect duration, Comment on conditions of measurement for EC values” and then click on the “Continue” button.

The “Extraction Plan - Table Selection” page should appear. Click on the boxes to the left of “Data2” and “data1” to select them as the tables for data to be retrieved from. Then click on the “Continue” button.

The “Extraction Plan - Field Selection” page should appear. Click on the boxes to the left of “T2Field1,” “D1Field1,” and “D1Field3” to select these as the field data to be retrieved from. Then click on the “Continue” button.

The “Extraction Plan - Match Parameter with Database Field” page should appear; for Data2.T2Field1, select “ECChem”; for Data1.D1Field1, select “ECChemCom”; and for Data1.D1Field3, select “ECChemDur.” This indicates the three keys to use to map the DIC parameters to the database parameters. Then click on the “Continue” button.

The “Extraction Plan - Join Selection” page should appear. Under the box labeled “Left,” use the list box to select “data1.ID.” Under the box labeled “Operator,” use the list box to select “=.” Under the box labeled “Right,” use the list box to select “Data2.ID.” Then click on the “Add Join” button. The text “data1.ID = Data2.ID” should appear in the window below. Then click on the “Continue” button.

The “Extraction Plan - Filter Selection” page should appear. Use the list boxes to enter in the data shown in Table 5.5. Then click on the “Continue” button.

Table 5.5. Filter Selection Data For DOT_03

Key Value	Field	Operator
Effect	data1.ID	>
LifeFormSci	Data2.T2Field3	=
ChemID	data1.D1Field1	=
ECDescript	data1.D1Field4	=
Bodypart	data1.D1Field5	=

The “Extraction Plan - Results” page should appear. Enter the following three key values in the five boxes: data5, 205, 555, 5555, and 4. Click on the “Test Extraction Plan” button to test the extraction plan entered. The results of the query run should appear. A table should appear that looks similar to the one listed in Section 5.2.3.3. Click on the button “Save Extraction Plan” to save the extraction plan. A message indicating that the save was successful should appear. This completes the process for mapping a database and creating an extraction plan. The browser can now be exited to end the test case.

5.1.2.3.5 Results

The DOT correctly stored and executed an extraction plan that included parameters from two separated tables and joined them to return the data requested by the extraction plan. The DOT also allowed the user to enter in a parameter mapping with five key values, ran a test to ensure that the extraction plan worked properly, and saved the extraction plan.

5.1.2.4 DOT_04

5.1.2.4.1 Description and Rationale

This test case addresses the capability of the DOT to return only unique data when the “unique” data selection is made. An extraction plan will be run in which duplicate values are returned. The case will then be run a second time. The DIC mapping used in test case DOT_01 will be used, but a separate extraction plan will be created for a parameter that has six key values.

5.1.2.4.2 Input Data

This test case requires that you use the dictionary file “EBF.DIC” and the database “DOTDB01.MDB.” Both of these files should be located on a computer that can be mapped and accessed through the Internet. All other input will be described in the procedure section below.

5.1.2.4.3 Expected Results

It is expected that the DOT will correctly return only unique data when the “unique” data selection is made. The results of the extraction plan run without the “unique” selection made should be as shown in Table 5.6:

Table 5.6. Expected Results for Query Without “Unique” Selection

ChemName	ChemID
data1	11
data2	22
data3	33
data4	44
data5	55
data1	11
data8	88
data8	88
data9	99

The results of the extraction plan run with the “unique” selection made should be as shown in Table 5.7. Note that the duplicate rows for data1 and data8 do not appear.

Table 5.7. Expected Results For Query With ‘unique’ Selection

ChemName	ChemID
data1	11
data2	22
data3	33
data4	44
data5	55
data8	88
data9	99

5.1.2.4.4 Procedure

Place the DOT server on a computer that can be accessed by the computer that will be used to carry out this test plan. Then bring up a browser (i.e., Netscape, Internet Explorer, etc.) and type in the URL to map the computer and folder location. If the DOT server is located on a computer that is behind a firewall, which is recommended, you will be prompted to enter a user ID and password. Enter those and click on the “Ok” button. The web page user interface should appear.

To create an extraction plan for the parameter:

To the left of the screen there should be a list of choices entitled “Database Tools.” Click on the Database Owners Tool. On the screen to the right, use the list box to select the database that was mapped for test case DOT_01 and then click on the “Continue” button.

The “Extraction Plan - Parameter Selection” page should appear on the screen. Use the list box to select “ChemID, ChemName - Chemical CAS ID, Chemical common name” and then click on the “Continue” button.

The “Extraction Plan - Table Selection” page should appear. Click on the box to the left of “data1” to select it as the table for data to be retrieved from. Then click on the “Continue” button.

The “Extraction Plan - Field Selection” page should appear. Click on the boxes to the left of “D1Field1” and “D1Field3” to select these as the field data to be retrieved from. Then click on the “Continue” button.

The “Extraction Plan - Match Parameter with Database Field” page should appear; for data1.D1Field1, select “ChemName” and for data1.D1Field3, select “ChemID.” This indicates the two database parameters that will be mapped to the DIC parameters. Then click on the “Continue” button.

The “Extraction Plan - Filter Selection” page should appear. No input is needed on this page. Click on the “Continue” button.

The “Extraction Plan - Results” page should appear. Click on the “Test Extraction Plan” button to test the extraction plan entered. The results of the query run should appear. The results of the query run should appear. A table should appear that looks similar to the one listed in Section 5.2.4.3 for the extraction plan without the “unique” selection.

You will now re-do the extraction plan with the “unique” selection. Click on the “Back” button on your browser to move to the previous page, which should be “Extraction Plan - Results.” Click on the “Back” button on your browser again to move to the previous page, which should be “Extraction Plan - Filter Selection.” In the middle of this page, there is text that states “Click here if you would like the results to be distinct.” Click in the box to the right of this text. A check should appear in the box. Click on the “Continue” button to get to the next page. Then click on the “Test Extraction Plan” button again to test the extraction plan entered. The results of the query run should appear. A table should appear that looks similar to the one listed in Section 5.2.4.3 for the extraction plan with the “unique” selection.

Click on the button “Save Extraction Plan” to save the extraction plan. A message indicating that the save was successful should appear. This completes the process for mapping a database and creating an extraction plan. The browser can now be exited to end the test case.

5.1.2.4.5 Results

The DOT returned only unique values for an extraction plan when the “unique” selection was made.

5.1.2.5 DOT_05

5.1.2.5.1 Description and Rationale

This test case addresses the capability of the DOT to provide error messages to the user when the DIC file specified or the database file specified cannot be located. This may be caused by typing in the file names or file path incorrectly. In this case, the file path to the DIC file and database file will be entered incorrectly. The DOT should provide appropriate error messages. It should be noted that the DOT does not check that the file it finds is formatted according to the FRAMES specifications. If it finds a file that is mapped, it will try to parse it and display the contents to the screen. If the file is not in the correct format, the data written to the screen will be incorrect, and it should be apparent to the database owner.

5.1.2.5.2 Input Data

All input will be described in the procedure section below.

5.1.2.5.3 Expected Results

It is expected that the DOT will provide appropriate error messages when the DIC file or database file cannot be found.

5.1.2.5.4 Procedure

Place the DOT server on a computer that can be accessed by the computer that will be used to carry out this test plan. Then bring up a browser (i.e., Netscape, Internet Explorer, etc.) and type in the URL to map the computer and folder location. If the DOT server is located on a computer that is behind a firewall, which is recommended, you will be prompted to enter a user ID and password. Enter those and click on the “Ok” button. The web page user interface should appear. The following sections will describe how to test for the error messages.

DIC error message:

To the left of the screen there should be a list of choices entitled “Database Tools.” Click on the Dictionary Parsing Tool. On the screen to the right, click on the “Parse A New Dictionary” button. You will be prompted to fill in the network location, name, short description, and your company name. Enter the full network path and dictionary file name. The dictionary file name will be “EBF2.DIC.” For the second box, the name box, enter in the name DOTtest. The short description and company name boxes are not required information and can be left blank. Click the “Continue” button. A screen should appear with the text “The dictionary file was not found!” and an option should be provided to re-enter the dictionary information. This completes the process for checking for the DIC error message.

Database error message:

To the left of the screen there should be a list of choices entitled “Database Tools.” Click on the Database Owners Tool. On the screen to the right, use the list box to select “<new>” and then click on the “Continue” button. You will then be prompted to enter in the database information. In the first box, select “Microsoft Access 2000 (.mdb file).” In the next box, enter the full network path and the name of the database, which is “temp.mdb.” In the next box, enter “Temp.” The next four boxes can be left blank. In the last box, use the list box to select “DOTtest,” which is the DIC file that you just mapped. Then click on the “Continue” button. A screen should appear with text indicating that the database file could not be found. This completes the process for checking for the database error message.

5.1.2.5.5 Results

The DOT provided appropriate error messages when it could not locate a DIC file or a database file.

5.1.2.6 DOT_06

5.1.2.6.1 Description and Rationale

This test case addresses the capability of the DOT to correctly use filters in an extraction plan. The DIC mapping used in test case DOT_01 will be used, but a separate extraction plan will be created for a parameter that has six key values.

5.1.2.6.2 Input Data

This test case requires that you use the dictionary file “EBF.DIC” and the database “DOTDB01.MDB.” Both of these files should be located on a computer that can be mapped and accessed through the Internet. All other input will be described in the procedure section below.

5.1.2.6.3 Expected Results

It is expected that the DOT will correctly use filters in an extraction plan (see Table 5.8).

Table 5.8. Expected Results For Query With a Filter

Bodypart
103
104
105
106

5.1.2.6.4 Procedure

Place the DOT server on a computer that can be accessed by the computer that will be used to carry out this test plan. Then bring up a browser (i.e., Netscape, Internet Explorer, etc.) and type in the URL to map the computer and folder location. If the DOT server is located on a computer that is behind a firewall, which is recommended, you will be prompted to enter a user ID and password. Enter those and click on the “Ok” button. The web page user interface should appear.

To create an extraction plan for the parameter:

To the left of the screen, there should be a list of choices entitled “Database Tools.” Click on the Database Owners Tool. On the screen to the right, use the list box to select the database that was mapped for test case DOT_01 and then click on the “Continue” button.

The “Extraction Plan - Parameter Selection” page should appear on the screen. Use the list box to select “Bodypart - Body part description” and then click on the “Continue” button.

The “Extraction Plan - Table Selection” page should appear. Click on the box to the left of “data2” to select it as the table for data to be retrieved from. Then click on the “Continue” button.

The “Extraction Plan - Field Selection” page should appear. Click on the box to the left of “T2Field2” to select it as the field data to be retrieved from. Then click on the “Continue” button.

The “Extraction Plan - Match Parameter with Database Field” page should appear. For data2.T2Field2, select “Bodypart.” This indicates the database parameter that will be mapped to the DIC parameter. Then click on the “Continue” button.

The “Extraction Plan - Filter Selection” page should appear. Enter the text “data2.t2field2 > 102” in the window provided. Then click on the “Continue” button.

The “Extraction Plan - Results” page should appear. Click on the “Test Extraction Plan” button to test the extraction plan entered. The results of the query run should appear. The results of the query run should appear. A table should appear that looks similar to the one listed in Section 5.1.2.6.3 for the extraction plan without the “unique” selection.

Click on the button “Save Extraction Plan” to save the extraction plan. A message indicating that the save was successful should appear. This completes the process for mapping a database and creating an extraction plan. The browser can now be exited to end the test case.

5.1.2.6.5 Results

The DOT allowed the user to enter in a parameter mapping with six key values, ran a test to ensure that the extraction plan worked properly, and saved the extraction plan.

5.2 DET

The DET is the tool that actually implements the database extraction plan that is generated by the DOT. It is responsible for sending the data query from FRAMES to the database, extracting the pertinent data, and returning it to FRAMES for use by the requesting model. (Note: The DOT testing must be completed before to the DET testing. The DET testing requires the extraction plans created during the DOT testing.)

5.2.1 Summary of Requirements

Requirements for the DET are described in Section 2.0. of this document. These requirements were reworded, as necessary, to create a list of fundamental requirements that are suitable for testing. Table 5.1 lists these requirements. To ensure that the DET meets the requirements listed in Table 5.9, test cases were developed to check performance. Table 5.10 shows the relationship between these requirements and the test cases described in Section 5.2.2.

Table 5.9. Fundamental Requirements for Testing the DET

Requirement Number	Requirement
1	The DET shall connect to any MS ACCESS 97 or ACCESS 2000 database.
2	The DET shall execute an extraction plan to retrieve data from the associated database.
3	The DET shall format extracted data so that it is usable by FRAMES.
4	The DET shall use key values that will be provided in requests for data and will act as filters for the extracted data. Keys are substituted in for the replacable parameters by the DET when an extraction is executed.
5	The DET must operate within FRAMES under Windows-95 or higher.

Table 5.10. Relationship Between Test Cases and DET Fundamental Requirements

		Test Case Name (DET_xx)		
		01	02	03
Requirements	1	X		
	2	X		X
	3	X		
	4		X	
	5	X		

5.2.2 TEST CASES

5.2.2.1 DET_01

5.2.2.1.1 Description and Rationale

This test case addresses the capability of the DET to execute an extraction plan for a requested parameter with one key value. This test case also addresses the basic functionality of the DET; therefore, several of the other requirements are addressed in this case. The requirements addressed are listed below. These requirement numbers are referenced in the results section (Note: the requirement numbers used below do not correspond to the requirement numbers listed in Table 5.9).

- 1) The DET shall connect to any MS Access 97 or Access 2000 database.
- 2) The DET shall execute an extraction plan to retrieve data from the associated database.
- 3) The DET shall format extracted data so that it is usable by FRAMES.
- 4) The DET must operate within FRAMES under Windows-95 or higher.

5.2.2.1.2 Input Data

This test case requires that you use the dictionary file "EBF.DIC," the database "DOTDB01.MDB," and the extraction plan created during the DOT testing test case DOT_01. As in the DOT testing, both of the files should be located on a computer that can be mapped and accessed through the Internet. All other input will be described in the procedure section below.

5.2.2.1.3 Expected Results

It is expected that the DET will execute without error and will meet all of the requirements listed in Section 5.2.2.1.1.

The results returned should be as shown in Table 5.11.

Table 5.11. Expected Results For Query with One Key Value

Effect
22
33
44

5.2.2.1.4 Procedure

Bring up a browser (i.e., Netscape, Internet Explorer, etc.) and type in the URL below:

<http://wd23657/frames/extractdata.asp?step=extract&DD=edf.dic&DB=??¶meter='Effect'&keycount=1&key1=Bodypart&value1=22>

<http://wd23657/frames/extractdata.asp?step=extract&DD=edf.dic&DB=??¶meter='Effect'&keycount=0>

Note that in the middle of the URL, there is the text “DB=??.” Replace the “??” with the actual database number. You can find this number by bringing up the DOT and reviewing the extraction plan created for test case DOT_01. The database number is listed on most of the pages after the text “Database ID Selected.”

This URL includes the path map and the specifics for the parameter that is being requested. The parameter being requested is “,” which was mapped in the test case DOT_01.

5.2.2.1.5 Results

- 1) The DET connected to an MS Access 2000 database.
- 2) The DET executed an extraction plan to retrieve data for a parameter with no key values from the associated database.
- 3) The DET formatted extracted data so that it is usable by FRAMES.
- 4) The DET operated within FRAMES under Windows-95 or higher.

5.2.2.2 DET_02

5.2.2.2.1 Description and Rationale

This test case addresses the capability of the DET to execute an extraction plan for a requested parameter with five key values. The DIC mapping used in test case DOT_03 will be used.

5.2.2.2.2 Input Data

This test case requires that you use the dictionary file “EBF.DIC,” the database “DOTDB01.MDB,” and the extraction plan created during the DOT testing test case DOT_03. As in the DOT testing, both of the files should be located on a computer that can be mapped and accessed through the Internet. All other input will be described in the procedure section below.

5.2.2.2.3 Expected Results

It is expected that the DET will execute without error and will return the results listed in Table 5.12.

Table 5.12. Expected Results For Query With Five Key Values

ECChemCo m	ECChemDu r	ECChe m
data5	55	data55

5.2.2.2.4 Procedure

Bring up a browser (i.e., Netscape, Internet Explorer, etc.) and type in the URL below:

<http://wd23657/frames/extractdata.asp?step=extract&DD=edf.dic&DB=?¶meter='ECChem'&keycount=5&key1=Effect&key2=LifeFormSci&key3=ChemID&key4=ECDescript&key5=Bodypart&value1=4&value2=205&value3=data5&value4=555&value5=5555>

Note that in the middle of the URL, there is the text “DB=?.” Replace the “?” with the actual database number. You can find this number by bringing up the DOT and reviewing the extraction plan created for test case DOT_01. The database number is listed on most of the pages after the text “Database ID Selected.”

This URL includes the path map and the specifics for the parameter that is being requested. The parameter being requested is “,” which was mapped in the test case DOT_01.

5.2.2.2.5 Results

The DET allowed the user to enter in a parameter mapping with five key values, ran the extraction plan, and displayed the results to the screen. The results should be as listed below.

1 ECChemCom,ECChemDur,ECChem “data5”,“55”,“data55”

5.2.2.3 DET_03

5.2.2.3.1 Description and Rationale

This test case addresses the capability of the DET to execute an extraction plan for a requested parameter with five key values. The DIC mapping used in test case DOT_03 will be used, but a separate request will be created for a parameter using only three of the possible five key values. The two key parameters that are not specified should be returned with the results.

5.2.2.3.2 Input Data

This test case requires that you use the dictionary file “EBF.DIC,” the database “DOTDB01.MDB,” and the extraction plan created during the DOT testing test case DOT_03. As in the DOT testing, both of the files should be located on a computer that can be mapped and accessed through the Internet. All other input will be described in the procedure section below.

5.2.2.3.3 Expected Results

It is expected that the DET will execute without error and will return the results listed in Table 5.13.

Table 5.13. Expected Results for Five Key Query With Only Three Keys Specified

ECChemCom	ECChemDur	ECChem	ECDescript	Bodypart
data5	55	data55	555	5555

5.2.2.3.4 Procedure

Bring up a browser (i.e., Netscape, Internet Explorer, etc.) and type in the URL below:

<http://wd23657/frames/extractdata.asp?step=extract&DD=edf.dic&DB=??¶meter='ECChem'&keycount=3&key1=Effect&key2=LifeFormSci&key3=ChemID&value1=4&value2=205&value3=data5>

Note that in the middle of the URL there is the text “DB=??.” Replace the “??” with the actual database number. You can find this number by bringing up the DOT and reviewing the extraction plan created for test case DOT_03. The database number is listed on most of the pages after the text “Database ID Selected.”

This URL includes the path map and the specifics for the parameter that is being requested. The parameter being requested is “,” which was mapped in the test case DOT_03.

5.2.2.3.5 Results

The DET allowed the user to enter in a parameter mapping with three of the five key values, ran the extraction plan, and displayed the results to the screen. The results should be as listed below.

1 ECChemCom,ECChemDur,ECChem,ECDescript,Bodypart “data5”,“55”,“data55”,“555”,“5555”

5.3 DCE

The DCE allows the user to specify what data are being requested from the database. The DCE makes calls to the DET and DOT to obtain the requested data and provide it for the requesting model. (Note: The DOT testing must be completed before the DCE testing. The DCE testing requires the extraction plans created during the DOT testing.)

5.3.1 Summary of Requirements

Requirements for the DCE are described in Section 2.0. of this document. These requirements were reworded, as necessary, to create a list of fundamental requirements that are suitable for testing. Table 5.14 lists these requirements. To ensure that the DCE meets the requirements listed in Table 5.14, a test case was developed to check performance.

Table 5.14. Fundamental Requirements for Testing the DCE

Requirement Number	Requirement
1	The DCE shall operate within FRAMES under Windows-95 or higher.
2	The DCE shall issue extraction queries to DET via HTTP.
3	The DCE shall extract all data from Eco Benchmark database via DET based on chemical and species alias selections.
4	The DCE shall display extracted data in user interface.

Requirement Number	Requirement
5	The DCE shall include a DES file for use in FRAMES.
6	The DCE shall read a GID file, specifically the CON section of a GID file to determine chemicals of concern.
7	The DCE shall read the dataset dictionary file (.DIC) to determine dataset parameters to retrieve.
8	The DCE shall retrieve a list of lifeforms from the Eco Benchmark database via DET and display for user selection.
9	The DCE shall retrieve a list of chemicals from the Eco Benchmark database via DET and display for user alias selection.
10	The DCE shall allow the user to alias one lifeform to another.
11	The DCE shall allow the user to alias one chemical to another.
12	The DCE shall allow the user to edit the data retrieved.
13	The DCE shall write an ECO GID section that contains the extracted parameters names and values.
14	The DCE shall create an error file on start up.
15	The DCE shall delete the error if there are no errors.

5.3.2 TEST CASE

5.3.2.1 DCE_01

5.3.2.1.1 Description and Rationale

This test case addresses the capability of the DCE to allow the user to specify what data are being requested from the database. The DCE makes calls to the DET and DOT to obtain the requested data and provide it for the requesting model. This test case also addresses the basic functionality of the DOT; therefore, several of the other requirements are addressed in this case. The requirements addressed are listed below. These requirement numbers are referenced in the results section.

- 1) The DCE shall operate within FRAMES under Windows-95 or higher.
- 2) The DCE shall issue extraction queries to DET via HTTP.
- 3) The DCE shall extract all data from Eco Benchmark database via DET based on chemical and species alias selections.
- 4) The DCE shall display extracted data in user interface.
- 5) The DCE shall include a DES file for use in FRAMES.
- 6) The DCE shall read a GID file, specifically the CON section of a GID file, to determine chemicals of concern.
- 7) The DCE shall read the dataset dictionary file (.DIC) to determine dataset parameters to retrieve.
- 8) The DCE shall retrieve a list of lifeforms from the Eco Benchmark database via DET and display for user selection.
- 9) The DCE shall retrieve a list of chemicals from the Eco Benchmark database via DET and display for user alias selection.

- 10) The DCE shall allow the user to alias one lifeform to another.
- 11) The DCE shall allow the user to alias one chemical to another.
- 12) The DCE shall allow the user to edit the data retrieved.
- 13) The DCE shall write an ECO GID section that contains the extracted parameters names and values.
- 14) The DCE shall create an error file on start up.
- 15) The DCE shall delete the error if there are no errors.

5.3.2.1.2 Input Data

This test case requires that you use the dictionary file “EBF.DIC” and the database “DOTDB01.MDB.” Both of these files should be located on a computer that can be mapped and accessed through the Internet. All other input will be described in the procedure section below.

5.3.2.1.3 Expected Results

It is expected that the DCE will execute without error and will meet all of the requirements listed in Section 5.3.2.1.1.

5.3.2.1.4 Procedure

Setting up the conceptual site model for the scenario:

Bring up the FRAMES User Interface (FUI). Then add a Contaminant icon and an Eco Benchmarks icon to the working space window to the right. Connect the two icons. The connection should go from the Contaminant icon to the Eco Benchmarks icon. To select the contaminant model, right click on the Contaminant icon and select General Info. Click on FRAMES Default Chemical Database Selection and type in “Contaminants” in the box to the right of the word “Label” above. Then click on the “Ok” button. At this point, you have selected the FRAMES contaminant database to use in this case. You will next select your contaminant and name and save the file. Right click on the Contaminant icon and select User Input. You will be prompted to name the file and save it to a folder. Create a folder called “DCE_01” under the FRAMES folder, and then name the file “DCE_01.GID” and save it to the newly created folder. The user interface for the FRAMES database will appear on the screen. Select the constituent “Fluoranthene,” and then save and exit. The signal light on the contaminant icon should be green.

You will next select an Eco Benchmarks model and enter its associated data through its user interface, which is the DCE. Right click on the Eco Benchmarks icon and select General Info. Click on “ERED Database” and type ERED in the box to the right of the word “Label” above. Then click on the “OK” button. At this point, you have selected the ERED database to use in this case. You will next enter the data for the database. Right click on the Eco Benchmarks icon and select “Run Model.” You will be prompted to enter a user ID and password. Enter those and click on the “Ok” button. The DCE user interface should appear.

Entering data through the DCE

A tree-view options menu will appear on the left side of the screen. Click on “Chemical Aliases” to select the chemical mapping. In the lower window entitled “Selected Chemicals of concern for this Assessment,” click on the number “206440,” which is the CAS ID for fluoranthene. The number “206-44-0” should appear in the upper window. This indicates which chemical has been mapped to the chemical of concern.

You will next select the lifeform and its alias. Click on “LifeForm Selection.” A list of lifeforms available in the ERED database will appear. Select “Asterias rubens.” Then click on “LifeForm Aliases” on the menu to the left. In the lower window entitled “Selected Lifeforms of concern for this Assessment,” click on the

text “Asterias Rubens.” The text “Asterias Rubens” should appear in the upper window. This indicates which lifeform has been mapped to the lifeform of concern.

You will next use the DCE to retrieve the data from the ERED database for the lifeform and chemical selected. Click on “Viewing and Editing (ERED) Data Associated with Mapped Chemical(s) and Lifeform(s).” On the screen to the right, click on the button “Retrieve Data.” Progress bars will appear to provide a status of the database querying. After the querying is complete, you can review the data retrieved.

Before exiting, check your FRAMES directory for a file with an ERR extension. This is the error file created by the DCE. If an error occurs during execution, a message will be written to this file. If no error occurs, this file should be deleted after the DCE is closed down. Use the File menu to save and exit the DCE.

5.3.2.1.5 Results

- 1) The DCE operated within FRAMES under Windows-95.
- 2) The DCE issued an extraction query to the DET via HTTP
- 3) The DCE extracted all data from Eco Benchmark database via DET based on chemical and species alias selections
- 4) The DCE displayed extracted data in the user interface
- 5) The DCE included a DES file for use in FRAMES
- 6) The DCE read a GID file, specifically the CON section of a GID file to determine chemicals of concern
- 7) The DCE read the dataset dictionary file (.DIC) to determine dataset parameters to retrieve
- 8) The DCE retrieved a list of lifeforms from the Eco Benchmark database via DET and displayed it for user selection
- 9) The DCE retrieved a list of chemicals from the Eco Benchmark database via DET and displayed it for user alias selection
- 10) The DCE allowed the user to alias one lifeform to another
- 11) The DCE allowed the user to alias one chemical to another
- 12) The DCE allow the user to edit the data retrieved
- 13) The DCE wrote an ECO GID section that contains the extracted parameters names and values
- 14) The DCE created an error file on start up
- 15) The DCE deleted the error when there was no error

6.0 Quality Assurance Program

Updates to the FRAMES technology software system will be developed under a quality assurance program documented in Gelston et al. (1998). Quality is defined as the capability of the software to meet client needs. Meeting client needs starts with a shared understanding of how the software must perform and continues throughout the software life cycle of design, development, testing, and implementation through attention to details.

Figure 6.1 outlines the software-development process that will be used for the FRAMES technology software system, highlighting the quality check points. The FRAMES technology software system activities flow down the left side of Figure 6.1 because it is software developed for the first time as opposed to a modification to existing software. The process shown is designed for compatibility with similar processes used by other government agencies. For example, this quality process compares favorably with that in U.S. Environmental Protection Agency (EPA) Directive 2182, *System Design and Development Guidance* (EPA 1997). It also compares favorably with the Office of Civilian Radioactive Waste Management's *Quality Assurance Requirements and Description, Supplement I, Software* (OCRWM 1995). Activities roughly equivalent across these processes are shown in Table 6.1.

Development of the FRAMES technology software system includes the implementation of a quality assurance checklist (see Figure 6.2). All team members understand the component requirements and design necessary to ensure quality. Completion of this checklist verifies that all documentation will be complete for transfer of the software to client use.

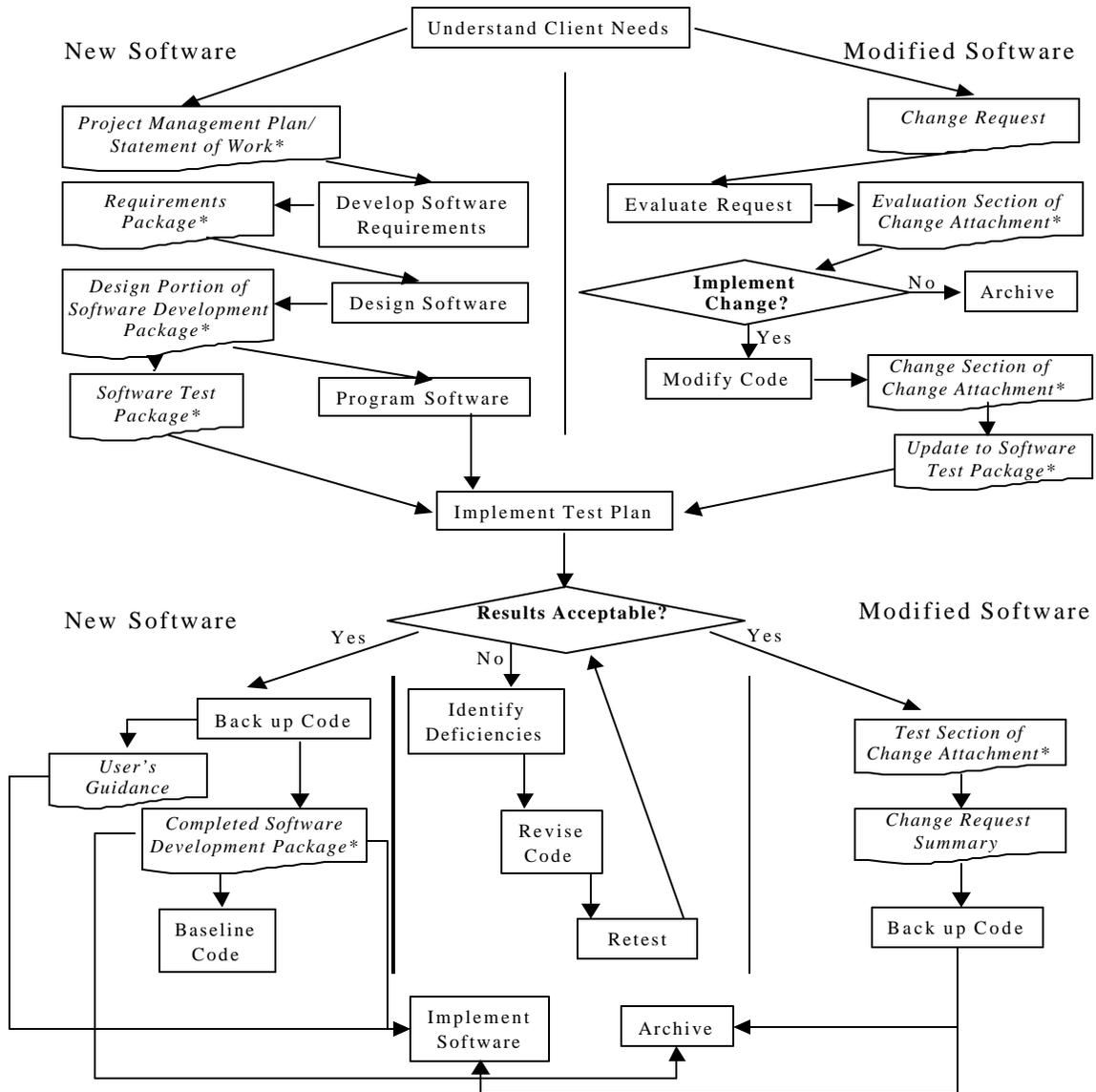


Figure 6.1. Ensuring Quality in the Environmental Software Development Process (* indicates quality review stage; box with waver bottom line and italics font indicates a document rather than an activity)

Table 6.1. Relationship of Pacific Northwest National Laboratory (PNNL) Environmental Software Development Process to Quality Assurance Requirements (OCRWM 1995; EPA 1997)

OCRWM Quality Assurance Requirement^(a)	EPA Essential Element of Information^(b)	Environmental Software Process Equivalent (Section)
	4—System Implementation Plan	Project Management Plan or Statement of Work
I.2.5A Functional Requirements Information Documentation; I.2.5C Requirements and Design Documentation	5—System Detailed Requirements Document	Requirements Package
I.2.1 Software Life Cycles, Baselines (see their Appendix C), and Controls	6—Software Management Plan	Project Management Plan or Statement of Work and Gelston et al. (1998)
I.2.2 Software Verification ^(c) and Software Validation; I.2.4 Software Validation ^(d)	7—Software Test and Acceptance Plan	Software Test Package
I.2.3 Software Verification; I.2.5C Requirements and Design Information Documentation	8—Software Design Document	Design Portion of Software Development Package
I.2.6A Configuration Identification		Completed Software
I.2.6B Configuration Control; I.2.6C Configuration Status; I.2.7 Defect Reporting and Resolution ^(e)	9—Software Maintenance Document	Modification Documentation
	10—Software Operations Document	User's Guidance and Training
I.2.5B User Information Documentation	11—Software User's Reference Guide	User's Guidance and Training
	12—System Integration Test Reports	Software Test Package

- (a) Note that OCRWM requirement I.2.8, Control of the Use of Software, is the responsibility of the OCRWM-related client.
- (b) Elements 1 through 3 are generally completed by clients in the U.S. Environmental Protection Agency before contract initiation with the project team.
- (c) Verification includes informal code testing by software engineers to ensure that code functions as required.
- (d) Validation includes testing by those other than the software engineers who developed the code to provide an independent confirmation that software functions as required.
- (e) Note that some changes requested by clients may not be made in the software unless funding has been allocated for such modifications.

- A. General Requirements Analysis
 - Documented in
 - Statement of Work
 - Contains information on (all of the following)
 - problem description
 - deliverables
 - project team
 - capabilities to be used
 - restrictions
 - difficulties envisioned
 - compatibilities with existing software/hardware
 - scope of the project
- B. Specific Requirements Analysis
 - Documented in
 - requirements section of documentation.
 - Contains information on (all of the following)
 - purpose of the software
 - structure of the software
 - hardware and software requirements
 - input and output requirements
 - scientific basis
 - assumptions
 - limitations
- C. Design Documentation
 - Documented in
 - design portion of documentation.
 - team task plans/Project Management Plan
 - Contains information on (all of the following)
 - code type and description
 - development team members
 - specifications
 - logic diagrams
 - "help" descriptions
 - methods to ensure consistency in components
 - mathematical formulations
 - need for pre/post-processors
- D. Development Documentation
 - Documented in
 - Specifications Document
 - Quality Assurance Archive
 - Contains information on (all of the following)
 - baseline hard copy of the source code
 - diskette copy
 - name of computer language(s) used
- E. Testing Documentation
 - Documented in
 - test plan that meets quality assurance requirements
 - Contains information on (all of the following)
 - description of software
 - testing scope

Figure 6.2. Quality Assurance Implementation Checklist for the FRAMES Technology Software System

F. User's Guidance

- Documented in
- ___hardcopy printout of user's guidance for system
- Contains information on (all of the following)
- ___description of software
- ___description of use of user interface
- ___mathematical formulations
- ___example problems
- ___explanation of modules included

G. General Quality Assurance Documentation

- Documented in
- ___Quality Assurance Program Document
- ___Quality Assurance Software-Specific Checklist
- Contains information on (all of the following)
- ___purpose of quality assurance program
- ___client-specified activities
- ___activities required to ensure quality in software

H. Quality Assurance Archive

- Documented in
- ___hard-copy files
- ___back up disk files in multiple storage locations
- Contains information on (all of the following)
- ___all quality assurance documentation
- ___client correspondence regarding software
- ___modifications made to baselined software
- ___disk copy back ups
- ___reproducibility of code (check code for comments)

Completed by _____ Date _____

Approved by _____ Date _____

System/Module Manager _____

Figure 6.2. Quality Assurance Implementation Checklist for FRAMES Technology Software System (contd)

7.0 References

Gelston, G. M., R. E. Lundgren, J. P. McDonald, and B. L. Hoopes. 1998. *An Approach to Ensuring Quality in Environmental Software*, PNNL-11880, Pacific Northwest National Laboratory, Richland, Washington.

Office of Civilian Radioactive Waste Management (OCRWM). 1995. *Quality Assurance Requirements and Description, Supplement I, Software*, U.S. Department of Energy, Washington, D.C.

U.S. Environmental Protection Agency (EPA). 1997. *System Design and Development Guidance*, EPA Directive Number 2182, Washington, D.C.

Whelan, G., K.J. Castleton, J.W. Buck, G.M. Gelston, B.L. Hoopes, M.A. Pelton, D.L. Strenge, and R.N. Kickert. 1997. *Concepts of a Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES)*, PNNL-11748, Pacific Northwest National Laboratory, Richland, Washington.

Appendix A
Ecological Benchmark File (EBF)

Appendix A: Ecological Benchmark File (EBF)

A.1 Ecological Benchmark Dictionary File (EBF.DIC)

51,EcoBenchMarks

Code,Dimensions,DataType,Min,Max,Units,Description,Key1,Key2,Key3,Key4,Key5,Key6

NumLifeForm,0,INTEGER,0,0,,"Number of life-forms"

LifeFormSci,1,STRING,0,80,,"Life-form scientific name (Genus species)",NumLifeForm

LifeformName,1,STRING,0,80,,"Life-form common name",NumLifeForm

akaLifeFormSci,1,STRING,0,80,,"Life-form scientific name (Genus species)",NumLifeForm

akaLifeFormName,1,STRING,0,80,,"Life-form common name",NumLifeForm

NumChems,0,INTEGER,0,,,"Number of chemicals",,

ChemID,1,STRING,0,20,,"Chemical CAS ID",NumChems

ChemName,1,STRING,0,80,,"Chemical common name",NumChems

NumLCDescript,0,INTEGER,0,,,"Number of LC descriptions"

LCDescript,1,STRING,0,80,,"Short description for LC percentages",NumLCDescript

NumLDDescript,0,INTEGER,0,,,"Number of LD descriptions"

LDDescript,1,STRING,0,80,,"Short description for LD percentages",NumLDDescript

NumECDescript,0,INTEGER,0,,,"Number of EC descriptions"

ECDescript,1,STRING,0,80,,"Short description for EC percentages",NumECDescript

NumEDDescript,0,INTEGER,0,,,"Number of ED descriptions"

EDDescript,1,STRING,0,80,,"Short description for ED percentages",NumEDDescript

NumBodypart,0,INTEGER,0,,,"Number of body part descriptions"

Bodypart,1,STRING,0,80,,"Body part description",NumBodyPart

NumEffect,0,INTEGER,0,,,"Number of effect types descriptions"

Effect,1,STRING,0,80,,"Description of effect",NumEffect

NumLCChem,3,INTEGER,0,,,"Number of LC values",ChemID,LifeFormSci,LCDescript

LCChemCom,4,STRING,0,100,,"Comment on conditions of measurement for LC values",ChemID,LifeFormSci,LCDescript,NumLCChem

L C C h e m , 4 , F L O A T , 0 , , " m g / k g " , " L C e f f e c t c o n c e n t r a t i o n values",ChemID,LifeFormSci,LCDescript,NumLCChem

LCChemDur,4,FLOAT,0,,"day",,"LC effect duration",ChemID,LifeFormSci,LCDescript,NumLCChem

NumLDChem,3,INTEGER,0,,,"Number of LD values",ChemID,LifeFormSci,LDDescript

LDChemCom,4,STRING,0,100,,"Comment on conditions of measurement for LD values",ChemID,LifeFormSci,LDDescript,NumLDChem

L D C h e m , 4 , F L O A T , 0 , , " m g / k g " , " L D e f f e c t c o n c e n t r a t i o n value",ChemID,LifeFormSci,LDDescript,NumLDChem

LDChemDur,4,FLOAT,0,,"day",,"LD effect duration",ChemID,LifeFormSci,LDDescript,NumLDChem

NumECChem,5,INTEGER,0,,,"Number of EC values",ChemID,LifeFormSci,ECDescript,Bodypart,Effect

ECChemCom,6,STRING,0,100,,"Comment on conditions of measurement for EC values",ChemID,LifeFormSci,ECDescript,Bodypart,Effect,NumECChem

E C C h e m , 6 , F L O A T , 0 , , " m g / k g " , " E C e f f e c t c o n c e n t r a t i o n values",ChemID,LifeFormSci,ECDescript,Bodypart,Effect,NumECChem

ECChemDur,6,FLOAT,0,, "day", "EC effect
 duration",ChemID,LifeFormSci,ECDescript,Bodypart,Effect,NumECChem
 NumEDChem,5,INTEGER,0,, "Number of ED values",ChemID,LifeFormSci,EDDescript,Bodypart,Effect
 EDChemCom,6,STRING,0,100,, "Comment on conditions of measurement for ED
 values",ChemID,LifeFormSci,EDDescript,Bodypart,Effect,NumEDChem
 EDChem,6,FLOAT,0,, "mg/kg", "ED effect concentration
 values",ChemID,LifeFormSci,EDDescript,Bodypart,Effect,NumEDChem
 EDChemDur,6,FLOAT,0,, "day", "ED effect
 duration",ChemID,LifeFormSci,EDDescript,Bodypart,Effect,NumEDChem
 LOELChemCom,4,STRING,0,100,, "Comment on conditions of measurement for LOEL
 values",ChemID,LifeFormSci,Bodypart,Effect
 LOELChem,4,FLOAT,0,, "mg/kg", "Lowest concentration to cause an
 effect",ChemID,LifeFormSci,Bodypart,Effect
 LOELChemDur,4,FLOAT,0,, "mg/kg", "Duration for lowest concentration to cause an
 effect",ChemID,LifeFormSci,Bodypart,Effect
 NOELChemCom,2,STRING,0,100,, "Comment on conditions of measurement for NOEL
 values",ChemID,LifeFormSci
 NOELChem,2,FLOAT,0,, "mg/kg", "Highest concentration not to cause an effect",ChemID,LifeFormSci
 NOELChemDur,2,FLOAT,0,, "mg/kg", "Duration for highest concentration not to cause an
 effect",ChemID,LifeFormSci
 ~Table,2,LifeFormSci,LifeFormName
 ~Table,2,ChemID,ChemName
 ~Table,3,LCChem,LCChemDur,LCChemCom
 ~Table,3,LDChem,LDChemDur,LDChemCom
 ~Table,3,ECChem,ECChemDur,ECChemCom
 ~Table,3,EDChem,EDChemDur,EDChemCom
 ~Table,3,NOELChem,NOELChemDur,NOELChemCom
 ~Table,3,LOELChem,NOELChemDur,LOELChemCom

A.2 Ecological Benchmarks File (EBF)

This file is stored as a section in the FRAMES GID file. The section name is EBF# where # is the identifying number. The specification for the EBF section is:

For each Parameter

Line with:

Parameter Name (string)
 Index 1 (integer)
 Index 2 (integer)
 Index 3 (integer)
 Index 4 (integer)
 Index 5 (integer)
 Index 6 (integer)
 Reference (integer)
 User Units (string)
 Units (string)

Value (string),(integer), or (float)

Example Lines

```
NumLifeform,0,0,0,0,0,0,,1
LifeFormSci,1,0,0,0,0,0,, "Brachionus calyciflorus"
LifeFormName,1,0,0,0,0,0,, "Rotifer"
akaLifeFormSci,1,0,0,0,0,0,, "Brachionus calyciflorus"
akaLifeFormName,1,0,0,0,0,0,, "Rotifer"
NumChems,0,0,0,0,0,0,,1
ChemID,1,0,0,0,0,0,, "71432"
ChemName,1,0,0,0,0,0,, "Benzene"
NumLCDescript,0,0,0,0,0,0,,2
LCDescript,1,0,0,0,0,0,, "LC50"
LCDescript,2,0,0,0,0,0,, "LC80"
```

All values are stored as text; this allows each module to use its own storage format (Big-Endian, Little-Endian, 32 bit real, 64 bit real ...). The number of digits of precision should be set by the model developer when floating point numbers are being written.