

US EPA ARCHIVE DOCUMENT

**DOCUMENTATION FOR THE  
FRAMES-HWIR TECHNOLOGY SOFTWARE  
SYSTEM, VOLUME 2:  
SYSTEM USER INTERFACE**

Project Officer  
and Technical Direction:

Mr. Gerard F. Laniak  
U.S. Environmental Protection Agency  
Office of Research and Development  
National Environmental Research Laboratory  
Athens, Georgia 30605

Prepared by:

Pacific Northwest National Laboratory  
Battelle Boulevard, P.O. Box 999  
Richland, Washington 99352  
Under EPA Reference Number DW89937333-01-0

U.S. Environmental Protection Agency  
Office of Research and Development  
Athens, Georgia 30605

October 1999

**DISCLAIMER**

This report was prepared as an account of work sponsored by the U.S. Environmental Protection Agency. Neither 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-AC06-76RLO 1830*



This document was printed on recycled paper.

(9/97)

## ACKNOWLEDGMENTS

A number of individuals have been involved with this effort. Mr. Gerard F. Laniak of the U.S. Environmental Protection Agency (EPA), Office of Research and Development, National Environmental Research Laboratory, Athens, Georgia, provided the overall technical direction and review throughout this work. This report was prepared by the Pacific Northwest National Laboratory<sup>1</sup> (PNNL) staff of John Buck, Mitch Pelton, Regina Lundgren, Karl Castleton, Gariann Gelston, Bonnie Hoopes, John McDonald, and Randal Taira. Additional PNNL staff supporting this effort include Wayne Cosby, Nancy Foote, Kristin Manke, Jill Pospical, Debbie Schulz, and Barbara Wilson. Useful inputs were provided by many U.S. EPA individuals working on the Hazardous Waste Identification Rule, including Messrs. Barnes Johnson, Stephen Kroner, and David Cozzie, and Drs. David Brown, Robert Ambrose, Zubair Saleem, Donna Schwede, and Sharon LeDuc, among many others.

---

<sup>1</sup>Operated by Battelle for the U.S. Department of Energy under Contract DE-AC06-76RLO 1830.

## Summary

The U.S. Environmental Protection Agency (EPA) is developing a comprehensive environmental exposure and risk analysis software system for agency-wide application. The software system will be applied to the technical assessment of exposures and risks relevant to the Hazardous Waste Identification Rule (HWIR). The software system adapted to automate this assessment is the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES), developed by the Pacific Northwest National Laboratory (PNNL). The process used to develop the FRAMES-HWIR Technology Software System includes steps for requirements analysis, design, specification, and development with testing and quality assurance composing a critical portion of each step. This report documents that process for one of the key components of the system, the System User Interface (SUI).

The SUI allows the user direct interaction with the FRAMES-HWIR Technology Software System and the other processes and modules associated with the system. The SUI will connect, interact with, and direct the other processors housed in the FRAMES-HWIR Technology Software System, managing the overall execution of the system. The interface will allow the user to 1) specify what mode of operation will be conducted; 2) specify the indices (e.g., source type chemical sites and concentration in waste level) for the different levels of analysis associated with the HWIR Assessment Strategy; 3) start, resume, and reset simulations; 4) specify where the results should be stored; and 5) specify what level of data storage is required (that is, minimum or maximum data storage).

The SUI manages the different processors, warning/error files, data files, and databases through the use of user-interactive computer screens. Associated with the SUI are three main screens: 1) SYSTEM CONFIGURATION, 2) SYSTEM MANAGEMENT, and 3) SYSTEM STATUS. These user interface screens are the main methods of interaction between the user and the FRAMES-HWIR Technology Software System. This processor:

1. Controls the execution of the FRAMES-HWIR Technology Software System processors.
2. Allows the FRAMES-HWIR Technology Software System to operate in several modes (such as, **Debug, Stop on Error, Stop on Warning**).
3. Has options through the SYSTEM MANAGEMENT SCREEN to control which data files from a simulation are stored.
4. Contains a SYSTEM CONFIGURATION SCREEN that identifies database and system executable locations to be accessed and directories into which FRAMES-HWIR Technology Software System files are stored.
5. Includes, as part of the SYSTEM STATUS SCREEN, the ability to allow the user to select indexes (e.g., source type chemical sites and concentration in waste level) for different levels of analysis to be executed for the simulation, as well as allow the user to determine the status of the simulations and to start, resume, and reset.

The SUI has been tested to ensure it meets the preceding requirements. The SUI was tested as a unit to ensure it performed as expected, and the test cases evaluated the capability of the SUI to meet the requirements that EPA requested. The SUI passed its tests. In addition, development of the SUI followed a quality assurance program designed to ensure the processor met EPA expectations.

## Acronyms and Abbreviations

COP	Computational Optimization Processor
$C_w$	concentration of chemical in waste stream
DSP	Distribution Statistics Processor
ELP I	Exit Level Processor I
ELP II	Exit Level Processor II
EPA	U.S. Environmental Protection Agency
FRAMES	Framework for Risk Analysis in Multimedia Environmental Systems
GRF	Global Results Files
HWIR	Hazardous Waste Identification Rule
MET	meteorological
MMSP	Multimedia Multipathway Simulation Processor
OCRWM	Office of Civilian Radioactive Waste Management
PNNL	Pacific Northwest National Laboratory
PSOF	Protective Summary Output File
RSOF	Risk Summary Output File
SDP	Site Definition Processor
SSF	Site Simulation Files
SUI	System User Interface

# Contents

Acknowledgments .....	iii
Summary .....	v
Acronyms and Abbreviations .....	vii
1.0 Introduction .....	1.1
2.0 Requirements for the System User Interface .....	2.1
2.1 Input Requirements for the System User Interface .....	2.2
2.1.1 System Configuration Screen .....	2.2
2.1.2 System Management Screen .....	2.6
2.1.3 System Status Screen .....	2.10
2.2 Scientific Requirements for the System User Interface .....	2.11
2.3 Output Requirements for the System User Interface .....	2.11
3.0 Design Elements .....	3.1
3.1 Input .....	3.1
3.2 Implementation .....	3.2
3.3 Output .....	3.2
4.0 Testing Approach and Results .....	4.1
4.1 Summary of Requirements .....	4.1
4.2 Test Cases .....	4.3
4.2.1 SUI_01 .....	4.3
4.2.2 SUI_02 .....	4.4
4.2.3 SUI_03 .....	4.5
4.2.4 SUI_04 .....	4.7
5.0 Quality Assurance Program .....	5.1
6.0 References .....	6.1



## Figures

1.1	Overview of the FRAMES-HWIR Technology Software System . . . . .	1.2
1.2	Details of the System User Interface Interactions with Other Components . . . . .	1.3
2.1	Database Subscreen of System Configuration Screen . . . . .	2.3
2.2	Directories Subscreen of System Configuration Screen . . . . .	2.4
2.3	Processors Subscreen of System Configuration Screen . . . . .	2.5
2.4	MMSP Source Modules Subscreen of System Configuration Screen . . . . .	2.5
2.5	MMSP Transport Modules Subscreen of System Configuration Screen . . . . .	2.7
2.6	MMSP Foodchain Modules Subscreen of System Configuration Screen . . . . .	2.7
2.7	MMSP Exposure/Risk Modules Subscreen of System Configuration Screen . . . . .	2.8
2.8	Selections Subscreen of the System Management Screen . . . . .	2.8
2.9	Options Subscreen of the System Management Screen . . . . .	2.9
2.10	System Status Screen . . . . .	2.10
5.1	Ensuring Quality in the Environmental Software Development Process . . . . .	5.2
5.2	Quality Assurance Implementation Checklist for the System User Interface . . . . .	5.4

## Tables

4.1	Testing Requirements for the System User Interface . . . . .	4.1
4.2	Relationship Between Test Cases and Requirements for the System User Interface . . . . .	4.2
5.1	Relationship of PNNL Environmental Software Development Process to Quality Assurance Requirements . . . . .	5.3

## 1.0 Introduction

The U.S. Environmental Protection Agency (EPA) is developing a comprehensive environmental exposure and risk analysis software system for agency-wide application. The software system is being prototyped (that is, initial design and implementation) for application to the technical assessment of exposures and risks relevant to the Hazardous Waste Identification Rule (HWIR). The HWIR is designed to determine quantitative criteria for allowing a specific class of industrial waste streams to no longer require disposal as a hazardous waste (that is, to exit Subtitle C) and allow disposal in Industrial Subtitle D facilities. Hazardous waste constituents with values less than these exit criteria levels would be reclassified as nonhazardous wastes under the Resource Conservation and Recovery Act.

The software system adapted to automate this assessment is the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES), developed by the Pacific Northwest National Laboratory (PNNL). The FRAMES-HWIR Technology Software System consists of a series of components within a system framework (Figure 1.1). The process used to develop the FRAMES-HWIR Technology Software System includes steps for requirements analysis, design, specification, and development with testing composing a critical portion of each step. One of the key components of the system is the System User Interface (SUI).

The interrelationships between input/output data files and data transfer, computational modules, the system processors, and the SUI are schematically illustrated in Figure 1.2. As shown in the figure, the SUI communicates risk simulation information from the user to the system processors and receives status information (warning and error files) from the system processors and modules. The SUI needs to know where the databases for the HWIR assessment are located. The SUI also specifies where all the FRAMES-HWIR Technology Software System files and executables for the processors and modules are located. The SUI allows the user to specify the number of realizations, sites, chemicals, source types, and waste levels, as well as the level of data storage for the simulation. The SUI also provides runtime information on the simulation, such as realization number and status of realization, and allows the user to start, reset, or resume the simulation.

This report includes information on requirements of the SUI and design elements necessary to meet those requirements. It also discusses testing plans, testing results, and the quality assurance program for the SUI. Specifications for the SUI are described in *Documentation of the FRAMES-HWIR Technology Software System, Volume 8: Specifications*. Other components developed by PNNL are described in companion documents as listed in the reference list; the system itself is documented in a summary report entitled *Overview of the FRAMES-HWIR Technology Software System*.

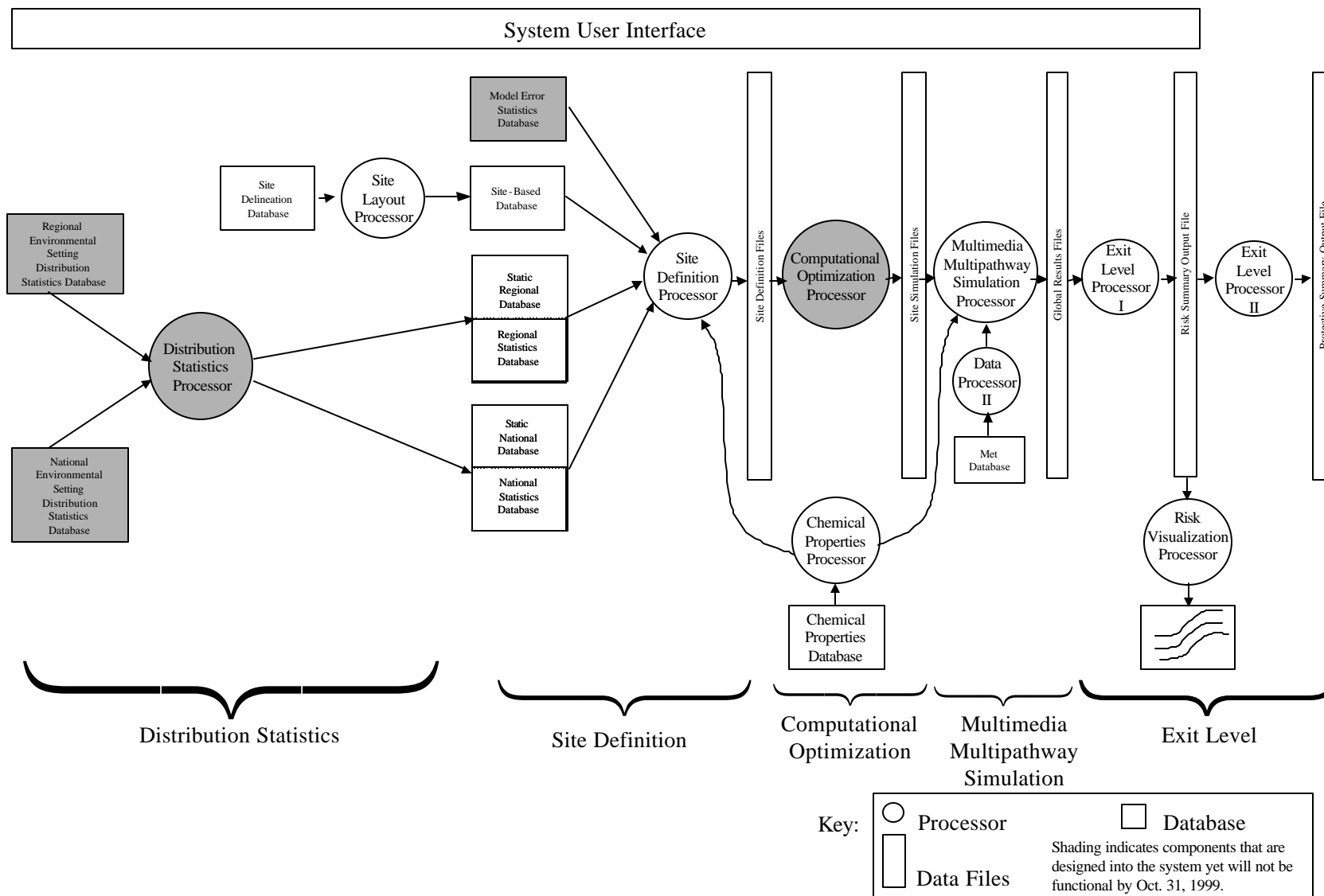


Figure 1.1 Overview of the FRAMES-HWIR Technology Software System

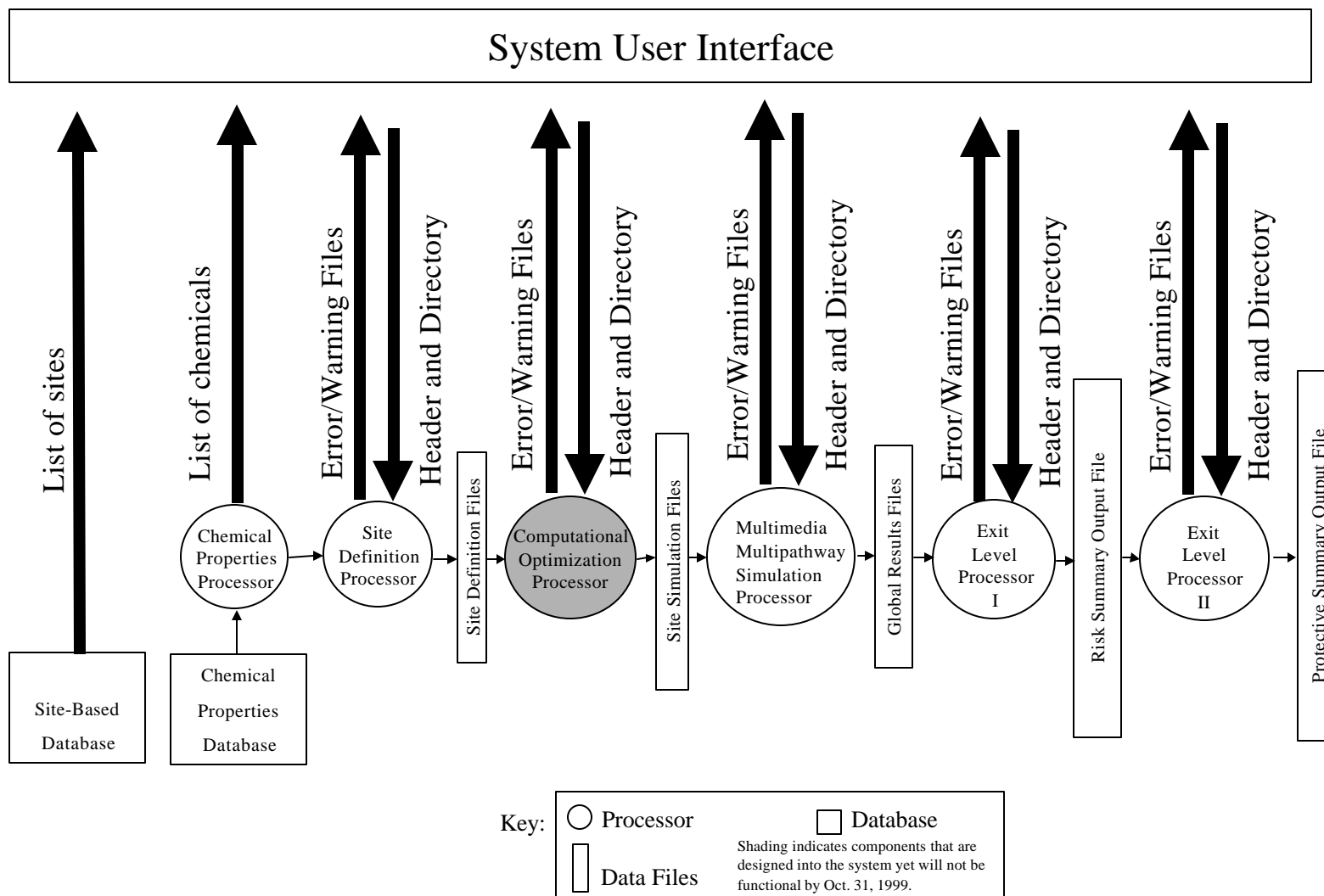


Figure 1.2 Details of the System User Interface Interactions with Other System Components

## 2.0 Requirements for the System User Interface

Requirements are characteristics and behaviors that a piece of software must possess to function adequately for its intended purpose. As mentioned, the SUI will connect, interact with, and direct the other processors housed in the FRAMES-HWIR Technology Software System, managing the overall execution of the software system. The interface will allow the user to 1) specify the mode of operation to be conducted; 2) specify the indexes for the different loops associated with the HWIR Assessment Strategy; 3) start, stop, reset, and resume simulations; 4) specify where the results are stored; and 5) specify what level of data storage is required (that is, only required data, all data generated, or some level in between).

Certain notational conventions are used throughout the rest of this volume. These conventions are defined in the following Hint box and provide helpful information on the operation of the FRAMES-HWIR Technology Software System.

In this document, all button names are **Boxed** and bolded, so you can easily identify them; file names are in italics; the name of a screen or subscreen is presented in small caps sans serif. Menu item names are denoted using bold-faced text type. Mode names (such as Stop on Error) appear on the screen in a font type that looks like this: **Stop on Error**. Status indicators, such as OFF and ON, are shown in the text as all caps with a serif typeface. Notes and their text are in bolded, italicized text and are usually boxed, as well.

The SUI manages the different processors, warning/error files, data files, and databases through the use of user-interactive computer screens: 1) SYSTEM CONFIGURATION, 2) SYSTEM MANAGEMENT, and 3) the SYSTEM STATUS screens. In summary, this processor

- 1) Controls the execution of the FRAMES-HWIR Technology Software System processors.
- 2) Allows the FRAMES-HWIR Technology Software System to operate in several modes (**Debug**, **Stop on Error**, or **Stop on Warning**).
- 3) Has options through the SYSTEM MANAGEMENT SCREEN to control which data files from a simulation are stored.
- 4) Contains a SYSTEM CONFIGURATION SCREEN that identifies database and system executable locations for access and directories into which FRAMES-HWIR Technology Software System files are stored.
- 5) Includes, as part of the SYSTEM STATUS SCREEN, the ability to allow the user to select indexes for different levels of analysis executed for the simulation, as well as allow the user to determine the status of the simulations, and allow the user to start, resume, and reset.

The following sections describe input, scientific, and output requirements for the SUI.


## 2.1 Input Requirements for the System User Interface

The SUI serves as the user interface for the entire FRAMES-HWIR Technology Software System. It also ensures a compatible file format throughout the system to facilitate system operation and communicates the status of the different processors and modules for a simulation (that is, current realization, warning, and error files). The SUI manages the different databases, data files, processors, and modules within the FRAMES-HWIR Technology Software System. The SUI is composed of three main screens: SYSTEM CONFIGURATION, SYSTEM MANAGEMENT, and SYSTEM STATUS screens described as follows.

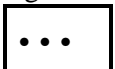
### 2.1.1 System Configuration Screen

The system configuration function is managed through the SYSTEM CONFIGURATION SCREEN of the SUI. This screen has several functions. In general, the SYSTEM CONFIGURATION SCREEN allows the user to select and define the system databases, data files, processors, and modules used for the simulation.

#### 2.1.1.1 Databases

To run a simulation, six databases must be available to the system software. These are the site-based, regional distribution, national distribution, static regional, static national, and site survey databases. The DATABASE subscreen of the SYSTEM CONFIGURATION SCREEN allows the user to specify where these databases are stored on the computer. (The formats of these databases are described in *Documentation of the FRAMES-HWIR Technology Software System, Volume 8: Specifications*; see Section 6.0 of this document.) These six databases are developed and populated by the EPA and must be installed on the computer before implementing this function. The user can click on the  button to select the correct databases. Figure 2.1 shows the database selection function of the SYSTEM CONFIGURATION SCREEN and the six databases that must be selected. The user must specify the drive and directories where these databases are located by using the SUI.

#### 2.1.1.2 Directories

The DIRECTORIES subscreen of the SYSTEM CONFIGURATION SCREEN also allows the user to specify where the system files are stored and where the chemical properties and meteorological (MET) databases are located. Figure 1.2 shows the relationship of these data files to the SUI and the Multimedia Multipathway Simulation Processor (MMSP). Figure 2.2 shows the DIRECTORIES subscreen of the SYSTEM CONFIGURATION SCREEN. The user can select the  button to specify the directories where these outputs files should be stored or type in the appropriate drive and directory (for example, C:\HWIR\OUTPUT). The key system data files that need to be defined are illustrated in Figure 2.2 and are as follows:

- Site Simulation File (SSF)
- Global Results File (GRF)
- Risk Summary Output File (RSOF)
- Protective Summary Output File (PSOF)

- Permanent Storage
- Chemical properties database
- MET database

The SSF, GRF, RSOF, and PSOF data files are produced by the system during a simulation. These data files are the standard data files of the system and are produced by the system processors. The SSF contains all the site data required by all of the modules, except data supplied by the chemical properties and MET databases, and information describing the assessment scenario to be implemented. Source types, transport and exposure media, and receptor characteristic data are included. The GRF contains all the key output data from the MMSP modules that were executed during the simulation. The RSOF is the output file of the Exit Level Processor I (ELP I) and contains key exposure and risk results from the simulations. The PSOF is produced by the Exit Level Processor II (ELP II) and contains summary exposure and risk information. The Permanent Storage allows the user to specify where the important files associated with the MMSP modules are to be stored. The level of importance attributed to the output files for storage purposes is defined by the SELECTIONS subscreen of the SYSTEM MANAGEMENT SCREEN that is described in the next section. The chemical properties database contains the data for all chemical variables. The MET database contains five different meteorological datasets for the site: hourly, daily, monthly, annual, and long-term climatological data.

### 2.1.1.3 Processors

The user must specify six system processors. These six system processors are the Distribution Statistics Processor (DSP), Site Definition Processor (SDP), Computational Optimization Processor

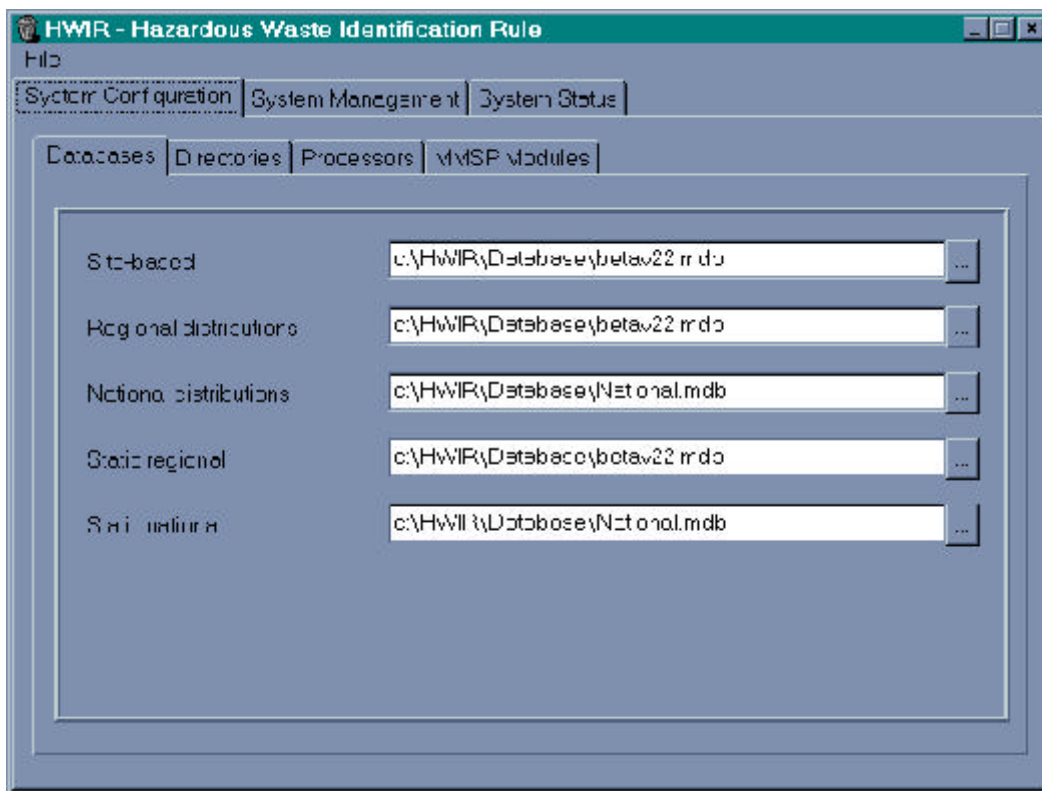


Figure 2.1 Database Subscreen of System Configuration Screen

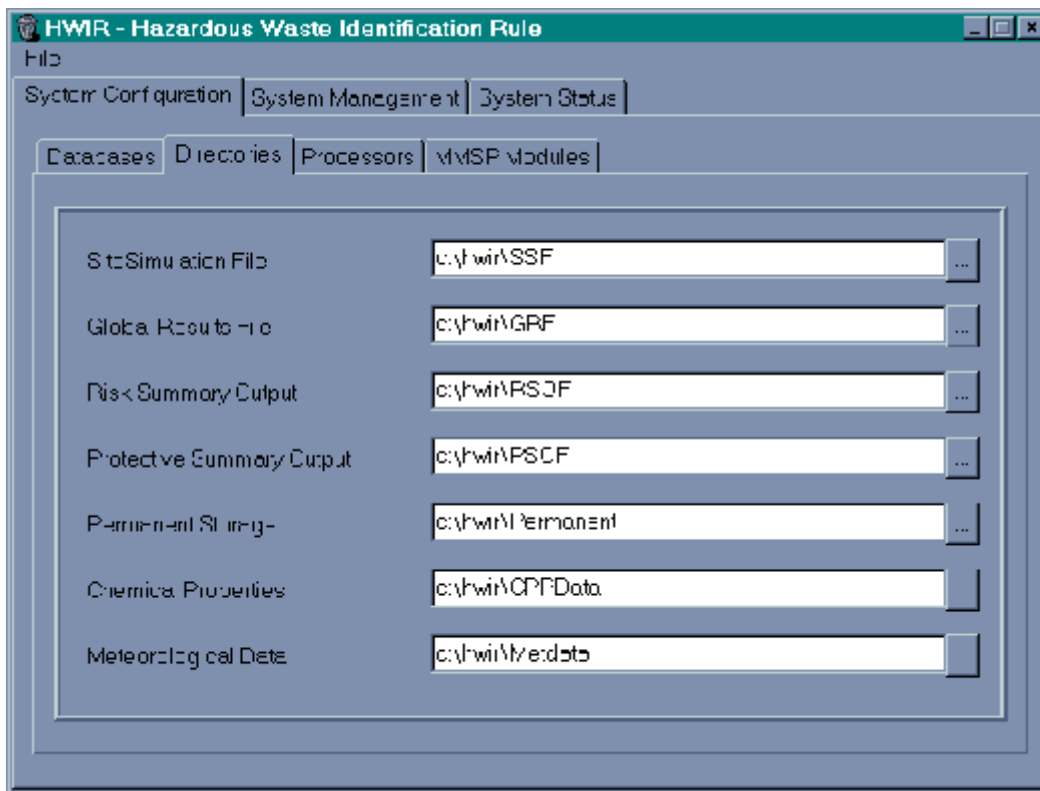


(COP), MMSP, ELP I, and ELP II. The DSP, COP, and ELP-II have check boxes, indicating whether they will be used for the simulation. If an x does not appear in the box, the system will not run that processor. The user can select the  button to choose the correct executables or type the appropriate drive, directory, and file name (for example, C:\HWIR\PROCESSOR\SDP.EXE). Each processor must be identified and opened before a simulation can be started. Figure 2.3 shows the PROCESSORS subscreen of the SYSTEM CONFIGURATION SCREEN.

#### 2.1.1.4 MMSP Modules

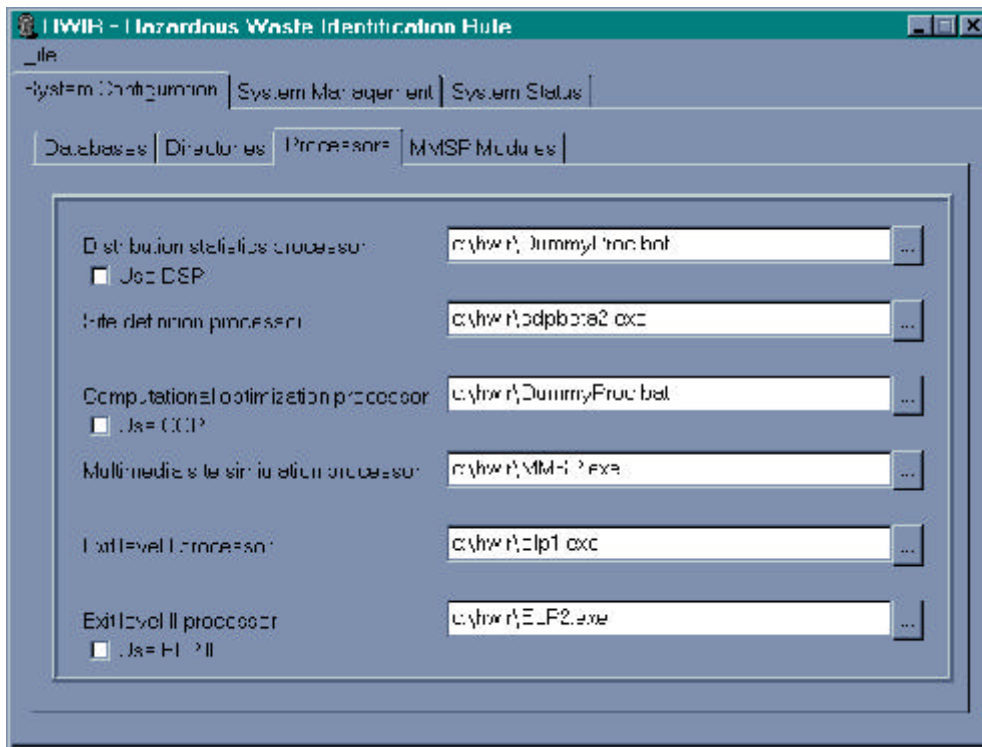
The user must specify four types of modules to run simulations: the source, transport, foodchain, and exposure/risk modules. The user clicks on the MMSP MODULES subscreen to specify these module types. The following paragraphs describe how the user specifies these modules.

Source Modules. After the user has clicked on the Source tab of the MMSP SOURCE MODULES subscreen, the Source tab can be selected. This tab allows the user to specify and open the executables for the five different source modules. These source modules are land application unit, waste pile, surface impoundment, aerated tank, and landfill. The user can click on the  button to select the correct executables or type the appropriate drive, directory, and file name (for example, C:\HWIR\LAU.BAT). Each processor must be identified and opened before a simulation can be started. Figure 2.4 shows the Source tab of the MMSP SOURCE MODULES subscreen.

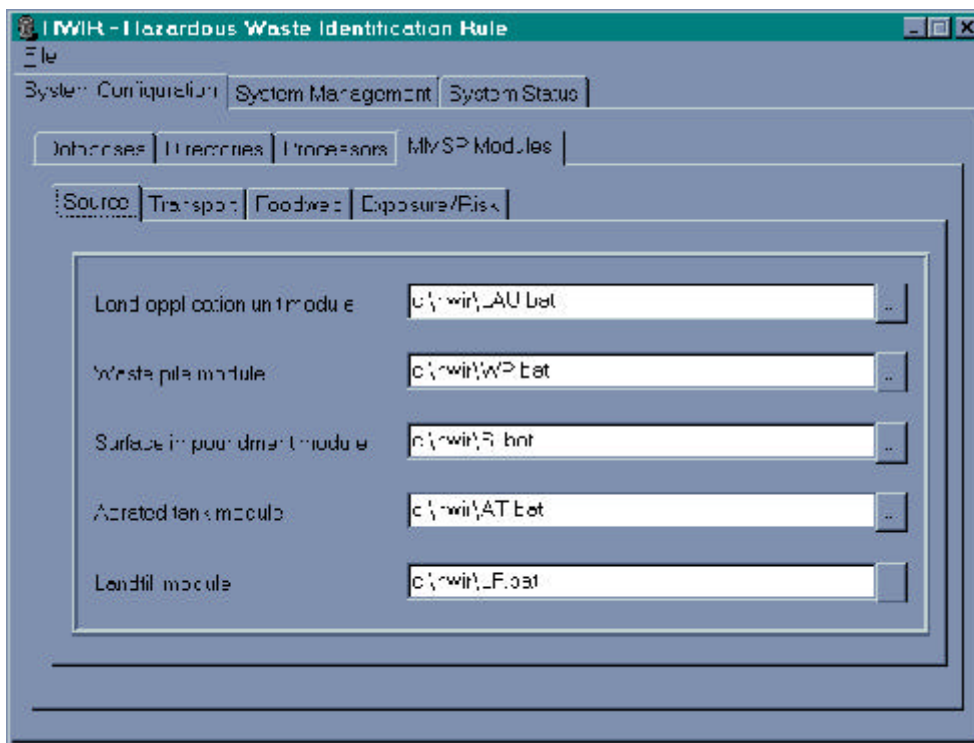


**Figure 2.2** Directories Subscreen of System Configuration Screen








**Figure 2.3** Processors Subscreen of System Configuration Screen



**Figure 2.4** MMSp Source Modules Subscreen of System Configuration Screen

**Transport Modules.** The different transport modules are user specified by clicking on the Transport tab of the MMSP SOURCE MODULES subscreen. This tab allows the user to specify and open the executables for the five different transport modules: vadose zone, aquifer, watershed, waterbody, and air. The user can select the  button to choose the correct executables or type the appropriate drive, directory, and file name (for example, C:\HWIR\AIRMODEL.EXE). Each processor must be identified and opened before a simulation can be started. Figure 2.5 shows the Transport tab of the MMSP SOURCE MODULES subscreen.

**Foodchain Modules.** The user can specify the different foodchain modules by clicking on the Foodchain tab of the MMSP SOURCE MODULES subscreen. This tab allows the user to specify and open the executables for the three different foodchain modules: aquatic, terrestrial, and farm. The user can click the  button to select the correct executables or type the appropriate drive, directory, and file name (for example, C:\HWIR\FARMFOOD.EXE). Each processor must be identified and opened before a simulation can be started. Figure 2.6 shows the Foodchain tab of the MMSP SOURCE MODULES subscreen.

**Exposure/Risk Modules.** The user can specify the different foodchain modules by clicking on the Exposure/Risk tab of the MMSP SOURCE MODULES subscreen. This tab allows the user to specify and open the executables for the four different exposure and risk modules: ecological exposure, ecological risk, human exposure, and human risk. The user can select the  button to choose the correct executables or type the appropriate drive, directory and file name (for example, C:\HWIR\EXPOSURE.EXE). Each processor must be identified and opened before a simulation can be started. Figure 2.7 shows the Exposure/Risk tab of the MMSP SOURCE MODULES subscreen.

## 2.1.2 System Management Screen

The functions of the SYSTEM MANAGEMENT SCREEN allow the user to define the sites, chemicals, source types, and waste levels for simulation. It also allows the user to select the level of storage of data files from the MMSP modules, the mode of operation, the number of realizations, and seed value for the statistical analysis of the simulation. Each of these functions is accessed by the user through subscreens associated with the SYSTEM MANAGEMENT SCREEN. These functions are described in the following sections.

### 2.1.2.1 Selections

The SELECTIONS subscreen (Figure 2.8) of the SYSTEM MANAGEMENT SCREEN allows the user to select the sites, chemicals, source types, and waste levels to be used for the simulation. The user can select a single, a subset, or all of the sites, chemicals, source types, and waste levels to be simulated. The sites appear on the subscreen as site identification names that come directly from the chemical properties database. The chemicals appear on the subscreen as the actual names defined in the site-based database. The five source types are land application unit, waste pile, surface impoundment, aerated tank, or landfill. The waste level categories define the waste concentration at the source on which the protection level is based.

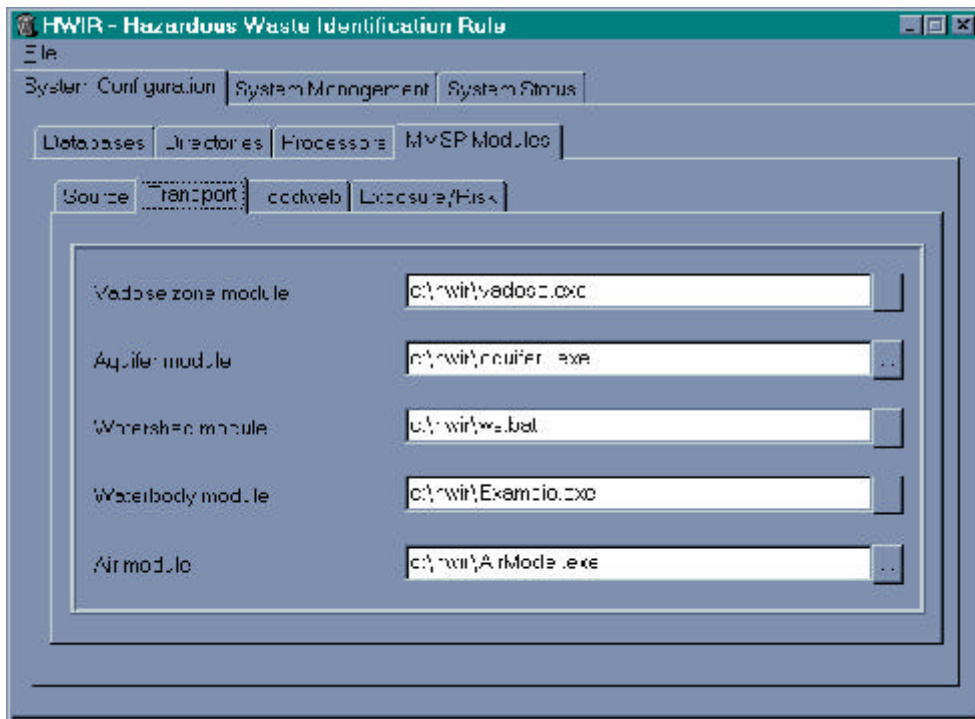


Figure 2.5 MMSP Transport Modules Subscreen of System Configuration Screen

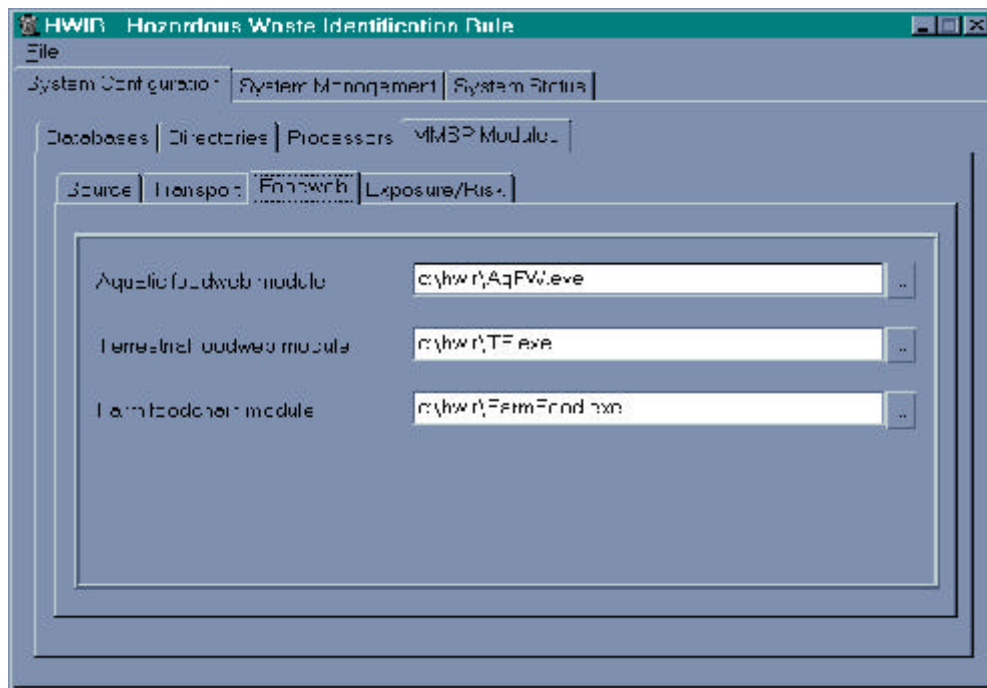


Figure 2.6 MMSP Foodchain Modules Subscreen of System Configuration Screen

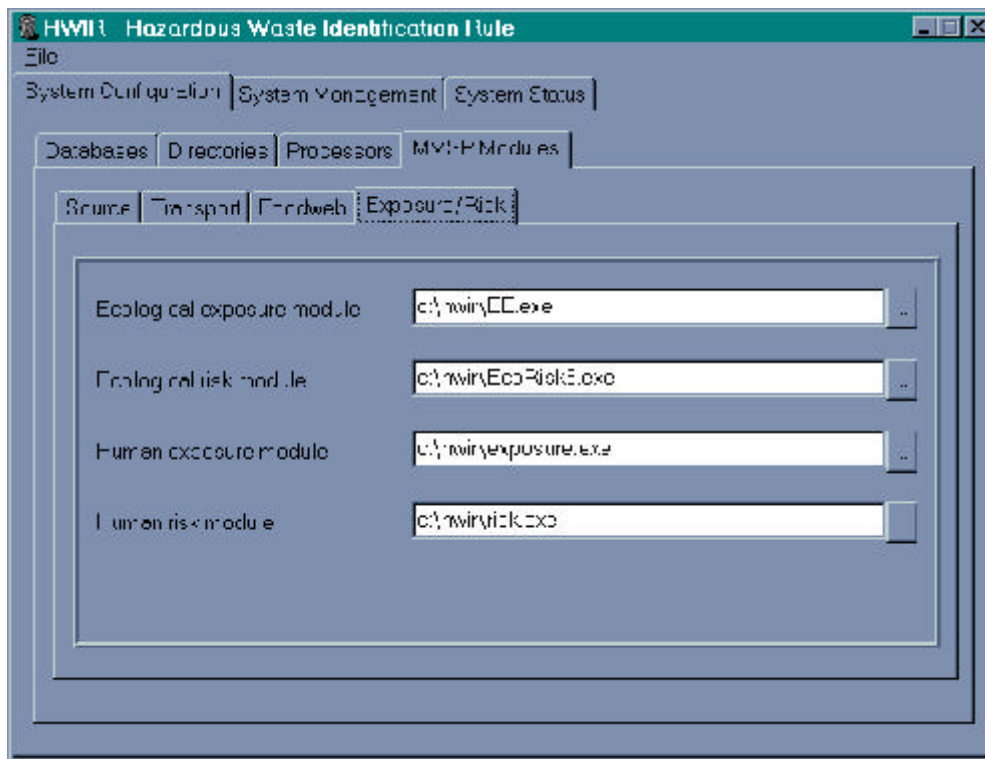


Figure 2.7 MMSP Exposure/Risk Modules Subscreen of System Configuration Screen

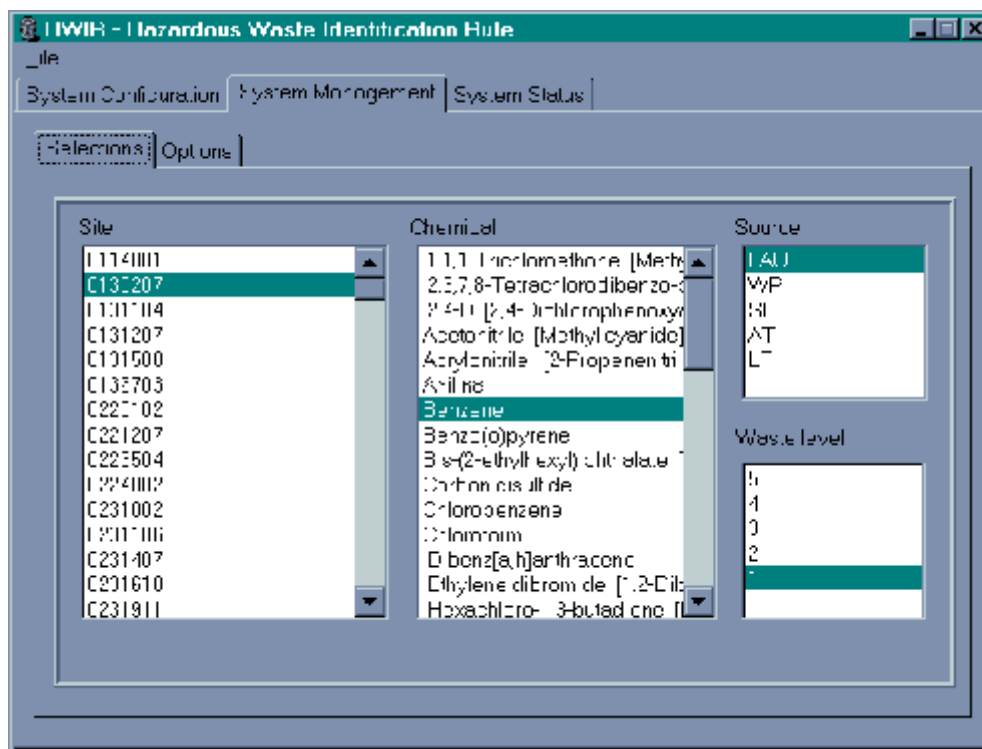


Figure 2.8 Selections Subscreen of the System Management Screen

### 2.1.2.2 Options

After the site, chemicals, source types, and waste levels are selected, the OPTIONS subscreen (Figure 2.9) of the SYSTEM MANAGEMENT SCREEN is used to select the data storage level, mode of operation, number of realizations, and seed value used as part of the probabilistic simulation. The storage level selections are currently Minimum or Maximum. The Minimum storage level stores only the GRF from the simulation; the *Maximum* storage level stores the GRF and all intermediate files created during the simulation. The number of realizations for the simulation is defined by the user. The value has statistical significance, and the user must understand the needs of the simulation to define this value appropriately.

The seed value is used by the statistical component of the simulation. This value defines the random sampling seed required for the simulation. If the user needs to duplicate a simulation, this value must be the same as in the original simulation.

The user can select three different operation modes for the simulation. The **Debug** mode stops the simulation after every change in realization, chemical, source type, and waste level. This mode is intended to test the system setup and operation before a production run. The **Stop on Error** mode stops the simulation only if an error occurs in a processor or module when production runs are being implemented. The **Stop on Warning** mode will stop the simulation only if a warning is present in a

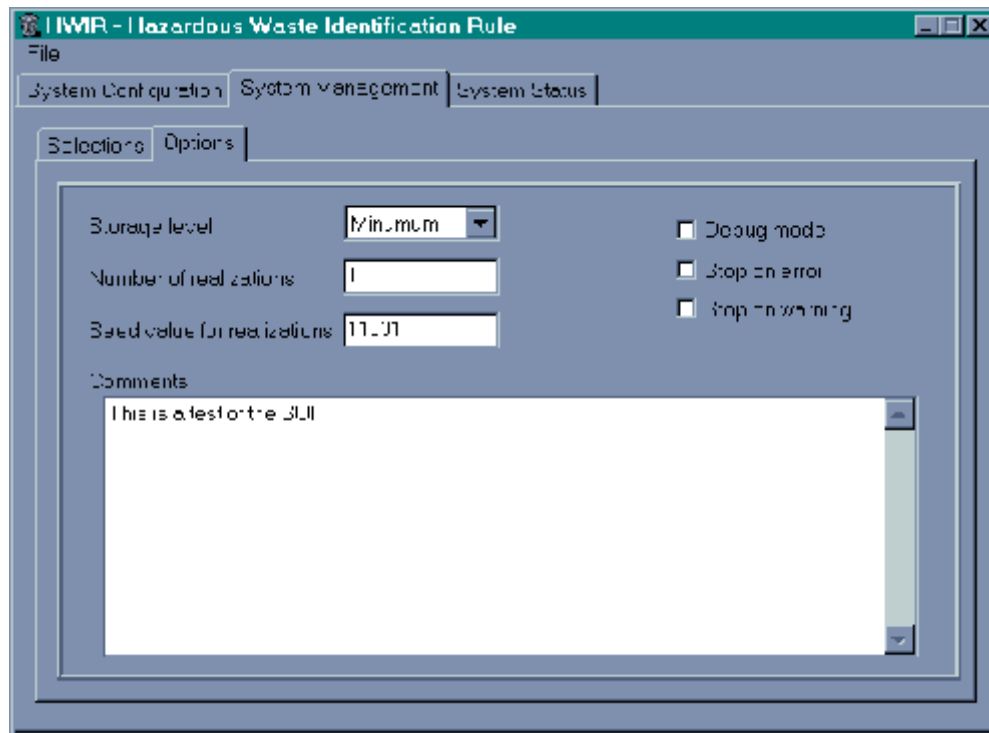


Figure 2.9 Options Subscreen of the System Management Screen

processor or module. The user can simultaneously select the **Debug**, **Stop on Error**, and **Stop on Warning** modes. The Comments box on this subscreen provides the user with the capability to document the run. This box is filled out by the user with all comments appropriate for the simulation.

### 2.1.3 System Status Screen

The main function of the SYSTEM STATUS SCREEN (Figure 2.10) is to allow the user to follow the progress of the simulation, as it iterates through the realizations, sites, chemicals, source types, and waste levels. Depending on the mode of operation the user selected for the simulation (see OPTIONS subscreen of SYSTEM MANAGEMENT SCREEN, Section 2.1.2.2), this screen behaves differently. If the **Debug** mode is selected, the simulation stops after each realization whether it be a site, chemical, source type, or waste level. The user can then select the **Resume** button to continue the simulation or evaluate the results of the simulation (task switch out of the FRAMES-HWIR Technology Software System to some other software). If the **Stop on Error** or **Stop on Warning** mode is selected, the software continuously steps through the simulation until it encounters an error or warning message from a processor or module. If an error or warning occurs, the simulation is suspended, and the user can evaluate the situation and decide whether to continue the simulation or stop it. If the simulation is not completed, all files created before the suspension of the simulation are stored in the specified locations and are available for future evaluation. If the simulation is suspended, the user can attempt to correct the problem, and then resume the simulation.

Any error or warning messages generated by a processor or module are displayed in the Messages box provided at the bottom of the SYSTEM STATUS SCREEN. These messages help the user

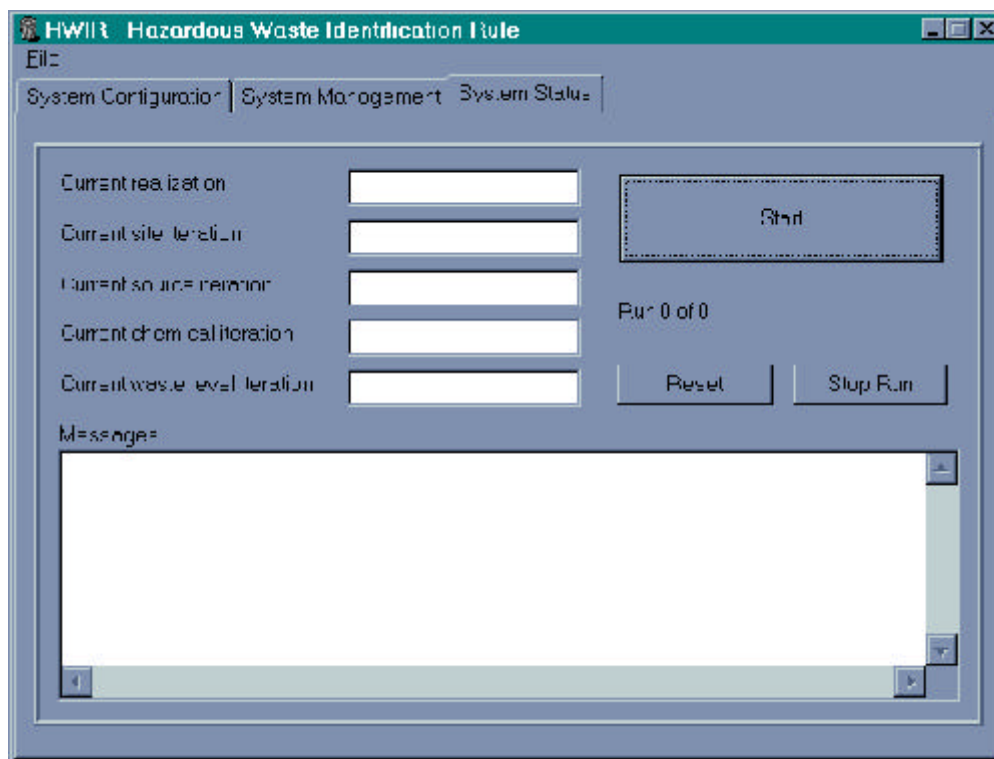


Figure 2.10 System Status Screen

determine the status of the simulation and whether to continue or suspend the simulation. Error and warning files that correspond to these messages are created by the system. The user can use these files to determine the problem and continue the simulation, or stop it and correct the problem. If the user stops the simulation and corrects the problem, the system must be reset, selections corrected from where the SUI stopped, and the simulation started over.

## **2.2 Scientific Requirements for the System User Interface**

No mathematical formulations, assumptions, or limitations affect requirements for the SUI. The SUI is required to test as a stand-alone piece from the other processors within the FRAMES-HWIR Technology Software System.

## **2.3 Output Requirements for the System User Interface**

Most output from the FRAMES-HWIR Technology Software System comes from the processors and modules under the management of the SUI. The only output requirement for the SUI is to provide cumulative system, processor, and module warning and error files associated with the series of simulations. These warning and error files are used to determine the severity of the problem, correct it, and resume the series of simulations via the SYSTEM STATUS SCREEN (see Figure 2.10). The warning and error files are written to the GRF directory. The SUI does help the user specify the location of the system output files, such as the SSF, GRF, RSOF, and PSOF through the DIRECTORIES subscreen of the SYSTEM CONFIGURATION SCREEN.



## 3.0 Design Elements

The SUI is designed to meet the requirements identified in Section 2.0. This section describes the input, implementation, and output design of the SUI.

### 3.1 Input

The SUI is a Windows® 95 menu-driven interface that collects information from the user and controls the activities associated with the simulation. The three main screens associated with the SUI are 1) SYSTEM CONFIGURATION, 2) SYSTEM MANAGEMENT, and 3) SYSTEM STATUS. These user interface screens are the main method of interaction between the user and the FRAMES-HWIR Technology Software System.

The SYSTEM CONFIGURATION SCREEN (see Figures 2.1 to 2.7) allows the user to specify all the databases, data files, and executables for a simulation. All system processor and module executables are specified by the user through the SYSTEM CONFIGURATION SCREEN. All system and module output files are specified by the user. An example of the type of information specified by the user is computer drive (for example, C:), directory (for example, \HWIWR\AIR), and, when appropriate, executable (for example, C:\HWIR\AIR\ISC-ST.EXE). This information is stored and transferred to the appropriate processor or module. If the user cannot locate a database, data file, or executable, it is the responsibility of the user to load the missing files onto the computer from outside the SUI.

The SYSTEM MANAGEMENT SCREEN (see Figures 2.8 and 2.9) consists of SELECTIONS and OPTIONS subscreens. The SELECTIONS subscreen allows the user to select the sites, chemicals, source types, and waste level categories for evaluation. One, some, or all sites, chemicals, source types, and waste level categories are selected by the user for the simulation. The OPTIONS subscreen allows the user to determine the storage level of input/output files for the simulation. The Minimum storage level saves only the GRF for the simulation, and the Maximum storage level saves all processor and module files for the simulation. The OPTIONS subscreen also allows the user to specify the number of realizations for the simulation and the seed value used for the statistical analysis. This subscreen allows the user to specify the mode of operation for the simulation. These modes are: a) **Debug**, b) **Stop on Error** c) **Stop on Warning**, or d) any combination of these three modes. The **Debug** mode stops the simulation after every change in realization, chemical, source type, and waste level; the **Stop on Error** mode stops the simulation only if an error occurs in a processor or module; and the **Stop on Warning** mode stops the simulation only if a warning occurs in a processor or module. The user can select the **Debug**, **Stop on Error**, and **Stop on Warning** modes in different combinations to meet the needs of the simulation. The Comments box on this subscreen enables the user to enter descriptive information about the current simulation.

The SYSTEM STATUS SCREEN (see Figure 2.10) informs the user of the status of the current simulation. Current realization, site, source type, chemical, and waste level are provided by this screen. Any error or warning message will display in the Messages box on the screen and include which processor or module the message came from. This screen also has the **Start/Resume** button for the simulation. After stopping, the user can resume the simulation at anytime or use the **Reset** button to



begin the simulation over (that is, put all the indexes back to the original settings and start the simulation from the beginning). This provides the user with full control over the simulation.

### 3.2 Implementation

To implement the system, the user activates the FRAMES-HWIR Technology Software System using the HWIR icon. The SUI allows the user to start, resume, and reset commands, and provides a running commentary as to the status of the system. The user has the option to implement particular management schemes through the SUI, which sets constraints on the levels of analysis, provides instructions for processors, and locates files for analysis. To fulfill the tasks, the SUI takes control over each of the processors and modules before the start of analysis. The levels of analysis (looping) and processor implementation are outlined as follows:

For selected Sites

For selected Source Types

For selected Realizations

For selected Chemicals

For selected Waste Levels

Execute DSP

Execute the SDP

Execute the COP

Execute the MMSP, passing file locations to the modules

Execute ELP I

Check for User Termination

End loop on selected Waste Levels

End loop on selected Chemicals

End Loop on Realizations

End Loop on selected Source Types

End Loop on selected Sites

Execute ELP II

The HWIR Assessment Strategy implemented only the Sites, Source Types, Chemicals, and Waste Level Loops.

If the SUI receives a warning from a processor or module, the SUI displays the warning and continues (unless the **Stop on Warning** mode is selected, in which case the simulation stops) with the simulation in the SYSTEM STATUS SCREEN (see Figure 2.10). If an error is reported to the SUI, the SUI displays the error message and current state of the indexes and stops execution.

### 3.3 Output

The SUI creates the header file in the Site Simulation Directory. The header file is discussed in *Documentation of the FRAMES-HWIR Technology Software System, Volume 8: Specifications* (see Section 6.0). The SUI displays information and status of the simulation within the screens of the interface. See Section 2.0 of this document to obtain information on the SUI screens and the Messages/Comments boxes.

The SUI also produces cumulative warning and errors files (warning.ALL and error.ALL) within the GRF directory. These files can be used to verify the errors for a particular run or series of runs. This file must be maintained manually by the user, as needed (for example, for testing purposes). When the

**Start** button is pressed, the warning and error files are deleted and new ones are created for the current simulation.

## 4.0 Testing Approach and Results

This section describes the testing conducted for the SUI. As mentioned, the SUI manages the overall execution of the FRAMES-HWIR Technology Software System by interacting with and directing the processors that compose the system. In addition, the SUI serves as the interface between the user and the FRAMES-HWIR Technology Software System. This section summarizes the requirements for the SUI in a form suitable for testing, shows how the test cases that follow relate to these requirements, describes each test case, and describes the baseline results for each test case. The test cases address the basic functionality listed in the requirements (black box testing).

### 4.1 Summary of Requirements

Requirements for the SUI are described in Section 2.0, and the design is described in Section 3.0. The requirements specified (or implied) in these sections were reworded into the list (Table 4.1) of concise, fundamental requirements suitable for testing.

**Table 4.1** Testing Requirements for the System User Interface

Requirement Number	Requirement
1	Control the execution of the system processors.
2	Allow the user full control over which executables compose the system, which databases are accessed, and where input/output files are stored.
3	Allow the user to decide which sites, source types, chemicals, and waste levels are considered, as well as the number of realizations to perform.
4	Allow the user to decide the storage level (that is, a level describing how much intermediate data are saved).
5	Allow the user to document the simulation description data (user-defined header information, including such items as name of simulation, objective, and unique aspects).
6	Report all errors/warnings that occur in the processors.
7	Allow the user to decide if the system should interrupt a run when an error occurs or automatically continue execution with the next waste level.
8	Allow the user to start a run, interrupt a run after each realization, resume an interrupted run, or reset a run after stoppage.
9	Report the current status of the system, that is, the current values for all loops (site, source, chemical, waste level, and realization) and the errors/warnings produced so far.
10	When an error is produced by a processor and the user has indicated the run should be interrupted when an error occurs (#7), interrupt the run and have the following options available for proceeding: 1) continue the run at the next waste level (resume) or 2) abort the run (reset).
11	Identify the last concentration level to be evaluated (this aids in ELP execution).
12	Have a <b>Testing/Debug</b> mode allowing incremental execution of processors.

Requirement Number	Requirement
13	While a run is executing, do not allow the user to respecify the executables composing the sytem, databases accessed, location of input/output file storage, site selection list, source selection list, chemical selection list, waste level list, number of realizations, or storage level. If any of these parameters are changed, the entire run must restart from the beginning.
14	Function within the Windows® 95 operating system, be user-friendly, and have the same functionality as standard Windows® 95 software.

Table 4.2 shows the relationship between these requirements and the test cases described in Section 4.2. Although some requirements apply to all test cases, this table shows which requirements are specifically evaluated by each case.

**Table 4.2** Relationship Between Test Cases and Requirements for the System User Interface

		Test Cases			
		01	02	03	04
Requirement	1	x	x	x	x
	2	x	x	x	x
	3	x	x	x	x
	4	x	x		
	5	x			
	6			x	
	7			x	
	8				x
	9	x	x	x	x
	10			x	
	11				x
	12				x
	13	x	x	x	x
	14	x	x	x	x

## 4.2 Test Cases

The following test cases can be run under the Windows® Explorer. Note that during testing, no other activities can be conducted on the computer because of the way the processors and modules open, execute, and close. Note also that during unit testing, only one site, six chemicals, and three source types were available for simulation.

All test cases evaluate the ability of the SUI to do the following tasks:

- C control the execution of the system processors—the SUI allowed processors to be specified and executed, as well as stopped and restarted
- C allow the user full control over executables, databases, and input/output files—the SUI allowed user input in all those categories and prompted the user if input was not provided
- C allow the user to decide what to simulate and the number of realizations to perform—the SUI allowed user input in all those categories
- C report the current status of the system—the SUI kept a running tab in the Messages box describing the status of the system. In addition, the realization, site identification, source identification, and chemical identification changed as each run was initiated
- C keep the user from changing critical simulation parameters in the middle of a run—once key simulation parameters were chosen and the **Start** button selected, the SUI hid screen tabs, except that of the System Status, thus preventing the user from changing parameters in mid-simulation
- C function under Windows® 95, be user-friendly, and have the same functionality as standard Windows® 95 software—the SUI allowed user input, was relatively intuitive, functioned under Windows® 95, and had similar functionality.

Therefore, according to unit testing, these requirements were satisfied by the SUI. The following test cases describe the evaluation of the SUI to meet other specific requirements, as identified in Table 4.2.

### 4.2.1 SUI\_01

#### 4.2.1.1 Description and Rationale

This test case evaluates the SUI's ability to allow the user to set the storage level and to document simulation description data.

#### 4.2.1.2 Input Data/Procedure

The tester double-clicks on the *SUI.exe* icon, which brings up the SUI screens. The tester chooses the **File** drop down menu, then **Open**, selecting file *hdsuitc1.ssf* from the directory. All fields should populate automatically in all screens, including the Comments box. Under the **Options** tab,

the tester ensures the storage level is set to minimum, the realizations to 100, and the seed value to 45.

The tester also sets the SUI to **Stop on Error** and **Stop on Warning**. The tester then clicks the **Start** button.

When the run is completed, the tester uses the Windows® Explorer to access the GRF subdirectory, opening the file *hdsuitc1.grf*. Scrolling down to the command **Storage Level**, the tester checks to ensure the value is zero for minimum storage.

#### 4.2.1.3 Expected Results

The SUI is expected to load the previously input data from the file, including the information in the Comments box. The SUI is also expected to report the status of the system in an accurate and timely fashion and to perform without error all 100 realizations. When the header file is checked after all realizations have been run, the file is expected to contain a command for a minimum storage level.

#### 4.2.1.4 Baseline Testing Results

The SUI loaded the file as expected, including the information in the Comments box. During execution, the SUI displayed in the Messages box the current realization, realization, source, chemical, and waste level. In the file *hdsuitc1.grf*, the storage level was set at zero. Therefore, the SUI passed this test case.

### 4.2.2 SUI\_02

#### 4.2.2.1 Description and Rationale

This case adds additional complexity to Test Case SUI\_01 by further testing the requirement that the SUI must allow for any number of sites, source types, chemicals, waste levels, and realizations, as well as store the maximum data.

#### 4.2.2.2 Input Data/Procedure

The input data consist of the locations and file names of the executables and databases composing the system, locations for file storage, run control information, and description documentation. The file *hdsuitc2.ssf* provides all this information for this test case.

The tester double-clicks on the *SUI.exe* icon, which brings up the SUI screens. The tester chooses the **File** drop down menu, then **Open**, selecting file *hdsuitc2.ssf* from the directory. All fields should populate automatically in all screens. Under the SYSTEM MANAGEMENT SCREEN, Selection tab, the tester selects the site, the first five chemicals, the first two source types, and two waste levels (1 and 2). Under the Options tab, the tester then selects the maximum storage level, sets the realizations to 25, and sets a seed value of 321. The tester also sets the SUI to Stop on Error and Stop on Warning. The tester then clicks the Start button.

When the run is completed, the tester uses the Windows® Explorer to access the GRF subdirectory, opening the file *hdsuitc2.grf*. Scrolling down to the command **Storage Level**, the tester checks to ensure the value is 1 for maximum storage.

#### 4.2.2.3 Expected Results

The SUI is expected to perform the run without error and properly loop through the specified source types, chemicals, waste levels, and realizations. The looping can be evaluated by monitoring the status information while the run is in progress. When the header file is checked after all realizations have been run, the file is expected to contain a command for a maximum storage level.

#### 4.2.2.4 Baseline Testing Results

The SUI cycled through the parameters as expected. No errors or warnings were produced. The header file contained a 1 for maximum storage. Therefore, the SUI passed this test case.

### 4.2.3 SUI\_03

#### 4.2.3.1 Description and Rationale

This test case evaluates the ability of the SUI to handle warnings and errors. At particular realizations, the dummy MMSP is rigged to produce an error and a warning. The test will be executed three times – once with the **Interrupt on Error** and **Warning** options and use of the **Resume** button, once with the **Interrupt on Error** and **Warning** options and use of the **Reset** button, and once with the **Continue on Error** and **Warning** options.

#### 4.2.3.2 Input Data/Procedure

The input data consist of the locations and file names of the executables and databases composing the system, locations for file storage, run control information, and description documentation. The file *hdsuitc3.ssf* provides all this information for this test case.

The tester double-clicks on the *SUI.exe* icon, which brings up the SUI screens. The tester chooses the **File** drop down menu, then **Open**, selecting file *hdsuitc3.ssf* from the directory. All fields should populate automatically in all screens. For the first portion of the test, the tester sets the SUI to **Stop on Error** and **Stop on Warning**. The tester then returns to the SYSTEM STATUS SCREEN and clicks the **Start** button. Note that during this test case, the MS-DOS window that opens for each processor does not close automatically for the surrogate MMSP file (file *mmsp.bat*). To ensure the test proceeds, the tester must manually close this window each time the MMSP is run. After each warning or error is encountered, the tester presses the **Resume** button until the run is complete.

After the first portion of the test, the tester checks the *message.all* file in the GRF subdirectory to ensure the messages match the screen viewed during execution. The tester then deletes this file.

For the second portion of the test, the tester sets the SUI to **stop on warning** and **stop on error** and presses the **Start** button. After the first warning is encountered, the tester presses the **Reset** button. The tester then follows the same protocol as the first portion of the test, until the error is encountered. The tester then presses the **Reset** button and allows the run to complete. After this portion of the test, the tester checks the *messages.all* file in the GRF subdirectory to ensure the messages match the screen viewed during execution. The tester then deletes that file again.

For the third portion of the test, the tester does not select any of the stopping options (**Testing/Debug**, **Stop on Warning**, or **Stop on Error**). After this portion of the test, the tester checks the *message.all* file in the GRF subdirectory to ensure the messages match the screen viewed during execution. The tester then deletes that file again.

#### 4.2.3.3 Expected Results

The errors and warnings are expected to be reported and recorded as part of the run status information. When no stoppages are specified, the run should not be interrupted (except during the operation of the MMSP, as noted previously), but should instead continue with the next loop realization. When the **Stop on Error** or **Stop on Warning** option is specified, the run should be interrupted when the warning or error occurs. Ordinarily, a warning allows the run to complete before stopping. An error should stop the run in mid-processing. The **Resume** option for proceeding should result in the run continuing and the **Reset** option should cause run execution to terminate.

#### 4.2.3.4 Baseline Testing Results

For the first portion of this test case (testing of the interrupt on warning and error, and resume functions), the SUI ran through the first realization as expected. On the first run of the second realization, when the MMSP was first encountered, the SUI issued the expected warning. When **Resume** was chosen, the SUI issued the same warning for each successive run in realization 2, stopping each time until **Resume** was chosen again. The SUI reacted the same way to the warning in realization 3. On the first run of the fifth realization, when the MMSP was first encountered, the SUI issued the expected error message and stopped the run. It also did not augment the successful run count for any of the runs in which errors were encountered. It reacted the same way each time **Resume** was chosen, then and in successive runs through the fifth realization. The file *messages.all* contained the same information as was displayed on the SUI status screen. Therefore, the SUI passed this portion of the test case.

For the second portion of this test case (testing the reset function), the SUI ran through the first realization as expected. On the first run of the second realization, when the MMSP was first



encountered, the SUI issued the expected warning. When **Reset** was chosen, the SUI started over again with the first run of the first realization, as expected. **Resume** was chosen to allow the SUI to run until the error message appeared (first run of realization 5). At that point, **Reset** was chosen again, and the SUI started over with the first run of the first realization. The file *messages.all* contained the same information as was displayed on the SUI status screen. Therefore, the SUI passed this portion of the test case.

For the third portion of this test case (for testing of continue on warning and error), the SUI ran through all runs of all realizations. The expected warnings appeared in the Messages box during all runs of realizations 2 and 3. The expected error messages appeared in the Messages box during all runs of realization 5. The file *messages.all* contained the same information as had been displayed on the SUI status screen. Therefore, the SUI passed this portion of the test case.

#### 4.2.4 SUI\_04

##### 4.2.4.1 Description and Rationale

This cases evaluates the **Testing/Debug** mode required for the SUI, including various features such as stepping and break points. It also evaluates the SUI's ability to identify the last concentration level ( $C_w$ ) to be executed for that run.

##### 4.2.4.2 Input Data/Procedure

The input data consist of the locations and file names of the executables and databases composing the system, locations for file storage, run control information, and description documentation. The file *hdsuitc4.ssf* provides all this information for this test case.

The tester double-clicks on the *SUI.exe* icon, which brings up the SUI menu boxes. The tester chooses the **File** drop-down menu, then **Open**, selecting the file *hdsuitc4.ssf* from the SUI\_TB directory. All fields populate automatically in all menu boxes. The tester then clicks on the **Start** button.

The SUI stops after each run in a realization. When the SUI stops, the tester returns to the Windows® Explorer and opens the file *hdsuitc4.ssf* in the main test bed directory. The tester then checks the parameter, Last  $C_w$ . This parameter should be an F for  $C_w$ s 1 and 2. It should be a T for  $C_w$  3.

##### 4.2.4.3 Expected Results

The features of the **Testing/Debug** mode are expected to function properly without warnings or errors, allowing the user to execute a single realization at a time. A T will only be shown for the parameter, Last  $C_w$ , when a run involving  $C_w$  3 has been executed.

#### 4.2.4.4 Baseline Testing Results

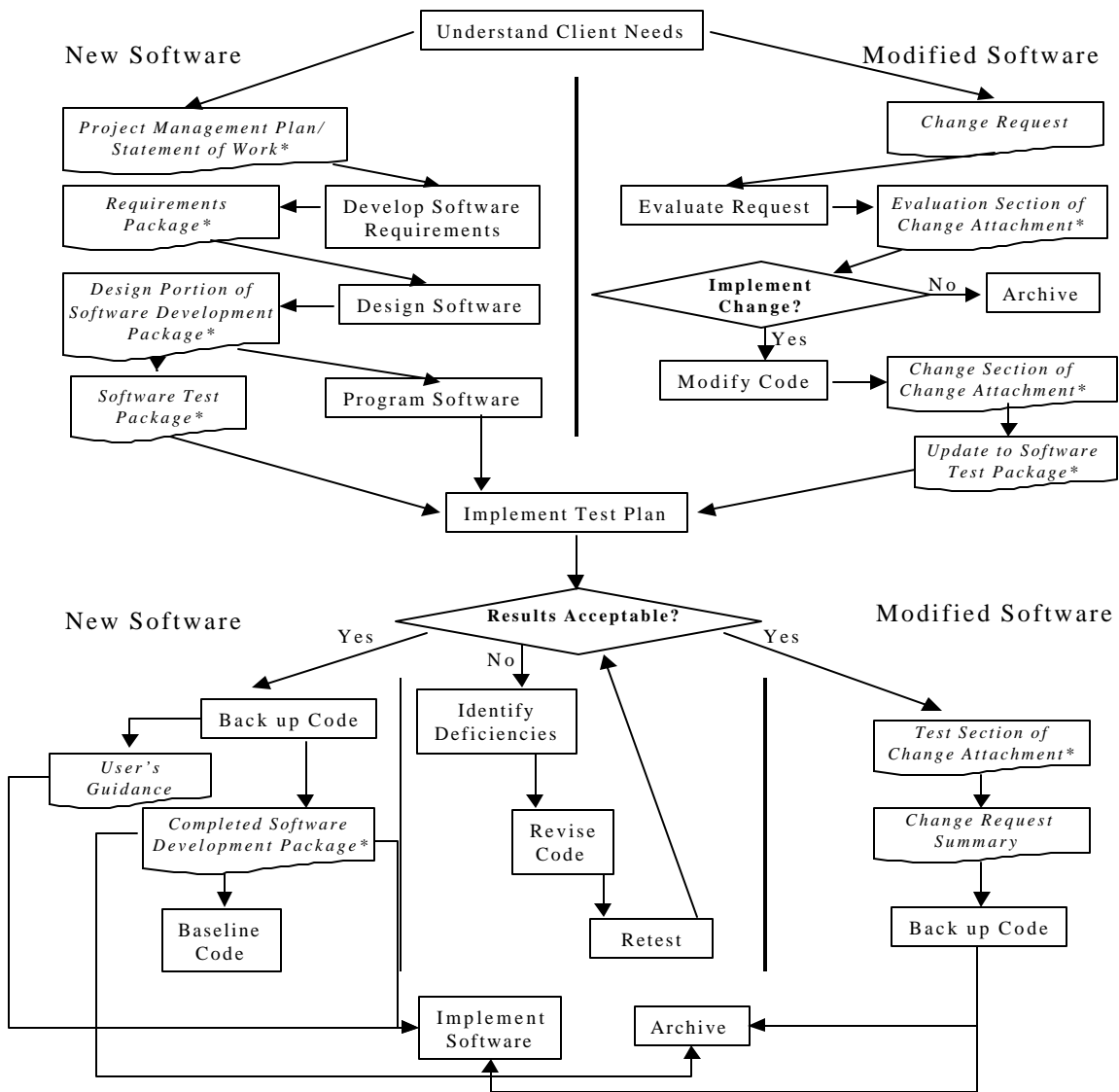
The SUI stopped after each run within the two realizations, as expected. At each stop, it was possible to change sites, realizations, chemicals, sources, and processors. The parameter, Last  $C_w$ , had a value of T only at the end of a run involving  $C_w$  3. No warnings or errors were produced. Therefore, the SUI passed this test case.

## 5.0 Quality Assurance Program

The SUI was developed under a quality assurance program documented by Gelston et al. (1998). Under this program, quality is defined as the ability 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 5.1 outlines the software development process used for the SUI, highlighting the quality check points. (Note the SUI activities flow down the left side of the figure, 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 the 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 5.1.

Development of the SUI included the implementation of a quality assurance checklist (see Figure 5.2). Understanding of this checklist by all team members resulted in the shared understanding of component requirements and design necessary to ensure quality. Completion of this checklist verified that all documentation was completed for transfer of the software to the client.



**Figure 5.1** Ensuring Quality in the Environmental Software Development Process (\* indicates quality review stage; box with wavy bottom line and italics font indicates a document rather than an activity)

**Table 5.1** Relationship of PNNL Environmental Software Development Process to Quality Assurance Requirements (OCRWM 1995, EPA 1997)

<b>OCRWM<sup>(a)</sup> Quality Assurance Requirements<sup>(b)</sup></b>	<b>EPA Essential Element of Information<sup>(c)</sup></b>	<b>Environmental Software Process Equivalent (Section)</b>
	4C System Implementation Plan	Project Management Plan or Statement of Work
I.2.5A Functional Requirements Information Documentation; I.2.5C Requirements and Design Documentation	5C System Detailed Requirements Document	Requirements Package
I.2.1 Software Life Cycles, Baselines (see Appendix C in OCRWM 1995) and Controls	6C Software Management Plan	Project Management Plan or Statement of Work and Gelston et al. (1998)
I.2.2 Software Verification <sup>(d)</sup> and Software Validation; I.2.4 Software Validation <sup>(e)</sup>	7C Software Test and Acceptance Plan	Software Test Package
I.2.3 Software Verification; I.2.5C Requirements and Design Information Documentation	8C Software Design Document	Design Portion of Software Development Package
I.2.6A Configuration Identification		Completed Software Development Package
I.2.6B Configuration Control; I.2.6C Configuration Status; I.2.7 Defect Reporting and Resolution <sup>(f)</sup>	9C Software Maintenance Document	Modification Documentation
	10C Software Operations Document	User's Guidance and Training
I.2.5B User Information Documentation	11C Software User's Reference Guide	User's Guidance and Training
	12C System Integration Test Reports	Software Test Package

- (a) OCRWM = Office of Civilian Radioactive Waste Management.
- (b) Note that OCRWM requirement I.2.8, Control of the Use of Software, is the responsibility of the OCRWM-related client.
- (c) Elements 1 through 3 are generally completed by clients in the U.S. Environmental Protection Agency before contract initiation with the project team.
- (d) Verification includes informal code testing by software engineers (see Appendix C in OCRWM 1995) to ensure that code functions as required.
- (e) Validation includes testing by those other than the software engineers who developed the code in order to provide an independent confirmation that the software functions as required.
- (f) Note that some changes requested by clients may not be made in the software, unless funding has been allocated for such modifications.

---

## Quality Assurance Checklist

### A. General Requirements Analysis

- Documented in
  - \_\_\_ Statement of Work (stored in project file; see Gene Whelan, Gariann Gelston, or current Integration Leader)
- 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 (PNNL-11914, Volume 2, Section 2.0)
- 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
  - \_\_\_ post-October 31 requirements

### C. Design Documentation

- Documented in
  - \_\_\_ design portion of documentation (PNNL-11914, Volume 2, Section 3.0)
  - \_\_\_ team task plans/Project Management Plan (stored in project file; see Gene Whelan, Gariann Gelston, or current Integration Leader)
- 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
  - \_\_\_ post-October 31 design elements

### D. Development Documentation

- Documented in
    - \_\_\_ Specifications Document (PNNL-11914, Volume 8)
    - \_\_\_ Quality Assurance Archive (see Gariann Gelston or current Integration Leader)
  - Contains information on (all of the following)
    - \_\_\_ baseline hard copy of the source code
    - \_\_\_ diskette copy
    - \_\_\_ name of computer language(s) used
- 

**Figure 5.2** Quality Assurance Implementation Checklists for the System User Interface

E. Testing Documentation

- Documented in
  - \_\_\_ test plan that meets quality assurance requirements (PNNL-11914, Volume 2, Section 4.0)
- Contains information on (all of the following)
  - \_\_\_ description of software
  - \_\_\_ testing scope
  - \_\_\_ relationship between test cases and requirements
  - \_\_\_ test activity description
  - \_\_\_ hardware and software needed to implement plan
  - \_\_\_ test case specifications
  - \_\_\_ expected results

F. User's Guidance

- Documented in
  - \_\_\_ hardcopy printout of user's guidance for system (PNNL-11914, Volume 11)
- 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 (PNNL-11880)
  - \_\_\_ Quality Assurance Software-Specific Checklist (PNNL-11914, Volume 2, Section 5.0)
- 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 (see Gariann Gelston or current Integration Leader)
  - \_\_\_ back up disk files in multiple storage locations (see Gariann Gelston or current Integration Leader)
- 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  
System/Module Manager \_\_\_\_\_ Date \_\_\_\_\_

Figure 5.2 (contd)

## 6.0 References

### Documentation for the FRAMES-HWIR Technology Software System

Volume 1: *Overview of the FRAMES-HWIR Technology Software System*. 1998. PNNL-11914, Vol. 1, Pacific Northwest National Laboratory, Richland, Washington.

Volume 2: *System User Interface Documentation*. 1998. PNNL-11914, Vol. 2, Pacific Northwest National Laboratory, Richland, Washington.

Volume 3: *Distribution Statistics Processor Documentation*. 1998. TetraTech, Lafayette, California.

Volume 4: *Site Definition Processor Documentation*. 1998. PNNL-11914, Vol. 4, Pacific Northwest National Laboratory, Richland, Washington.

Volume 5: *Computational Optimization Processor Documentation*. 1998. TetraTech, Lafayette, California.

Volume 6: *Multimedia Multipathway Simulation Processor Documentation*. 1998. PNNL-11914, Vol. 6, Pacific Northwest National Laboratory, Richland, Washington.

Volume 7: *Exit Level Processor Documentation*. 1998. PNNL-11914, Vol. 7, Pacific Northwest National Laboratory, Richland, Washington.

Volume 8: *Specifications*. 1998. PNNL-11914, Vol. 8, Pacific Northwest National Laboratory, Richland, Washington.

Volume 9: *Software Development and Testing Strategies*. 1998. PNNL-11914, Vol. 9, Pacific Northwest National Laboratory, Richland, Washington.

Volume 10: *Facilitating Dynamic Link Libraries*. 1998. PNNL-11914, Vol. 10, Pacific Northwest National Laboratory, Richland, Washington.

Volume 11: *User's Guidance*. 1998. PNNL-11914, Vol. 11, Pacific Northwest National Laboratory, Richland, Washington.

Volume 12: *Dictionary*. 1998. PNNL-11914, Vol. 12, Pacific Northwest National Laboratory, Richland, Washington.

Volume 13: *Chemical Properties Processor Documentation*. 1998. PNNL-11914, Vol. 13, Pacific Northwest National Laboratory, Richland, Washington.

Volume 14: *Site Layout Processor Documentation*. 1998. PNNL-11914, Vol. 14, Pacific Northwest National Laboratory, Richland, Washington.



Volume 15: *Risk Visualization Tool Documentation*. 1998. PNNL-11914, Vol. 15, Pacific Northwest National Laboratory, Richland, Washington.

## Quality Assurance Program Document

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.

## Additional References

Marin, C., and Z. Saleem. 1997. *A Preliminary Framework for Finite-Source Multimedia, Multipathway and Multireceptor Risk Assessment (3MRA)*. Draft, October 1997, U.S. Environmental Protection Agency, Office of Solid Waste, Washington, D.C.

Office of Civilian Radioactive Waste Management (OCRWM). 1995. *Quality Assurance Requirements and Description, 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.