6.0 3MRA Version 1.x – UA/SA Enhancements

The discussion in this section focuses primarily on the software elements of the SuperMUSE design approach along with key enhancements needed for 3MRA model evaluation that are embodied in a pending release of 3MRA Version 1.x. Aspects of future development efforts by ORD regarding UA/SA, and development of flexible modeling frameworks, are also briefly highlighted to bring attention. The discussion highlights the overall research program in model evaluation technology underway at ERD/NERL/ORD, and the benefits that these efforts will likely have on future capabilities to more exhaustively evaluate 3MRA and other EPA models and modeling systems.

Representing a robust solution to the computational dilemma imposed by sampling-based UA/SA for many Windows models, design of **SuperMUSE**, a 225 GHz PC-based, Windows-based **Super**computer for **M**odel **U**ncertainty and **S**ensitivity **E**valuation, was discussed in Section 5. Discussed in the following sections, SuperMUSE management software for this generic PC-based supercomputing approach is equally amenable to supporting model evaluation tasks for Linux-based models. However, this discussion focuses primarily on immediate needs for evaluating the Windows-based model 3MRA. Assuming random assignment in cluster tasking, the integrated SuperMUSE management software and hardware design approach is linearly capable of exploiting cost-benefit relationships in adding CPUs, or increasing CPU speeds or RAM memory, relative to associated impacts on average model runtime.

While many of the software tools discussed here are currently functional, extensively tested at the unit-level and system-level, and well documented and in beta-use, they were not released in 3MRA Version 1.0 in order to more fully address quality assurance elements of the independent re-compilation quality assurance program, additional system-level testing, and final documentation development.

6.1 Overview of SuperMUSE's Supporting Software Toolset

With the proliferation of workstation clusters connected by high-speed networks, providing efficient system support has become an important problem (Cruz and Park, 1999). As a recognized modern programming language for solution of distributed high performance computing problems on heterogeneous platforms (Laure, 2001), Java was selected for supporting software development for SuperMUSE. The Transmission Control Protocol (TCP) and the Internet Protocol (IP) were selected as the underlying basis for network communication.

One popular alternative standard to TCP/IP for programming in clusters is the Message Passing Interface (MPI) (Sunderam *et al.*, 1994). MPI allows the developer to write a program such that when an opportunity exists for the process to be run on another machine, then that will occur. Parallel Virtual Machine (PVM) (Gropp *et al.*, 1999), another approach, allows for similar functionality. For the "embarrassing parallelism" exploited by SuperMUSE, the complexity that could be handled by MPI and PVM is absent, since each client executes simulations independent from other clients. Only during synchronization of the tasks (i.e., which clients take which tasks), and collation of client databases, is client-client network communication active. Even at these times, only brief or summary information is being passed. Future testing is planned to benchmark some of the aspects involved in using TCP/IP, MPI, and PVM techniques.

6.1.1 Supercomputing Supporting Software System Needs

To exploit capabilities of the SuperMUSE parallel computing environment, several software tools were needed. Key functionalities needed that were identified included:

- Facilitating the distribution of workloads among PCs,
- Managing files and data across PCs,
- Facilitating model-specific data analysis tasks, and
- Extension of alternative model input sampling tools.

The above categorization is not drawn along absolute lines in all cases in the discussion that follows, but the separation, as distinguished above, is helpful initially in identifying major functions of each software tool set group listed. For example, the generic "Command Tasker" concept discussed below is fundamentally a tool for managing files and data across SuperMUSE. However, task-specific implementations of this tool concept (e.g., Client Collector) can be viewed to fall into the first category, since the "Command Tasker" actually represents a specific form of a "Model Tasker", which falls more generically into the first category.

Together, the items of key functionalities noted above are grouped as such into 3MRA Version 1.x, though several of the tools are model independent. 3MRA model-specific tools facilitating data analysis are equally amenable to improving capabilities for stand-alone PC execution mode, above and beyond existing capabilities in 3MRA Version 1.0. While some management tools categorized in the second group are actually specific to 3MRA, the third group identifies tools that specifically address post-processing functions for 3MRA. For example, the Site Summary tool, described in Section 6.4.2 can be used to extract selected lists of variables (both model inputs and outputs) contained within site simulation files (SSF) and global result files (GRF) files for each 3MRA simulation. This ability allows the user to collect data in a manageable data structure across many simulations, for purposes of further evaluation of model sensitivities.

Due to resource limitations in meeting immediate 3MRA model evaluation needs, development of alternative sampling tools (e.g., LHS, OAT designs, etc.) that complement the existing random sampling capability in 3MRA Version 1.0, are not currently available in 3MRA Version 1.x, but are to be incorporated in future software releases (i.e., FRAMES 3MRA Version 2.0; see Section 6.8 and Figure 1-1).

Appendix C presents initially developed draft documentation for six system-level processors created to facilitate file management and distributed execution management tools described here.

Where feasible, Java software application programs were developed to support extension of SuperMUSE functionality across multiple operating system platforms for use in evaluating any PC-based model or modeling system.

6.1.2 Tools Facilitating the Distribution of Workloads Among PCs

For the category referencing the facilitation of distribution of workloads among PCs, a group of software products were developed and are classified within the tool set group "Distributed Execution Program Management Tool Set", described in detail in Section 6.2. These represent tools that distribute model simulation tasks across the SuperMUSE PC cluster, and, further, that assist the SuperMUSE system user in monitoring PC clients at system boot-up and during runtime execution. The following four tools fall into this category:

- CPU Allocator (CPU$_{A}$) U
- Model Tasker (MT) U
- Tasker Client (TC) U
- Client Monitor (CM) U

Development of these tools have been completed, indicated by U, and are currently being employed in SuperMUSE simulations for 3MRA. A conceptual layout of the relationship between the CPU Allocator, 3MRA Version 1.0's Model Tasker (i.e., SUITasker), and the Tasker Client tools is depicted in Figure 6-1a. The above tools are used only in parallel execution mode (i.e., these are not amenable nor needed for 3MRA stand-alone execution).

The CPU Allocator is basically the "master brain" of the SuperMUSE operation, managing model applications and PC clients in the cluster. A given Model Tasker basically parallelizes the system user interface of a specific model or modeling system. The Tasker Client is a client-side PC manager that orchestrates tasking received from a Model Tasker. The Client Monitor is a boot-up control manager that ensures proper client boot-up before allowing clients to take-on tasks.

All of the above noted software products are model-independent except for the Model Tasker; a separate Model Tasker is needed for each individual model or modeling system application to be implemented on SuperMUSE. Individual modules (or models) incorporated within a modeling system such as 3MRA Version 1.0 (e.g., the surface water model Exams, the air model ISCST), or within the more generic framework concept embodied by FRAMES 2.0 (Section 6.8), can take advantage of the associated Model Taskers developed specifically for these modeling systems (i.e., SUITasker or 3MRA System User Interface Tasker; and FUITasker or FRAMES User Interface Tasker, respectively; see Section 6.2.2).

The tools noted here are platform independent Java software application programs, except for the Client Monitor Tool; the latter compiled in Visual Basic and based on Win32 API routines.

6.1.3 Tools for Managing Files and Data Across PCs

To meet file management tasking needs in operating SuperMUSE, the following tools were identified in conceptual design stages:

- Update Client (UC) U
- Command Tasker (CT) U
- Process Messages (PM) U
- Client Collector for Aggregated ELP1 (ACC) U
- Client Collector for Disaggregated ELP1 (DCC)

The noted tools have all been developed, except the DCC, and are currently being employed in SuperMUSE calculations for 3MRA Version 1.0. These tools, described in detail in Section 6.3, are used typically only in parallel execution mode. The Process Messages Tool can be executed through batch files in stand-alone execution mode, providing a more formal relational database structure, but is redundant to tasking already performed by the 3MRA's System User Interface (SUI). The SUI currently stores similar messaging in the ASCII flat file "messages.all".

The Update Client (UC) Tool allows system-wide execution of operating system commands (i.e., DOS commands in Windows) serially from a single server. The Command Tasker (CT) Tool is similar, but allows parallel execution of operating system commands actually executed by individual PC clients. The UC and CT Tools are, in their generic form, model-independent. The Process Messages (PM) Tool is specific to management of "error" and "warning" messaging through use of the 3MRA's I/O dll (i.e., the 3MRA Version 1.0 HWIRIO.dll file - the Input/Output Dynamic Link Library; a modular set of routines to simplify the interface between existing and new codes; provides a flexible and expandable storage format for input and output).

The Client Collector Tools are specific to collection of 3MRA ELP1 (Section 6.3.4) data structures across the cluster. An aggregated version of the Client Collector (ACC) has already been constructed and is in use; the disaggregated version (DCC) is associated with disaggregated ELP processing. The ACC Tool is a specific form of a "log-scale" Command Tasker. All the Tools listed generally allow for some type of execution of operating system commands across the cluster. System command calls are either originated: (1) directly from a front-end server in SuperMUSE; or (2) by PC clients under the supervision of such a server.

6.1.4 Tools Facilitating 3MRA Post-Processing Data Analysis Tasks

Auxiliary tools that further enhance 3MRA model data and post-processing of simulation results include:

- Site Visualization (SV) U
- Site Summary (SS) U
- Aggregated ELP1 (for MySQL databases) U
- Aggregated ELP2 (for MySQL databases) U
- Aggregated ELP2Vis (for MySQL databases) U
- Disaggregated ELP1 (for MySQL databases)
- Disaggregated ELP2 (for MySQL databases)
- Enhanced SUI (i.e., Disaggregated ELP User Interface)
- Automated ELP2 Post-Processing Tool for Multiple Databases
- Automated RSA Post-Processing Tool
- Automated TSDE Post-Processing Tool

Tools denoted with U, described in detail in Section 6.4, have been developed and are currently being employed in SuperMUSE calculations for 3MRA Version 1.0. The disaggregated routines and enhanced SUI are currently in development and undergoing initial testing. The three automated post-processing schemes are in the initial software development and testing stages. The SV Tool, delivered as part of 3MRA Version 1.0, is typically applied only in stand-alone execution mode. The other tools listed are amenable to use in either stand-alone or parallel execution modes. Openware MySQL database structures are being employed to replace 3MRA Version 1.0 routines that currently utilize Microsoft MS-Access data structures.

The SV Tool, working upon SSF and GRF files, allows the user to graphically represent data associated with a single scenario. The Site Summary (SS) Tool allows the user to extract data from SSF and GRF files into several database structure types for one or more scenarios, where each Site Summary database record represents data from a single model run.

The planned "3MRA Modeling System Data Analysis Tool Set", encompassing all the above tools, provides needed enhancements to 3MRA Version 1.0 to generally allow for use of openware MySQL databases, and management of those databases across the parallel computing cluster. Initial post-processor enhancements have focused on replicating existing 3MRA Version 1.0 Microsoft Access database structure designs for exit level processor (ELP) routines that aggregate data across sites and realizations. MS-Access, the data structure used for initial development of stand-alone 3MRA Version 1.0 ELP tools, is poorly suited to parallel processing designs. A fundamental approach for Version 1.x is to eventually phase-out the MS-Access data structure approach in favor of more flexible capabilities offered by MySQL data structures that work well for both stand-alone and parallel model execution modes.

Sensitivity Analysis for Exit Level Model Outputs

Currently in 3MRA Version 1.0, using the SS Tool, one can preserve input-output relationships easily only for variables found in SSF and GRF files. For each model run, one can associate unique values for module inputs to unique values for module outputs. Discussed in Section 4.5.8, data compression associated with the standard aggregated ELP1 approach rolls-up GRF data across sites and iterations for scenario sets with more than one simulation.

Thus, in the current 3MRA aggregated ELP1 design, a 1-1 correspondence between input vectors and output data is not maintained across sites, or across Monte Carlo iterations, preventing sensitivity analysis on exit levels. The standard 3MRA Version 1.0 ELP1 aggregated approach was necessitated by the significant data storage needs associated with conducting the many simulations for the national assessment strategy (originally envisioned for a single PC). Disaggregated data structures (Section 6.4.5) allow for preservation of the input-output relationship for protection criteria outputs (Section 4.6) produced by the exit level processing routines. ORD is currently developing an ability to collect disaggregated ELP1 data (Sections 6.3.4 and 6.4.5), maintaining separation of risk summaries across sites or iterations in a single data structure, thus preserving a 1-1 correspondence between input vectors and output vectors needed to conduct sensitivity analysis on exit levels (i.e., per 5 C_w's).

The disaggregated ELP approach will require more direct user control to define a more limited set of exposure profiles produced by 3MRA. This will be needed to limit the overall database size for a given set of scenarios of interest, hence, the need also for an enhanced System User Interface (SUI). Ultimately, the 3MRA sensitivity analysis approach will entail initial assessment with aggregated routines to determine those exposure profiles of greatest concern (i.e., those driving the exit levels). Once identified, exposure profiles of greatest concern could then be focused-on using a disaggregated analysis.

Output Sampling Error (OSE) Uncertainty Analysis for 3MRA Exit Level Outputs

Discussed in Sections 2.6.6, 4.3.3, 4.5.8, and 4.6.1, the standard (aggregated) 3MRA Version 1.0 ELP approach is being utilized with a scheme allowing separation of database structures by iteration (i.e., per national assessment realization or iteration). This allows for quantification of empirical uncertainty and OSE under limited computational constraints, forming a pseudo-2nd order uncertainty analysis. The developmental, automated ELP2 tool noted previously would, for example, allow for automatic calculation and visualization of CDFs of empirical input uncertainty for various population percentiles of variability. The developmental, automated tools for RSA and TSDE sensitivity analysis techniques are also needed, in addition to disaggregated ELP schemes, to more easily perform associated post-processing of sampling-based experiments investigating "exit level" outputs.

6.2 Distributed Execution Program Management Tool Set

As outlined in Section 6.1.2, the following four tools have been developed to facilitate the distribution of workloads among PCs. These are essentially tools that distribute model simulation tasks across the SuperMUSE PC cluster:

- CPU Allocator (CPU_A) U

- Model Tasker (MT) U
- Tasker Client (TC) U
- Client Monitor (CM) U

Shown in Figure 6-1a, with software program locations indicated in Figure 5-2, the "Distributed Management Program Toolset", comprised of the CPU_A, MT, and CT Tools, provides an effective, platform-independent parallelization tool set. Supporting uncertainty and sensitivity analysis evaluation tasks, the distributed processing scheme is capable of managing millions of simulations for 3MRA or other computer models. More specific details and documentation on these tools are presented in Appendix C.

The CPU Allocator and Tasker Client are model independent; a Model Tasker is model dependent, and deconstructs a model's system user interface to generate a set of tasks (e.g., individual model simulations) amenable to distributed processing across the PC-based parallel supercomputer. For example, a Model Tasker was developed for 3MRA, identified in Figure 6-1a as the SUITasker. Several Model Taskers can be active, but must currently reside on separate servers. The Java parallel toolset is readily extended to Linux by additional recompilation of the 3MRA input/output dll (io.dll) called by the SUITasker, and could run 3MRA science modules equally well on Linux if the science modules were compiled for Linux operating systems.

6.2.1 CPU Allocator

(server-end, model independent coordination supervisor)

The CPU Allocator accepts job descriptions from one or several Model Taskers, and provides proportional load balancing across active Taskers (e.g., 50% to MT_1 and 50% to MT_2). Load-balancing is accomplished through uniform random sampling of active Model Taskers when assigning a given PC client to a specific Model Tasker. Random assignment is currently based solely on the number of active Model Taskers, exclusive of the actual number of tasks to be completed by each Model Tasker.

The CPU Allocator functions as a TCP/IP server that registers Model Tasker scenario set descriptions and randomly assigns Tasker Clients to Model Taskers when clients indicate that they are free to execute tasks. Several CPU Allocators can be active, but each must reside on a separate machine (e.g., SuperMUSE versus MiniMUSE; see Section 5.2.4). The CPU Allocator can also be used to restart or shutdown clients without affecting the status of active Model Taskers.

Software development enhancements in progress will let the CPU Allocator set non-proportional load-balancing factors among several active Model Taskers, allowing prioritization of system CPU capacity to individual experiments (e.g., 90% to MT_1 and 10% to MT_2).

6.2.2 Model Tasker

(server-end, model dependent task manager)

A Model Tasker essentially deconstructs a model's system user interface to generate a set of tasks (e.g., individual model simulations) amenable to parallel processing. Several Model Taskers can be active in the system, but an additional server for each additional active Model Tasker is needed, although the first active Model Tasker can be launched on the same server as the CPU Allocator. A Model Tasker has been developed for the 3MRA Version 1.0 modeling system, referred to in Figure 6-1a as the SUITasker (i.e., System User Interface Tasker).

The SUITasker reads a stand-alone 3MRA header file (created in stand-alone mode via the SUI) to define the overall scenario set to be simulated. It functions as a TCP/IP server that accepts Tasker Clients directed to it by the CPU Allocator. Providing slightly different looping than the stand-alone scenario looping (Figure 4-3), the task list is created, managed, and updated with various statistics to track job performance. In this parallel execution mode, looping order follows iteration, site, source, chemical, and C_w. In the original approach, C_w looping was actually contained within a single task delivered to the client (i.e., 5 C_w's, representing five model runs, were delivered as a single task, and run on the client). As discussed in Section 4.6, the final design for 3MRA splits SUITasker tasks by C_w (i.e., 1 C_w is run at a time on a given client) to improve quality assurance of the ELP data structures in the event of PC client hardware or software failures.

The SUITasker maintains a watchdog queuing approach, and can handle errant clients that unexpectedly fail to complete requested tasking (e.g., manages client power failures without client UPS backup). Client-side error trapping criteria can also be used to flag failed tasks (Figure 6-1b) in the SUITasker queue, which can then be reassigned to the general queue. Additional features include the ability to reassign or remove individual tasks from the queue and to specify chemical-specific runtime timeout values. If a client fails to return results within the timeout period, the task is reassigned to another client and the errant client is assigned a temporary failed state of "failed machine" (Figure 6-1c). If a client requests a second job within the first job's timeout period (e.g., the client rebooted due to a power failure), the existing job in the queue originally assigned to the client is simply reissued to that client.

A FUITasker (i.e., FRAMES User Interface Tasker) is planned for development to parallelize the next generation of FRAMES 2.0 modeling system capabilities (see Section 6.8).

6.2.3 Tasker Client

(client-end, model independent task manager)

The Tasker Client is loaded on each PC client at start-up. Each Tasker Client then periodically calls the CPU Allocator when not tasked. If no Model Tasker is active, it is told to idle. The Tasker Client has no user interface, and functions as a TCP/IP client for the CPU Allocator and active Model Taskers. The Tasker Client will connect to the CPU Allocator, receive a Model Tasker machine ID, disconnect, and then connect to the assigned Model Tasker.

It then receives a command from the Model Tasker with associated files to execute. The file set is first written to the client disk, and the model is then executed in batch sequence.

In the case of the 3MRA SUITasker, a master batch file and a single scenario header file (e.g., HdRun1.ssf) are passed to the PC client. Execution is controlled through the master batch file, that: (1) first performs a file cleanup from the last run; and (2) makes a subsequent call to a "run.bat" file, which operates upon the single scenario 3MRA header file. The run.bat file contains commands analogous to the steps that the stand-alone SUI goes through when executing a single scenario. The use of an intermediate run.bat batch file, a common file copied to all clients before the Model Tasker is launched, allows explicit user control over actual system level processor calls made during each model run (e.g., SDP or Site Definition Processor, MMSP or Multimedia Multipathway Simulation Processor, Site Summary, etc.). This approach allows the user to make modifications to the normal SUI execution approach without having to recompile the SUITasker. As an example, wildcards passed through the master batch file can be used to control the location of ELP data storage (e.g., allowing instant segregation of ELP1 data by iteration #).

The Tasker Client will also restart or shutdown the local PC if told to do so by the CPU Allocator.

Example Run.bat File

An example SUITasker run.bat file is shown below where ">" indicates continuation of the DOS command from the line above. In this example, the Access version of the ELP1 is not executed, and the MySQL version is executed. Note how the ProcessMessages Tool is used to perform SUI-type warning and error message management.

```
rem echo off
rem Arg 1 SSF Path (example: \\Machine\Drive\Permanent\Study\Date\Source\Site\CASID\Realization\CW\ssf)
rem Arg 2 GRF path (example: \\Machine\Drive\Permanent\Study\Date\Source\Site\CASID\Realization\CW\grf)
rem Arg 3 Header filename
rem Arg 4 MySQL Server (Destination of Warnings, Errors, & Results Messaging)
rem Arg 5 Study name (e.g., ParSUITest; Note MySQL is case sensitive)
rem Arg 6 Realization (i.e., iteration)
rem Arg 7 Source (i.e., WMU type)
rem Arg 8 Site
rem Arg 9 CASID
rem Arg 10 Cw

del grf\error.
del grf\warning.
del grf\*.grf

rem Run the SDP
start /min /wait c:\hwir\sdp.exe %1 %2 %3

start /min /wait c:\jre\bin\java -cp c:\jre\lib;c:\jre;mysql.jar HWIRNet.ProcessMessages %1 %3 %2\warning. %4 %5
>15 2000 SDP
if exist grf\Error goto ERROR

rem Run the MMSP
```

```
start /wait c:\hwir\mmsp.exe %1 %2 %3
start /min /wait c:\jre\bin\java -cp c:\jre\lib;c:\jre;mysql.jar HWIRNet.ProcessMessages %1 %3 %2\warning. %4 %5
>15 2000 MMSP
if exist grf\Error goto ERROR

rem Run the ELP1 (Access Version)
rem start /min /wait c:\hwir\ELP1.exe %1 %2 %3
rem start /min /wait c:\jre\bin\java -cp c:\jre\lib;c:\jre;mysql.jar HWIRNet.ProcessMessages %1 %3 %2\warning.
>%4 %5 15 2000 ELP1
rem if exist grf\Error goto ERROR

rem Run the ELP1mySQL
rem call c:\hwir\ELP1MySQL.bat %1 %2 %3 %5%6 \\localhost
start /wait c:\hwir\ELP1MySql.exe %1 %2 %3 %5%6 \\localhost
start /min /wait c:\jre\bin\java -cp c:\jre\lib;c:\jre;mysql.jar HWIRNet.ProcessMessages %1 %3 %2\warning. %4 %5
>15 2000 ELP1mySQL /f
if exist grf\Error goto ERRORELP1
echo Deleting backup check file error.txt.....
%comspec% /c del c:\mysql\backup\good.txt

rem Run the Site Summary Tool
start /min /wait c:\jre\bin\java -cp c:\jre\lib;c:\jre;mysql.jar gov.epa.hwir.util.Summary %1 %2 %3 %5.csv
>\\%4\%5.mysql 15 2000
if exist grf\Error goto ERROR

del c:\ParError.txt

goto END

:ERROR
start /min /wait c:\jre\bin\java -cp c:\jre\lib;c:\jre;mysql.jar HWIRNet.ProcessMessages %1 %3 %2\error. %4 %5 15
>2000
del c:\ParError.txt
goto END

:ERRORELP1
start /min /wait c:\jre\bin\java -cp c:\jre\lib;c:\jre;mysql.jar HWIRNet.ProcessMessages %1 %3 %2\error. %4 %5 15
>2000 ELP1mySQL
:END
```

6.2.4 Client Monitor

(client/server-end, model independent Windows-based client manager)

An additional client boot-up tool, the Client Monitor (CM) Tool, currently compiled in Visual Basic, was also developed. It is comprised of two utilities: Launch&Watch (client-end) and Client Monitor (server-end). The server-side user interface is depicted in Figure 6-2.

This tool orchestrates and monitors client boot-up activity to ensure successful launching of local, client-based MySQL servers and any other critical support applications needed to ready clients for Model Tasker jobs. It essentially protects the integrity of active Model Taskers, preventing multiple scenario errors in the event of a non-fatal client malfunction (i.e., client boots, connects, but can’t properly handle tasks). Upon successful support program startups, the client-side tool launches the Tasker Client. The CM Tool is critical for managing power outages, ensuring proper client operation after re-boot. The tool was initially created to provide an easy way to detect the status of all PC clients at start-up or re-boot, and to address occasional problems noted in successfully launching local MySQL servers on PC clients. Notable in 3MRA SuperMUSE operations, all PC clients and centralized servers typically launch a "local" MySQL server service (i.e., identified in "run.bat" command line processing as "localhost").

6.3 File and Data Management Tools

To meet file management tasking needs in operating SuperMUSE, the following tools were identified in conceptual design stages and those with denoted with a U were implemented:

- Update Client (UC) U
- Command Tasker (CT) U
- Process Messages (PM) U
- Client Collector for Aggregated ELP1 (ACC) U
- Client Collector for Disaggregated ELP1 (DCC)

All of the above are model-independent except for the Client Collectors, which are specific to collection of 3MRA ELP1 data structures across the system.

6.3.1 Update Client

(server-end, model independent file manager)

Originally called "propagate", the Update Client (UC) Tool represents a file system management tool employing a dropdown equipment list. It allows the user to write re-usable, command-based, batch file scripts containing special wildcard variables associated with attributes of the PC client set. The server-side user interface is depicted in Figure 6-3. More specific details and documentation of this tool are presented in Appendix C.

For selected machines, the UC Tool executes the set of DOS (i.e., Windows) or shell (i.e., Linux) commands over the network using the wildcard designations for various PC client attributes (e.g., %n=MachineID, %i =IP address, and other various OS configuration variables such as Windows OS version, #hard drives, #CPUs, etc.). The tool facilitates client-list selection, individually or as designated PC groups (i.e., Win98, Win2K) for serial execution of batch scripts across the selected subset of PCs in the cluster, and is executed from a single server. This tool is preferred over the Command Tasker when serial processing is sufficiently quick, for example in distributing small-sized file updates (e.g., 1 to 2 seconds of execution per client, or where only one or a few clients are selected). This tool delivers useful server-side "services" that would otherwise require proprietary server licensing (e.g., executes file management tasks that a Windows 2000 Server license would otherwise be needed to perform).

6.3.2 Command Tasker

(server-end, model-independent batch file manager)

Similar to the capabilities of the Update Client Tool, and actually representing a type of Model Tasker, this tool performs as a server-end batch file manager. The Command Tasker (CT) Tool constructs and implements a binary tree of task dependencies for: (1) collection of common aggregated data/files; (2) or reversibly, distribution of common data/files; or (3) general execution of client-side commands. The server-side user interface is depicted in Figure 6-4.

External to actual Model Tasker and Tasker Client simulation-based operations, the CT Tool was developed to expand file management capabilities. For example, it is used to aggregate MySQL databases from four clients to two clients to 1 one client. This collection-distribution approach can take on the order of $\mathrm{Log}_2(N)$ time instead of on the order of N time (as does the Update Client Tool), where N is the number of PC clients. The Command Tasker acts as a Model Tasker in managing activities across the cluster, allowing the user to issue commands to clients (e.g., DOS commands for Windows or shell scripts for Linux) that are executed by the Tasker Client. The CT tool can be used to issue either single command lines, or calls to a batch file for multiple command line processing.

The Command Tasker reads a comma-delimited text file that lists all tasks to be processed, where dependency on completion of previous tasks in the list can be specified. Extensively generic in form, it is currently used for conducting log-scale database collections for 3MRA experiments, and for more quickly executing file management tasks that take individual PCs substantial time to complete. Supporting utilities allow quick generation of the command list based on PC client equipment profiles.

This tool is relatively new, and is still undergoing documentation and final testing.

6.3.3 Process Messages

(client-end, model dependent file manager)

Providing key connections to a back-end data server, this auxiliary Java application conducts client job processing and generic modeling system message management. The Process Messages (PM) Tool has no user interface. It represents a capability for performing additional calls within the client-side batch file scheme (i.e., run.bat) to manage standard model "warning" and "error" flat files. It reads normally produced 3MRA Version 1.0 <warning.> and <error.> files, and, via Java DataBase Connectivity (JDBC) software, records these in a centralized MySQL data server identified by the SUITasker. While this tool depends on use of the 3MRA Version 1.0 I/O dll, it can be used for other models meeting this I/O specification.

More specific details and documentation of this tool are found in Appendix C. An example MySQL "Error" table, created by the PM Tool, is presented in Figure 6-5.

A looping procedure is also performed at command execution (see run.bat in Section 6.2.3) to allow for multiple retries for establishing a database connection with a set time delay (e.g., 15 retries with a 2000 millisecond delay between retries). Other features include passing of a designated label associated with warning messages (e.g., ELP1MySQL), and an ability to truncate messaging to 80 characters or to allow the full message to be passed (i.e., /f = full). In addition to its documentation function, this tool facilitates the monitoring of scenario set execution on SuperMUSE throughout the experiment.

6.3.4 Aggregated ELP1Client Collector

(server-end, model dependent database manager)

A Client Collector specific to 3MRA was created that utilizes the Command Tasker to effect execution. The Client Collector, a Java-based application, has no user interface. Providing a connection to the back-end data server, this tool facilitates scalability in parallel execution database design for "aggregated" ELP1 processing. Intensive model output database operations, such as the ELP1, can be implemented whenever client simulation processing utilizes local MySQL or other database hosts for interim data storage (e.g., when dealing with >1000s of query operations/minute/client). For 3MRA, this approach was far more efficient than a centralized database server for client-side ELP1 execution. For example, in SuperMUSE, due to the intensive data processing associated with ELP1 execution, 8 PC clients typically represent the maximum number of clients that a single, centralized, Windows-based server can handle without causing delays (i.e., without build-up of ELP1 data delivery tasking in the central server queue).

An example Command Tasker implementation of the aggregated ELP1 Client Collector approach is shown in Figure 6-6a. The associated CPU Allocator with both an active SUITasker and this Command Tasker implementation of the Client Collector is shown Figure 6-6b. A Tasker Client implementation of an example ELP1 MySQL database collection task within SuperMUSE, delivered by the Command Tasker to PC Client #1015C, is shown in Figure 6-6c.

The Client Collector is implemented to collate a single experiment-wide ELP1 database. It is most efficiently run concurrently with the SUITasker, using a separate server to collect ELP1 results for iterations already completed by the SUITasker, while subsequent iterations of individual model runs are still being simulated across the cluster. The efficiency of this dual-run approach is realized after the first collection level is completed, where increasing numbers of PCs would otherwise sit idle. As client databases for a given iteration are continually aggregated, the experiment-wide ELP1 database eventually resides on a single PC client. For example, with 152 PC clients, after the first aggregation across PCs is performed, 76 clients would sit idle, after the second round, 76+38 = 114 clients would sit idle, etc., etc. It takes 8 separate collection levels (i.e., $\mathrm{Log}_2(256)$) to complete the process for between 129 and 256 PC clients. This approach would generally need to be performed after the SUITasker functions are completed for a given 1-dimensional total uncertainty analysis using aggregated ELP1 processing. Even in this latter approach, separate SUITasker jobs could be issued to reduce PC client idle time, for example by separating model runs by chemical or WMU.

This tool, like the Command Tasker, is also relatively new and is still undergoing documentation and final testing. Extensive system-level testing to date of the combined Command Tasker and Client Collector method has shown that the collection routine performs well, with absolute precision observed in either the concurrent collection approach, or in collections performed after the SUITasker has performed all simulations. In the concurrent collection approach, following the example in Figure 6-6c, machine 0602 local MySQL server would be passing ELP1 data to machine 1015C, while it performs a 3MRA model simulation.

6.3.5 Disaggregated ELP1 Client Collector

(server-end, model dependent database manager)

This tool has not been developed yet, awaiting completion of the disaggregated ELP1 routine discussed in Section 6.4.5 below. Once developed, this tool will provide an enhanced connection to the back-end data server/collector for the disaggregated ELP routine. In the disaggregated routine, record sets are added per simulation, and not aggregated. Because fewer records are being stored, and since only record "inserts" are performed, some simulation experimentation may be able to be efficiently handled using a centralized server approach. However, intensive model output database operations using local MySQL server hosts on clients for interim data storage may need to take advantage of periodic record collection, similar to the aggregated approach employed. During system-wide scenario set simulation, the new collector would be implemented periodically to gather the single experiment-wide database structure. Concurrent post-processing may also need to be invoked to manage data storage requirements for smaller hard drives on PC clients. This routine would likely be developed as an option switch within the existing Client Collector Tool.

6.4 3MRA Modeling System Data Analysis Tool Set

Described in the overview in Section 6.1.4, the following 3MRA Version 1.x tools provide enhanced capabilities facilitating post-processing functions and data analysis, where a U denotes tools developed, tested, and implemented to date:

- Site Visualization (SVT) U
- Site Summary (SST) U
- Aggregated ELP1 (for MySQL databases) U
- Aggregated ELP2 (for MySQL databases) U
- Aggregated ELP2Vis (for MySQL databases) U
- Disaggregated ELP1 (for MySQL databases)
- Disaggregated ELP2 (for MySQL databases)
- Enhanced SUI (i.e., Disaggregated ELP User Interface)
- Automated ELP2 Post-Processing Tool for Multiple Databases
- Automated RSA Post-Processing Tool
- Automated TSDE Post-Processing Tool

Unlike the SVT tool, which utilizes GNUPlot graphics representation, MySQL database versions for the ELP1 and ELP2 processors were also created with extended ELP2 visualization capabilities using a scalable vector graphics (SVG) approach. The ELP2Vis tool allows expanded visualization options for large numbers of simulations. Of the ELP tools, only the ELP2Vis is currently constructed as a Java application, where the others are coded in C^{++}.

The disaggregated routines outlined above are currently being developed as enhancements to the existing aggregated routines, and would be invoked through command line argument switches. The older ELP routines supporting MS-Access database structures, currently found in the 3MRA Version 1.0 release, will eventually be phased-out in favor of the openware MySQL database structure approaches.

Automated post-processing routines for implementation of the RSA and TSDE sensitivity analysis procedures also are currently under development, and will generally follow approaches discussed in Sections 2.7.5 and 2.8 (Chen and Beck, 1999; Beck and Chen, 2000; Osidele and Beck, 2001a, 2001b).

6.4.1 Site Visualization

(client-end, model dependent simulation visualization)

Specific to 3MRA, this tool facilitates post-processing data visualization and compilation for standard 3MRA output (*.grf) file sets generated for a single model simulation. The SV Tool offers a limited user interface to define *.ssf and *.grf file paths. It utilizes a GNUplot graphics approach with a coordinated html-based presentation capability for displaying a graphical summary of an individual simulation. It produces charts showing the major outputs of various 3MRA science modules. The SV Tool extracts data from SSF and GRF file sets for an individual scenario and waste concentration, and utilizes 3MRA's I/O dll.

With its integration and statistical processing capabilities (e.g., percentiles, etc), the charts produced via GNUPlot show the major outputs of each MMSP module, starting with the waste management unit source term and ending with human and ecological exposure modules. Collecting the charts into an html file, this site-based tool allows for visual summary of model outputs, spanning the entire MMSP model domain. The SVT tool is further described in the *3MRA Modeling System: Technology Design and Users Guide*, along with presentation of example output. Because of its utility as a tool for stand-alone execution mode, the SVT is actually incorporated within the 3MRA Version 1.0 release. The tool, however, is offered at this point as provisional only, where final documentation and testing is still currently underway.

6.4.2 Site Summary

(client-end, model dependent i/o file manager/extractor)

This Site Summary (ST) Toll, an auxiliary Java application, conducts client job post-processing tasks. There is no user interface involved with the SS Tool. Like the Process Messages Tool, it is expected to execute without user intervention during a set of 3MRA simulations. The Site Summary Tool extracts results from SSF and GRF files, storing user-selected variables defined by an experiment-wide, delimited script file called in the run.bat file. Several calls can be made for a given simulation, allowing separation of many variables into groups stored in separate database tables. A description of variables for each extraction set is created in a separate "Description" table during the first Site Summary call.

More specific details and documentation of this tool are presented in Appendix C. An example MySQL "Description" table created by the Site Summary Tool is presented in Figure 6-7a, and a portion of a corresponding example "Result" Table is shown in Figure 6-7b.

The form of the database name used in the command line controls how the database is updated. Several forms are recognized (i.e., *.csv, *.mdb, and *.mysql). If the database ends in *.csv, an entry will be added to a file on the machine with the given name. If the filename ends in *.mdb, which is a Microsoft Access database, that database will have entries appended to the tables in that database. Finally, if the output database has a name that ends in *.mysql, and is of the form "//server/study1.mysql", then a string following "//" and before the "/" is assumed to be the server and the name following the "/" and before the ".mysql" is assumed to be the database on that server to use.

The Site Summary Tool provides for surgical extraction of all available input and output data, per simulation, facilitating subsequent sensitivity and uncertainty analysis. After each simulation, it allows for the storage of individual parameters from model input and output files, storing values for user-selected variables, or statistics of a variable, in a "Results" table, on either the client's local data server or a centralized data server. If the values read represent a set of numbers, the user has several options for summarizing those values, or, alternatively, writing-out every value.

Statistical-Based Summaries

Statistically based summaries facilitate description of time series data and data across one or more indexes. These summaries allow for parameterization as a normal or lognormal distribution (i.e., reporting the mean and standard deviation) as well as specifying a list of desired percentiles to be reported. While the SS Tool allows extraction of variables of up to 6-dimensions (as individual model outputs for each index permutation), currently, it can only construct summary statistics across three dimensions, using an extraction instruction for each dimension (e.g., n_x, n_y, n_z; See Appendix C). In the case of summary statistics, a separate field is created in the results table for each component of the extraction (i.e., mean, standard deviation), with a corresponding description record for each component created in the "Description" table.

Different sites in the 3MRA database have different index structures (e.g., different sites can have a different number of waterbody networks; a maximum value of 32 can, for example, was found at one site in the current site-based 3MRA database). Because of this, when using the SS Tool across sites, it is necessary to specify the maximum index value anticipated across all sites, and to allow for insertion of "Null" values for missing entities. If only a single site is being evaluated, or the maximum index value is constant across all sites, then the associated "Num-Type" index variables can be successfully used in the extraction command to set the maximum index value expected (i.e., one could use the model input "NumWBN" to set the index length if evaluating a single site).

6.4.3 Aggregated ELP1 for MySQL Databases

(client-end, model dependent risk assessment data assimilation/compression tool)

Specific to 3MRA, this system processor facilitates intensive post-processing data assimilation and data compression to produce profile-level risk assessments, utilizing a MySQL data structure with enhanced data normalization. The newer MySQL approach consolidates data results across all simulations for profiles of interest, similar to the existing 3MRA Version 1.0 ELP1 MS-Access routine (e.g., national risk summaries are aggregated across "Site" & "Iteration"). Using the SuperMUSE parallel execution mode with wildcards in the associated "run.bat" batch file, one can also easily separate databases across "Site" & "Iteration". The MS-Access version was originally based on a data structure using hundreds of tables, with separate databases created for each chemical-WMU combination. One major difference with the MySQL approach is the consolidation of "RSOF" risk summaries (i.e., risk summary output files) into three main tables within a single, common, database structure. These three tables are delineated as:

- RTemplate table; human cancer risk summaries
- HTemplate table; human hazard risk summaries
- ETemplate table; ecological hazard risk summaries

Example data table structures produced by the aggregated ELP1 MySQL version are shown in Figures 6-8b through 6-8r. The overall table set is depicted in Figure 6-8a, followed by the three risk template tables, followed by the scenario description table "SiteIterIndex", and followed by various supporting index tables. The field "tableName" was basically used to normalize the older MS-Access data structure. This created a separate database table, using similar "tableName" field values, for each of the descriptions relating to a given "exposure profile". Note here that based on an internal software programming convention, C_w field values are all shifted down by 1 (i.e., C_w5 and C_w1 are represented as integer values of "4" and "0" respectively, in fields denoted by C_w). In Figures 6-8b, 6-8c, and 6-8d, the numeral counts in bins represent the total number of simulations stored thus far that were protective for the given exposure profile and percentage of population.

This tool is currently undergoing documentation and system-level testing.

6.4.4 Aggregated ELP2 for MySQL Databases

(client-end, model dependent risk assessment data visualization/compilation tool set)

Specific to 3MRA, this system processor facilitates intensive post-processing data visualization and data table compilation for overall simulation experiment-wide risk assessment results, utilizing the collected ELP1 database. The newer MySQL ELP2 routines actually represent two software tools: a standard interface similar to the 3MRA Version 1.0 ELP2 MS-

Access routine, and an enhanced visualization interface that extends the original Risk Visualization Processor in the ELP2 (i.e., 3MRA Version 1.0 Elp2.exe). These two routines operate on the MySQL ELP1 database discussed in Section 6.4.3.

ELP2Vis for MySQL Databases

Visualization routines utilize a scalable vector graphic approach with a coordinated html-based presentation capability for displaying various profiles of interest. The ELP2Vis processor is Java-based, and allows for visualization of multiple exposure profiles in a single summary. An example of ELP2Vis html summary output for Benzene in landfills is provided in Figure 6-9, which provides two exposure profiles side-by-side. Figure 6-10 presents the same information, less thumbnail graphics. The C_w values presented above the various risk summary graphics represent the calculated exit level for the exposure profile defined by the scenario. The scenario description table used to generate this output is summarized in Figure 6-11. This table was generated within the ELP1 database structure, and is used by the ELP2Vis to describe exposure profiles selected by the user.

Graphics of the html summary for Scenario 1 (i.e., the first exposure profile description) are enlarged in Figure 6-12a through Figure 6-12j. In electronic form, these are directly accessed by selection on the html page, where the detailed html page places the pertinent human health or ecological scenario description information directly above the actual graphic. For data markers that fall along the right side y-axis, this indicates that the actual C_w is greater than that shown. The Benzene-landfill example, also discussed in Section 7, is based on calculations for 100 national realizations performed using the January 2002 3MRA Developers Version 1.0. For the 100 iterations used in this analysis, OSE uncertainty was not generally addressed in Figure 6-12.

The enhanced visualization capability provides useful information for the analyst and decision-maker alike in understanding the various effects of changes in distance, risk level criteria, etc. The ability to show multiple exposure profiles of interest is also helpful in describing the effects of choosing values for various decision variables under the control of the decision-maker.

As a final task to be completed in development of the aggregated ELP2Vis processor, an automated routine is needed to facilitate data analysis tasks for multiple ELP1 databases. This enhancement, to be constructed in the form of an output data structure, will essentially store graphical data for the curves shown in Figure 6-12, by iteration or site. This will allow analysis of wastestream concentration exit level data by iteration and site, providing an ability to calculate empirical uncertainty and OSE associated with the sampled ELP1 database.

ELP2 for MySQL Databases

This processor provides for standard data table generation and single profile analysis similar to the existing 3MRA Version 1.0 ELP2 for MS-Access databases. Like the ELP2 for MS-Access databases, this routine provides experiment-wide wastestream concentration exit level calculations (i.e., a "threshold" risk-based waste concentration) for various distances from the source, pathways, receptors, cohorts, and risk criteria. Output is keyed to % of protection at sites across multiple sites, using variable risk/protection criteria selected by the user. Like the MS-Access version, the MySQL version utilizes consolidated (aggregated) data, providing a single wastestream concentration exit level for each profile-criteria combination of interest, for each chemical-source combination. The *3MRA Modeling System: Technology Design and Users Guide* provides examples of standard ELP2 tabular and graphical outputs.

The two MySQL-based ELP2 tools described here are currently undergoing final documentation and system-level testing.

6.4.5 Disaggregated Exit Level Processor Set

A basic approach in 3MRA sensitivity analysis will be to use aggregated data structures for consideration of a set of sites, determine the dominant exposure pathways, media, and receptors with greatest risk profiles (i.e., determine those profiles driving final exit level calculations), and then use a disaggregated ELP simulation approach to evaluate in more depth those model inputs that generate the most sensitivity. This would be done on a chemical-WMU basis, and involves simulating the same set of scenarios twice. One of the driving forces behind development of this disaggregated approach is the current destruction of the mapping between individual input vectors and the risk summaries currently saved in the aggregated ELP1 approach. In the aggregated ELP1, many input vectors (i.e., the set of input vectors describing all sites and iterations sampled) are all currently mapped to a single wastestream concentration exit level calculation.

The enhanced disaggregated processor set is broken down into three components. Upon its completion, it will encompass capabilities for performing both aggregated and disaggregated analysis. Similar to the aggregated approach, an ELP1 and ELP2 are used. In this case however, an enhanced system user interface is needed to provide for decision-making prior to simulation of the desired scenario set. This is equivalent to the post-simulation decision-making currently performed in the existing 3MRA Version 1.0 standard ELP2 interface, in selecting a given exposure profile of concern to generate a given wastestream concentration exit level. Thus, the disaggregated approach will move the ELP2 exposure profile decision selection process to the front of the simulation process. This will be accomplished by moving those portions of the existing standard ELP2 panel to a panel in the SUI. The subsequent disaggregated ELP2 would then only be able to select exposure profiles that had been already simulated. In 3MRA Version 1.0, all exposure profiles are calculated and stored in the aggregated ELP1 database structure.

Eventually to be implemented, an additional approach will be provided within this tool set to summarize risk across total populations, in addition to the current capability that aggregates populations on a site protection basis. For example, this would provide for calculation of a single national protection criteria based on the entire national population, as opposed to the existing dual-criteria approach currently implemented in 3MRA Version 1.0 which provides for a given population protection level (e.g., 90%) at a given percentage of sites (e.g., 95%). Details of the latter are outlined in Marin *et al.* (1999, 2003).

Exit Level Processor User Interface

(server-end, model-dependent risk assessment user interface tool)

Specific to 3MRA, this "disaggregated" module set (SuperELP-UI, SuperELP1, and SuperELP2) facilitates intensive post-processing data assimilation for gathering more directed profile-level risk assessment results, on a simulation-by-simulation basis, utilizing a MySQL data structure. The primary purpose, a "disaggregated" database of results generated will preserve a 1-to-1 correspondence between input and output vectors (per simulation). Across selected wastestream concentrations (C_W), the associated exit level ($C_{W_{exit}}$) for any given, otherwise unique model input vector, is comprised of 5 simulations that vary C_W. The UI tool will provide for user specification of selected exposure profiles of interest, allowing for a more limited list of risk concerns to evaluate. Essentially, this implements choices on the existing ELPII UI selection panel used for generating cumulative distribution functions. The enhanced UI will also provide analysis of centralized data storage costs for a given scenario set, prior to launch, and will generally allow the user to specify 1 or more distances, 1 or more pathways, 1 or more receptors, and 1 or more cohorts, etc.

A draft example of the enhanced SUI panel for disaggregated ELP processing is shown in Figure 6-13. Figure 6-13a shows the panel for specifying human exposure profiles and Figure 6-13b shows the panel for specifying ecological exposure profiles. Aggregated approaches are to be encompassed within the SuperELP processor set, offered as a selection choice in the enhanced SUI. Figure 6-13c shows the panel for specifying aggregated or disaggregated processing for a given SUI implementation.

Without loss of generality, only features of the disaggregated routine are detailed further in the following.

SuperELP1

(client-end, model-dependent risk assessment data assimilation tool)

Specific to 3MRA, this enhanced processor will facilitate intensive post-processing data assimilation for the disaggregated, profile-level risk assessment results. It represents a fundamentally different data structure, where risk bins will not be aggregated across sites and realizations, but instead, actual population counts will be recorded for each bin, creating a separate set of database records for each model run. For parallel execution, the disaggregated data structure approach may rely on periodically invoking an enhanced Client Collector, discussed previously, while experiment-wide simulations are actually being conducted.

A draft example of the disaggregated ELP1 data structure for the RTemplate table is shown in Figure 6-14a, and the equivalent disaggregated SiteIterIndex table is shown in Figure 6-14b. In comparing aggregated versus disaggregated database structures (see Sections 4.5.7 and 4.5.8), recall that the updating of individual "risk-bin/C_w" or "hazard-bin/C_w" fields in the standard aggregated ELPI data structure reflects the basic determination per model run that the given population percentile was (i.e., add 1) or was not (i.e., add 0) protected for the given exposure profile defined by "tableName". In the aggregated approach, the last risk/health bin for each C_w and 0% population protection stores the number of model runs that have been considered, by chemical and WMU-type, allowing calculation of the % sites protected for the global scenario set considered. In the disaggregated data structure shown in Table 6-14a, where separate record sets are created per model run, population percentiles remain embedded in the actual population counts. In the disaggregated approach, the last risk/health bin for each C_w would store the total number of receptors for each the exposure profile, for each model run.

SuperELP2

(client-end, model-dependent risk assessment data analysis and visualization tool)

Specific to 3MRA, the enhanced ELP2 processor will facilitate intensive post-processing data visualization and data table compilation for disaggregated results summaries. It will utilize the collected SuperELP1 MySQL database, and will allow for calculation, visualization, and tabulation of $C_{w_{exit}}$ across C_W for each unique simulation in the Super ELP1 database. It also would allow for subsequent generation of aggregated results for the selected exposure profiles across sites or across populations, and across realizations, in essence collapsing disaggregated results to aggregated results that would otherwise be provided in an aggregated simulation approach.

6.5 UA/SA Sampling Tools

Not discussed in the overview Section 6.1 for 3MRA Version 1.x, auxiliary tools that also may eventually be developed, depending on need, that would further enhance 3MRA model input sampling capabilities include:

- Distribution Statistics Processor
- Enhanced Stat.dll
- Latin Hypercube Sampling Tool
- Other Model Input Sampling Tools

6.5.1 Distribution Statistics Processor

The Distribution Statistics Processor (DSP) is accounted for in the design of 3MRA Version 1.0, but the associated processor itself has not yet been developed to date. This processor would randomly sample from statistical distributions representing measurement uncertainty and sampling error related to the statistics of parameters required by the models (e.g., performs sampling of distribution type, mean, standard deviation, and range, etc,). This aspect was extensively discussed in Section 4.6.1, where the DSP would be needed to implement a two-dimensional uncertainty analysis (Figure 2-9) to account for empirical uncertainty in description of random model inputs in 3MRA (i.e., ISE).

The DSP would receive input from the available regional and national environmental setting distribution statistics databases. Output from the DSP would populate the statistical portion of the regional statistics and national statistics databases. Regional and national environmental setting distribution statistics databases (see the Database panel in the SUI) represent a collection of data containing stochastic parameters (a subset of the Site Definition File data) whose stochastic characteristics are themselves described by statistical distributions. Specifically, one database contains information on environmental parameters collected for specific regions of the country, identified by latitude/longitude descriptions. The other contains information on environmental parameters at a national level.

The Site Definition Processor (SDP) would use the data generated by the DSP. The SDP of 3MRA Version 1.0 is already constructed to take advantage of DSP output once made available in the associated distribution databases. Otherwise, the SDP defaults (as it does now) to use of static national and regional databases. Detailed data structure designs for the regional and national distribution databases are presented in Appendix D, which is excerpted from the 3MRA background document *Documentation For The FRAMES-HWIR Technology Software System, Volume 8: Specifications*.

6.5.2 Enhanced 3MRA HwirMC.dll and Stat.dll

Enhancements to the existing 3MRA Version 1.0 SUI, SDP.exe, Stat.dll, and HwirMC.dll routines (see Section 4.3.4) would be the easiest approach for implementing simpler sampling designs, such as those used in OAT designs. Currently, for example, there is no switch that can be set to request that the modeling system perform a deterministic model calculation at a given site using values of the central tendencies of the random input variables. This can only be manipulated currently through modification of associated records in the various databases used by 3MRA. For example, to implement a fully deterministic run, one could set all "distribution types" in the 3MRA database system to "constant".

While not contemplated for the immediate future, enhanced SUI, SDP.exe, Stat.dll, and HwirMC.dll routines could eventually be constructed to allow for more flexible management of sampling-based experiments in 3MRA, for example, for investigating screening level sensitivity analysis techniques (Section 2.7.3).

6.5.3 Latin Hypercube Sampling (LHS)

Latin Hypercube Sampling was discussed in Section 2.5, where other sampling designs were described for various sensitivity analysis techniques (e.g., variance-based; FAST, Sobol's Method, etc.). A form of the original Iman and Conover (1982) LHS routine is available in beta 3MRA 2.0 (Figure 1-1; Section 6.8). Eventually ORD would envision implementation of an LHS for 3MRA, possibly also within 3MRA Version 1.x. This would be useful in further evaluating multiple chemicals or metals on SuperMUSE, or individual ones in more depth. Informed by the UA/SA approach outlined in Section 9, LHS may reliably be able reduce the overall computational burden for both stand-alone and parallel execution modes.

Ideally the LHS to be implemented will provide for both a median interval-sampling scheme as well as a random sampling scheme within a given interval. Implementation requires interface with the SDP for 3MRA applications, and modification of SUI and FUI interfaces, respectively to allow for selection of the number of intervals and median versus random scheme. ORD is also currently working with Sandia National Laboratories to obtain a contemporary LHS source code, based on an updated Sandia algorithm. Acquisition of this source code will depend on final availability. Generally, ERD/NERL/ORD preferably seeks unrestricted access to source code for any particular sampling methods to be used in 3MRA in support of regulatory contexts.

6.5.4 Other Model Input Sampling Tools

The following additional sampling tools are under some form of research and development within ERD/NERL/ORD, and would eventually be brought to bear upon 3MRA type model evaluation strategies in the future. See Section 2.7 for additional information on various sensitivity analysis techniques referenced.

Uniform Coverage By Probabilistic Rejection (UCPR)

As stated in Section 2.7.5, the main purpose of the UCPR procedure is to systematically search the input factor domain for combinations of values that produce *behavior* {B} simulations. UCPR enhances the statistical power of TSDE by augmenting the sample of behavior-producing input factors. In a series of iterations, the UCPR (Klepper and Hendrix, 1994) progressively updates the set of behavior-producing input factors by selecting trial values in close proximity (distance-wise) to the current set. Investigation of the tool is currently being undertaken by ERD, within a research context, for potential development of a combined RSA-TSDE-UCPR sensitivity analysis methodology.

One at a Time (OAT) Sampling Routine

A tool implementing generic OAT designs and 1st and 2nd order Morris' sensitivity analysis algorithms (Saltelli *et al*., 2000).

Factorial Sampling Routine

A tool for implementing generic, factorial sampling designs, including Andres' Iterated Fractional Factorial Design (IFFD) (Saltelli *et al*., 2000).

LP_τ and Winding Stairs Sampling Routines

A tool for implementing a LP_τ sequence sampling design, and a tool for implementing a winding stairs sampling design, both for application to Sobol's sensitivity analysis algorithm (Saltelli *et al.*, 2000).

Fourier Amplitude Sensitivity Test (FAST) Sampling Routines

A tool for implementing various transformation-based sampling design strategies for application to the FAST algorithm (Saltelli *et al.*, 2000).

Monte Carlo Markov Chain Routine

A tool for implementing a modified Monte Carlo sampling design based on use of Markov Chains to reduce the overall burden associated with a random sampling scheme (Kuczera and Parent, 1998).

6.6 Initial 3MRA Time Trial Analysis

Assessing model runtime for various experimental designs is a critical step in apportioning computing resources for any given set of experiments, and is needed to evaluate the range and complexity of experiments that might eventually be feasibly undertaken to evaluate the 3MRA modeling system. Average 3MRA runtimes can vary between 50 and 300 seconds depending on chemical evaluated. However, some individual simulations can take up to 24 hours depending on waste site and chemical/metal selected, and the PC client assigned, i.e., those with slower CPU speeds of 333 MHz.

6.6.1 Initial 3MRA Benchmarking

To properly benchmark the impacts of runtime added due to the parallel processing scheme, a comparison of 3MRA runtimes was made between stand-alone and parallel execution modes on SuperMUSE. Using 85 identical 1 GHz PC clients in SuperMUSE for parallel computations, a scenario set describing the 201 national assessment sites was selected representing all 419 site and waste management unit combinations. One chemical was chosen with 5 assumed C_w values, using only a single iteration. In this analysis, the Site Summary Tool captured a total of 61 input and output variables.

Shown in Figure 6-15, the average overhead runtime cost due to paralleling the 3MRA code was 6.0 seconds/simulation for full messaging capabilities, or 7.2% of the average stand-alone model runtime for Carbon Disulfide. A more direct comparison of stand-alone PC versus SuperMUSE, with maximum storage turned off and no message or Site Summary Tool use, showed an increase of only 0.57 seconds/simulation, or a relative cost increase of 0.7% over average PC model runtime.

In another 3MRA runtime comparison, representing similar parallel execution totaling 90,085 simulations, the average runtime for 15 metals and 28 organic chemicals, with full messaging and processing, was 120 seconds compared to an average of 90 seconds for Carbon Disulfide. On average, SuperMUSE could currently complete roughly 4 million similar MMSP simulations per month, assuming an average of 20 seconds for the SDP per run. Such execution in stand-alone mode, using a few PCs, would be prohibited by the:

- Actual time required to execute a given scenario set in stand-alone mode,
- Need to optimize job assignments across PCs, and
- Human capital needed to collect and collate errors, warnings, model input data, and model results.

For the time trials and uncertainty analyses discussed in Section 6.2 and 7, respectively, an interim 3MRA Version 1.0 (Developers Release - January, 2002) was used for all simulations.

6.6.2 Num-Type Site Layout Parameter Regression Analysis

Several key site description factors were evaluated to develop a better capability to predict model runtime for any given set of 3MRA scenarios that may be executed under the UA/SA plan. In addition to key chemical properties, 3MRA "Num-type" parameters, descriptive of site-layout configurations, shown in Table 6-1 as potentially important regressors, were evaluated against overall average model runtimes (i.e., Total MMSP runtime).

Model results were analyzed using a combination of parametric and non-parametric statistical tests. A multiple linear regression model was developed to predict runtime of any model scenario set selected. A handful of chemicals were varied in this analysis to evaluate their associated impacts on runtime. Because various chemical properties are dependent on chemical scenario selection, chemicals were selected judiciously to cover a wide range of chemical behaviors *in situ* (e.g., VOC, SVOC, metal). The chemicals used in the initial analysis included:

- Carbon Disulfide
- Benzene
- PCE
- Nitrobenzene
- Hexachlorobutadiene
- TCDD (Dioxin)
- Benzo(a)pyrene
- Mercury
- Arsenic

Based on 10,000 simulations, graphical results of the runtimes obtained from the comparative time trial analysis using the above test chemicals are shown in Figure 6-16. The corresponding coefficients of variation for each chemical shown are in Figure 6-17. Multiple linear regression results are summarized in Table 6-2. These analyses showed generally that:

1. The two strongest "Num" regressors for runtime were #Human Receptors & #Waterbody Networks (WBN). Little additional information was gained with 3 or more parameters.
2. Using "Num" type parameters, predicting runtime for
   a. Semi-volatile organics = good
   b. Volatile organics = fair to poor

   c. Metals = poor
3. Vapor pressure showed a generally strong correlation as a predictor of runtime between metals (slower) and both weakly and strongly volatile organic compounds (fastest). However, moderately (semi)-volatile organic compounds were characteristically unpredictable, having the overall slowest runtimes observed.

This 3MRA runtime analysis is, of course, preliminary and incomplete relative to assessing all potential model inputs that can drive average model runtimes. Nonetheless, the analysis shows the difficulty in predicting runtimes for very high order models across multiple chemicals.

6.6.3 Chemical Specific Runtimes

A more recent runtime analysis using the final 3MRA Version 1.0 software release provides a perspective of overall model runtimes experienced across the 43 chemicals currently parameterized in the 3MRA database. Total MMSP runtime (i.e., the time needed to run the 17 science modules) by chemical is shown in Figure 6-18. Total MMSP runtime by source type is shown in Figure 6-18. This analysis was based on 447,284 simulations conducted with the current SuperMUSE configuration, representing five national realizations for 43 chemicals. A chemical CASID cross-reference list is provided in Table 6-3.

Across all sites, sources, iterations, chemicals, and C_w's, the average MMSP runtime was 99 seconds, with a standard deviation of 132 seconds. These runtimes exclude SDP, ELP1, and Site Summary runtimes, which for this experiment averaged 86 seconds total, for an average total 3MRA Version 1.0 runtime per scenario of 185 seconds. In general, the SDP and ELP1 runtimes were typically 20 seconds each, but in this experiment, over 1000 Site Summary extractions were also performed, accounting for the additional 46 seconds of runtime.

The two chemicals with the fastest average MMSP model runtimes were Thiram and Divalent Mercury, at 52 seconds and 57 seconds, respectively. The chemicals with the slowest average MMSP model runtimes were Dibenz[a,h]anthracene and 2,3,7,8-TCDD, at 182 seconds and 185 seconds, respectively. The two chemicals with the slowest, maximum MMSP runtimes were Hexachlorobutadiene and Arsenic, at 13,367 seconds (3.7 hours) and 10,114 seconds (2.8 hours), respectively.

An ELP1 collection across SuperMUSE per national realization for the 7 chemicals to be studied in the UA/SA plan outlined in Section 9, if simulated concurrently on SuperMUSE, entails 151 individual collection tasks at an average runtime cost of 680 seconds per task. Ignoring MySQL processing time on the remote source PC client per task, this would entail, on average, 11 minutes (151*660/60/152PCs) of the entire SuperMUSE computational capacity per national realization, or roughly 2 days per 100 realizations for the 7 chemicals. Adding roughly 21 days of simulation time for the 100 national realizations for aggregated ELP1 processing, the total computational cost (runtime) would be approximately 23 days at existing SuperMUSE capacity levels, excluding MiniMUSE (Section 5.2.4).

6.6.4 MMSP Runtimes by Source

As shown in Figure 6-19, the slowest MMSP runtimes, by source type, were generally ordered by: land application units, landfills, waste piles, and surface impoundments and aerated tanks that were about the same. All source types showed predictably slower runtimes at the higher wastestream concentrations; average runtimes for the aerated tanks were the most invariant to changes in wastestream concentrations. It was also generally observed that landfills had the greatest variability in runtime ranging from the lowest to highest C_w. Lower C_w's for landfills showed less variability in runtime than lower C_w's for land application units, although these roles reversed at the higher wastestream concentrations.

6.6.5 MMSP Runtimes By Science Module

In Figure 6-20, the relative percentage contribution to average MMSP runtime is broken-down by science module. Individual average module runtimes across the 43 chemicals studied are further described in Figure 6-21. Two of the seventeen science modules in 3MRA, the human risk module (27.7%) and the surface water module (23.1%), account for half of the total runtime. The human exposure module ranks third with an average allocation of 15.8% of total MMSP runtime. The data shown in Figure 6-21 represent statistics of average runtimes generated chemical by chemical. Generally, variance in runtime increased with average runtime by module except most notably for the watershed module.

Coefficients of variation (CV = StdDev/Mean) were all less than 1. The CV was highest for the ecological risk module (0.92), followed by the terrestrial foodweb module (0.77) and the aquatic foodweb module (0.73). Other module-specific CVs averaged 0.41. This analysis pooled data across source types. Evaluating specific source types, the surface impoundment module showed the least variation in average runtime across chemicals, with a CV of 0.019, followed by waste piles (0.065), aerated tanks (0.21) and landfills (0.73). Due to a problem associated with the extraction instruction set used for capturing Site Summary results for this experiment, runtime data for the land application unit module itself was omitted from the above analysis.

6.6.6 Solubility Errors Encountered During Runtime

For the calculations shown in Figures 6-18 through 6-21, the actual full scenario set launched was 5*90,085 or 450,425 runs. The difference between launched runs and successful runs is attributed to runtime errors associated with media solubility limitations. Solubility limitations in the 3MRA modeling system are most frequently associated with source terms, but may also be encountered in various fate and transport modules (e.g., vz, aq, sw, etc.). These limitations do not represent model errors, but rather limitations of the model to handle conditions encountered. Solubility errors that can be encountered during model runs are more fully delineated in the *3MRA Modeling System: Technology Design and Users Guide*. A file with examples of actual solubility error messages reported by the SUI was made part of the released Version 1.0 file set.

6.7 System Level Quality Assurance Testing Via SuperMUSE

In initial testing of preliminary beta versions of 3MRA 1.0 on SuperMUSE (i.e., Version 0.98r and the developers' version released in January 2002), and as a result of visualization software developed to support 3MRA evaluation, many previously unknown software errors were identified and have been corrected. These represent errors that were not detected through the relatively rigorous component level Quality Assurance Testing Plan, executed as part of the 3MRA modeling system development plan. In testing of the beta 3MRA Version 1.0 since early 2003, only three additional logical errors (Section 6.7.1) were detected. These errors have also been corrected. Across chemicals, 3MRA currently averages <0.27% scenario runs resulting in incomplete calculation processing, due only to solubility limitations of various media/modules.

Since incrementally embarking on SuperMUSE development and its expansion, beginning in early 2001, it is estimated that to date over 36 million 3MRA simulations have been executed (counting Version 0.98r, beta Version 1.0, and final Version 1.0). Software testing typically follows an exponential decay curve in manifestation of previously undetected logical coding errors. As a result of this extensive system-level testing program, today there are currently no known logical errors in the 3MRA Version 1.0 software release, and there is increasingly higher confidence that few, if any, logical errors will be detected at runtime in the near future. Few software systems that have achieved solution of such a complex, integrated environmental modeling problem can state this. The approximate allocation of SuperMUSE capacity for 3MRA model evaluation tasking is depicted in Figure 6-22.

Nonetheless, model evaluation and additional testing will continue vigorously into the future to further assert increasing confidence in the modeling system. However, from the viewpoint of the 3MRA decision-maker, the focus of SuperMUSE now begins to move from extensive system-level verification tasking to the more formal, and readily more useful, endeavor of evaluating uncertainty in 3MRA predictions, and the identification of sensitive model inputs.

6.7.1 Problems Corrected as a Result of System-Level Verification Tasking

As one example of model verification results derived from system level model comparison efforts conducted thus far, an error in 3MRA was identified early-on when comparing stream baseflow discharge estimates between 3MRA and TRIM. The error was a result of improper scaling of coefficients of the baseflow equation used by 3MRA, which was corrected. As a set of other examples of verification tasking, more than 30 programming errors in the original August 2000 beta 3MRA Version 1.0 were also identified through the extensive system level simulation via both stand-alone and SuperMUSE parallel execution modes. These errors, by and large related to problems at the integrated system level, have also been corrected.

One of the last few errors identified to date, an unforeseen convergence of input conditions that the aquifer module was not previously programmed to handle, was identified and corrected in 2003. Finally, through testing of the random number generator, some non-random patterns were identified related to the previous manner in which the SDP was generating seed values for the national assessment (see Section 4.3.4). A revised routine within the SDP was recently developed and tested to correct the problem. Exclusive of the latter random seed error more recently identified, the final release of 3MRA Version 1.0 has undergone additional simulation testing, through over 5.5 million additional model runs, without additional error. Aside from the aquifer module error, random seed error, and an error in the AirIO model described next, over 12 million simulations have been conducted without identification of additional programming errors.

Error Detectable Only Through Random Parallel Simulation

An important point emphasized next in verifying large or complex modeling systems is the benefit gained from multiple, concurrent approaches used for model verification undertaken by the 3MRA development team. Collectively, these cover unit–level verification studies described in Volume III and the system-level testing studies derived from system-level evaluations, via both stand-alone and parallel execution modes.

As an example of this later point, despite extensive testing approaches offered by unit-level testing and stand-alone execution mode system-level testing, parallel processing identified imprecision in ELP1 data results. This was originally conjectured to possibly be attributable to the mixing of different operating systems on SuperMUSE (i.e., Windows 98 and Windows 2000). ELP1 data results were first noticed to have slightly different "protected site counts" in results simulated in a variety of ways across these two operating systems.

As an alignment of coincidence and the parallel execution scheme, investigation into this phenomena eventually revealed that the naming convention originally adopted for LFO files generated by the air model (i.e., AirIO.exe, which calls the legacy air model ISCST) actually imparted an approximation of the air model outputs otherwise thought to be uniquely described for a given set of input conditions. The imprecision was determined to be unrelated to operating system choice, or the ISCST model. Because of the intended efficiency of re-use of LFO files for aerated tanks, which can be different from realization to realization for a given site, it turned out that stand-alone execution mode likely would have never detected this error. Since the normal stand-alone approach, which is implemented as serial execution of a set of model runs, is not actually simulated as such on SuperMUSE, the error was readily identified in precision testing using SuperMUSE's parallel execution mode.

While not comprehensive in detailing the massive effort and information gathered since the 1995 HWIR review by SAB, in both developing and testing 3MRA, the previous text highlights the paramount importance that OSW and ORD have assigned in systematically implementing a thorough program of model evaluation needed to assert the highest level of quality assurance practical in 3MRA's composition and performance for its intended purpose.

6.8 FRAMES 3MRA Version 2.0

In initial beta testing, FRAMES Version 2.0 will eventually house 3MRA, and also additional models, modeling systems, and databases, and offers a number of enhanced capabilities over FRAMES 3MRA Version 1.0 (see Figure 1-1). First and foremost is the ability to conceptualize a risk assessment on the fly. In the FRAMES 3MRA Version 1.0, the

Conceptual Site Model (CSM) is stored in a database that makes it more difficult to change, since connections between models are strictly confined to a particular subset of simulation alternatives in Version 1.0. Free-form CSM definition will provide the user with the ability to recombine 3MRA components, and other models or databases in FRAMES 2.0, in different ways to answer alternative risk assessment questions without having to modify existing 3MRA Version 1.0 components.

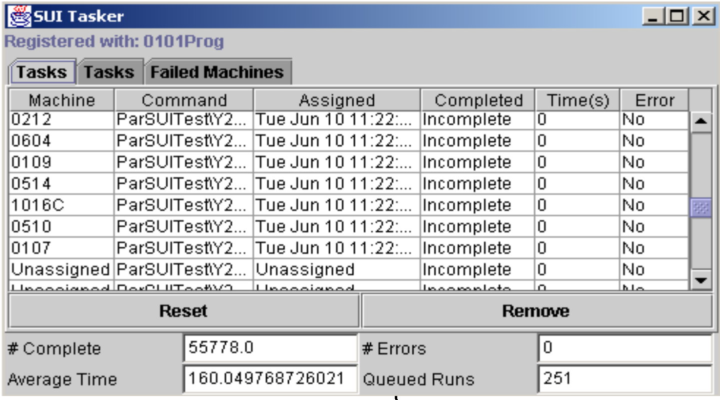
Other key enhancements found in the FRAMES 2.0 architecture include automated units conversion, a user interface for site-specific data entry, and an overall more flexible framework that provides for relatively rapid importation of new models and databases. Germane to this work is the expansion and facilitation of tools used to conduct uncertainty and sensitivity analyses, including an n-stage iterator that will allow nesting of Monte Carlo sampling design, and inclusion of alternative sampling strategies such as Latin Hypercube (Helton and Davis, 2003). For example, the n-stage iterator could be used to facilitate multi-dimensional uncertainty analysis. Together, the enhanced 3MRA 2.0 features will facilitate a much-needed capability to jointly explore parameter, model, and modeler uncertainty within a common architecture. For example, model error can be explored in FRAMES 2.0 via a framework capability easily facilitating multiple module (model) comparisons as well as model uncertainty via this same interface, thereby allowing for evaluation of alternative modeler decision-making under the constraints of a common data set and a common set of models to select from, all acting upon a unique problem description

Together with future development of FRAMES 2.0 type framework designs, the Java-based paralleling software tool set developed or under development for uncertainty and sensitivity analyses will continue to move towards platform-independence, and can be readily extended to other NERL and EPA exposure and risk assessment models and modeling systems. This is an important consideration given that EPA is using hundreds of models to assess environmental issues. With future application migration in mind, the UA/SA tool set concepts presented in Sections 5 and 6 represent a key component need for future Agency modeling frameworks and a high priority model development activity.

Model Tasker

Model dependent – Create an MT for each model

. (MT$_1$ launched on Program Server 0101Prog)



SUI Tasker

Registered with: 0101Prog

Machine	Command	Assigned	Completed	Time(s)	Error
0212	ParSUITest\Y2...	Tue Jun 10 11:22:...	Incomplete	0	No
0604	ParSUITest\Y2...	Tue Jun 10 11:22:...	Incomplete	0	No
0109	ParSUITest\Y2...	Tue Jun 10 11:22:...	Incomplete	0	No
0514	ParSUITest\Y2...	Tue Jun 10 11:22:...	Incomplete	0	No
1016C	ParSUITest\Y2...	Tue Jun 10 11:22:...	Incomplete	0	No
0510	ParSUITest\Y2...	Tue Jun 10 11:22:...	Incomplete	0	No
0107	ParSUITest\Y2...	Tue Jun 10 11:22:...	Incomplete	0	No
Unassigned	ParSUITest\Y2...	Unassigned	Incomplete	0	No
Unassigned	ParSUITest\Y2	Unassigned	Incomplete	0	No

# Complete	55778.0	# Errors	0
Average Time	160.049768726021	Queued Runs	251

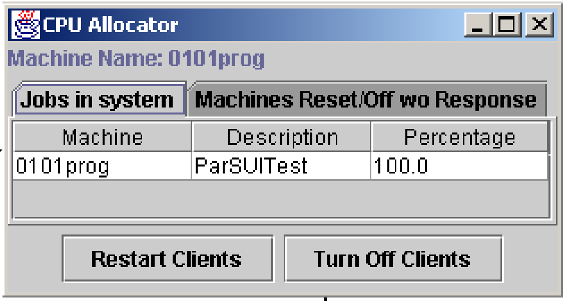
At launch, MT's register with the CPU Allocator.

CPU Allocator

Model independent.

(launched on Program Server 0101Prog)

Oversees Model Taskers (MT$_i$; i = 1…n$_t$)



CPU Allocator

Machine Name: 0101prog

Machine	Description	Percentage
0101prog	ParSUITest	100.0

1. TaskerClient TC$_1$ announces availability.

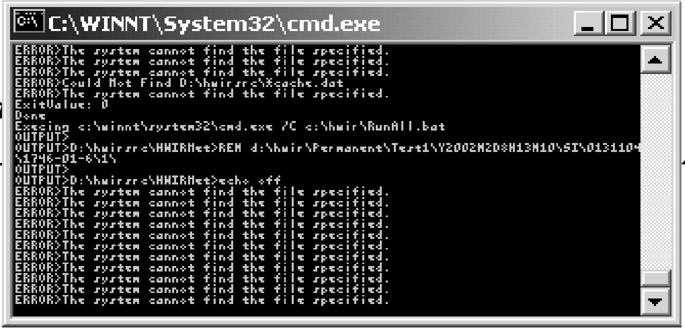
2. If no MT, then idles. This TC$_1$ call was assigned to MT$_1$.

3. TC$_1$ requests job from MT$_1$.

4. Do job X ≡ a single task line in any MT. For the 3MRA Tasker (i.e. SUITasker), represents a unique scenario.

5. Send job X warnings, errors, and results to Data Server's project file area identified by MT$_1$ (e.g. MySQL ParSUITest database).

6. TC$_1$ says done with job X.



C:\WINNT\System32\cmd.exe

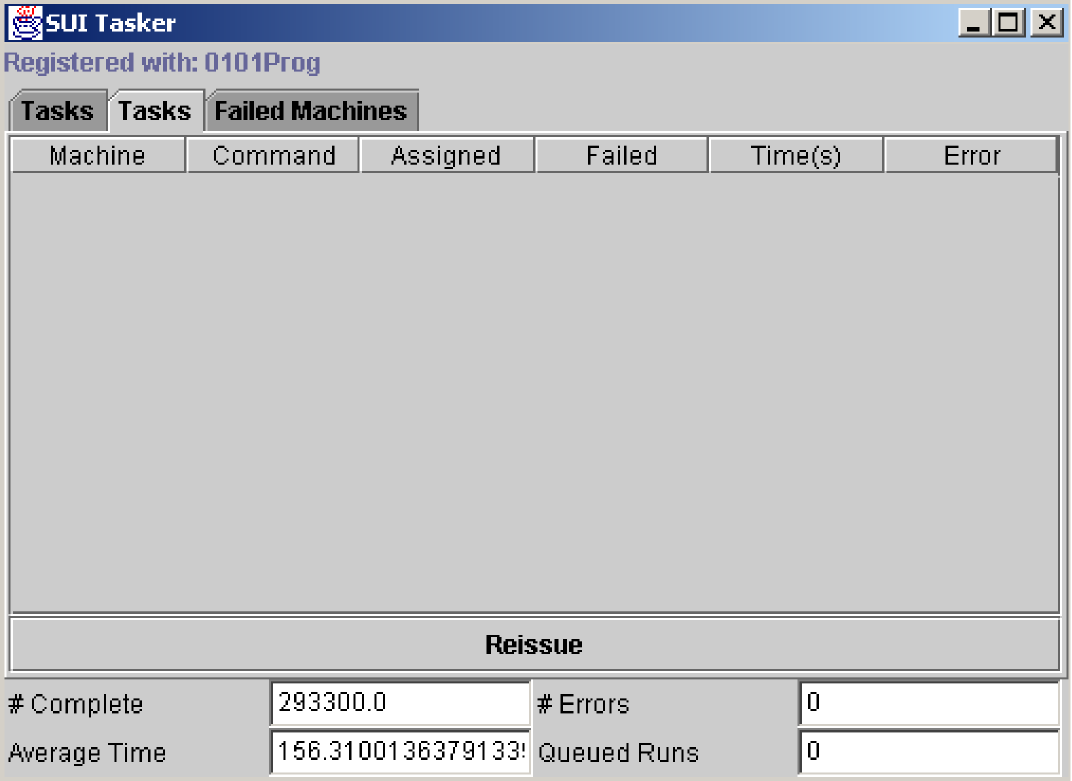
```
ERROR>The system cannot find the file specified.
ERROR>The system cannot find the file specified.
ERROR>The system cannot find the file specified.
ERROR>Could Not Find D:\hwir\src\%cache.dat
ERROR>The system cannot find the file specified.
ExitValue: 0
Done
Execing c:\winnt\system32\cmd.exe /C c:\hwir\RunAll.bat
OUTPUT>
OUTPUT>D:\hwir\src\HWIRMet>REM d:\hwir\Permanent\Test1\Y2002M2D8H13M10\SI\0131104
\1746-01-6\1\
OUTPUT>
OUTPUT>D:\hwir\src\HWIRMet>echo off
ERROR>The system cannot find the file specified.
ERROR>The system cannot find the file specified.
ERROR>The system cannot find the file specified.
ERROR>The system cannot find the file specified.
ERROR>The system cannot find the file specified.
ERROR>The system cannot find the file specified.
ERROR>The system cannot find the file specified.
ERROR>The system cannot find the file specified.
ERROR>The system cannot find the file specified.
ERROR>The system cannot find the file specified.
ERROR>The system cannot find the file specified.
```

Tasker Client (TC$_k$; k = 1…n$_c$)

Model independent.

Executes DOS commands in batch files delivered by a MT.

Figure 6-1a. SuperMUSE Parallel Management Software Toolset - Conceptual Layout for 3MRA.

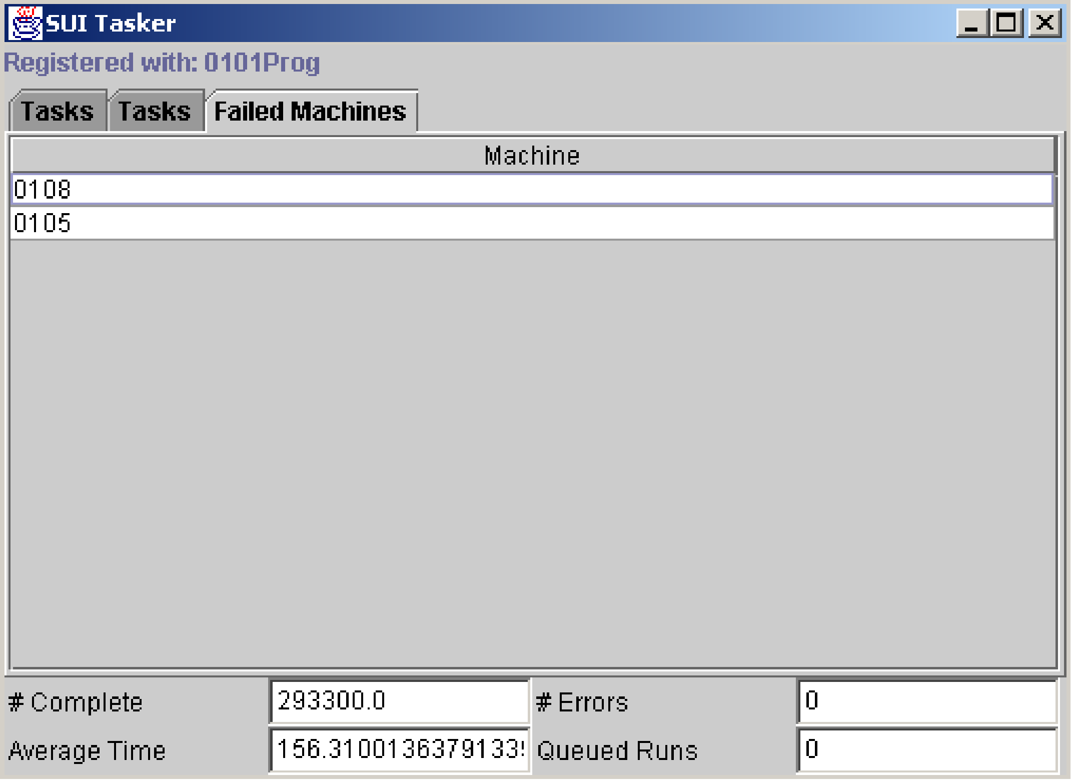


Registered with: 0101Prog

Machine	Command	Assigned	Failed	Time(s)	Error

# Complete	293300.0	# Errors	0
Average Time	156.31001363791331	Queued Runs	0

Figure 6-1b. SUITasker Failed Tasks Panel.

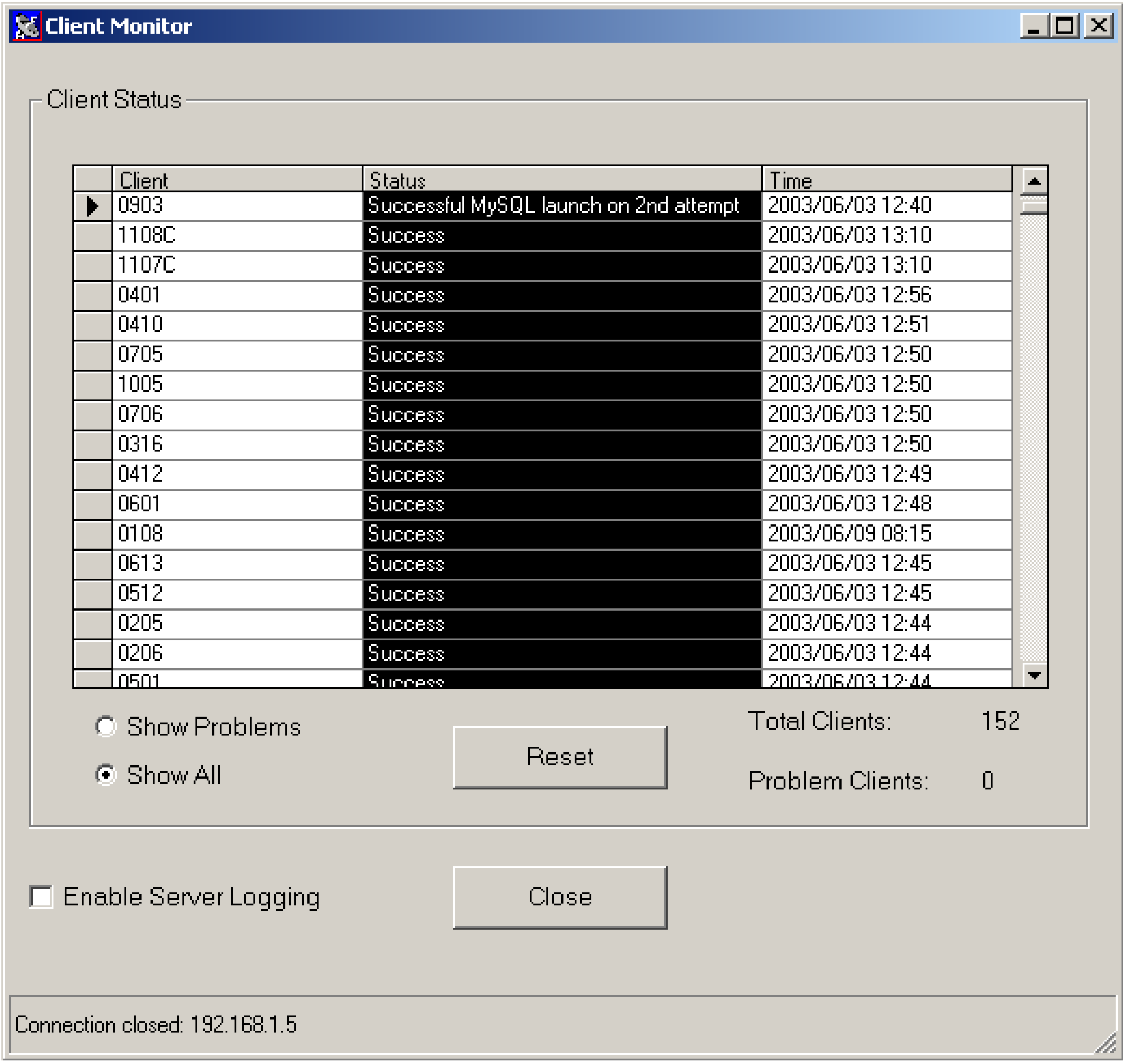


Registered with: 0101Prog

Machine
0108
0105

# Complete	293300.0	# Errors	0
Average Time	156.31001363791331	Queued Runs	0

Figure 6-1c. SUITasker Failed Machine Panel.



Client Status

Client	Status	Time
0903	Successful MySQL launch on 2nd attempt	2003/06/03 12:40
1108C	Success	2003/06/03 13:10
1107C	Success	2003/06/03 13:10
0401	Success	2003/06/03 12:56
0410	Success	2003/06/03 12:51
0705	Success	2003/06/03 12:50
1005	Success	2003/06/03 12:50
0706	Success	2003/06/03 12:50
0316	Success	2003/06/03 12:50
0412	Success	2003/06/03 12:49
0601	Success	2003/06/03 12:48
0108	Success	2003/06/09 08:15
0613	Success	2003/06/03 12:45
0512	Success	2003/06/03 12:45
0205	Success	2003/06/03 12:44
0206	Success	2003/06/03 12:44
0501	Success	2003/06/03 12:44

○ Show Problems

● Show All

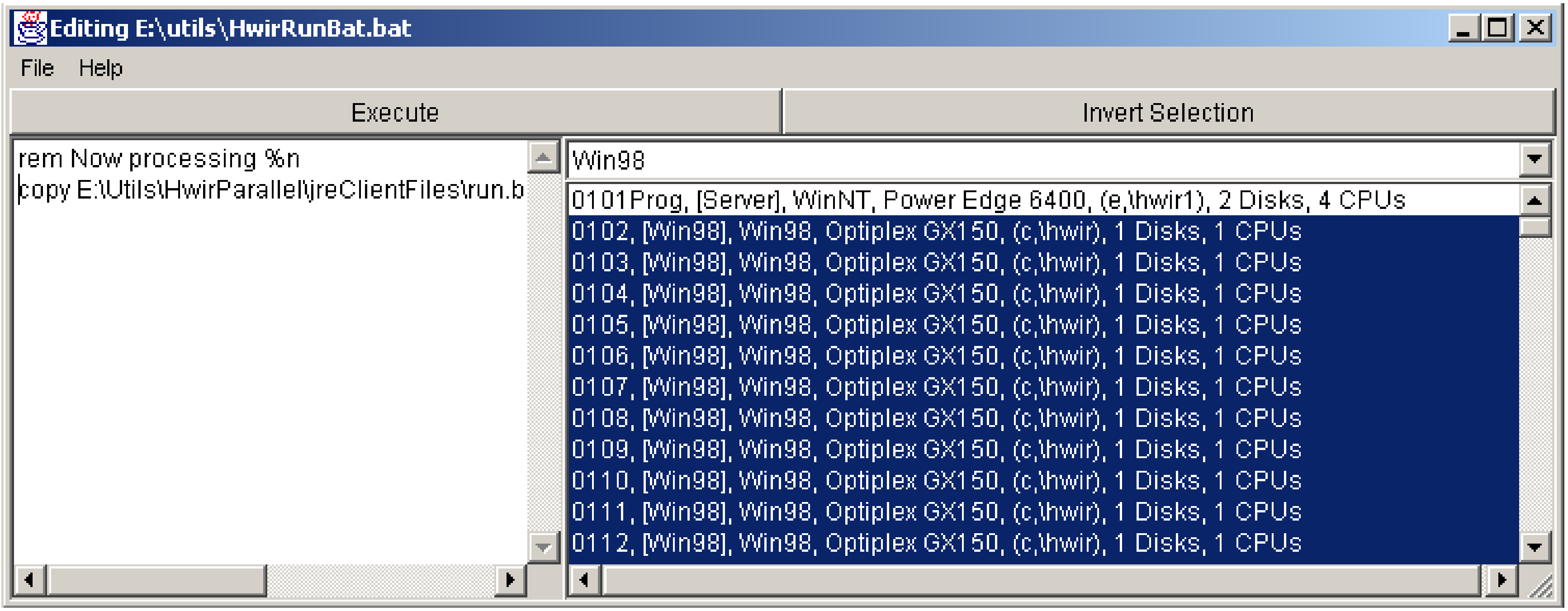
Total Clients: 152

Problem Clients: 0

□ Enable Server Logging

Connection closed: 192.168.1.5

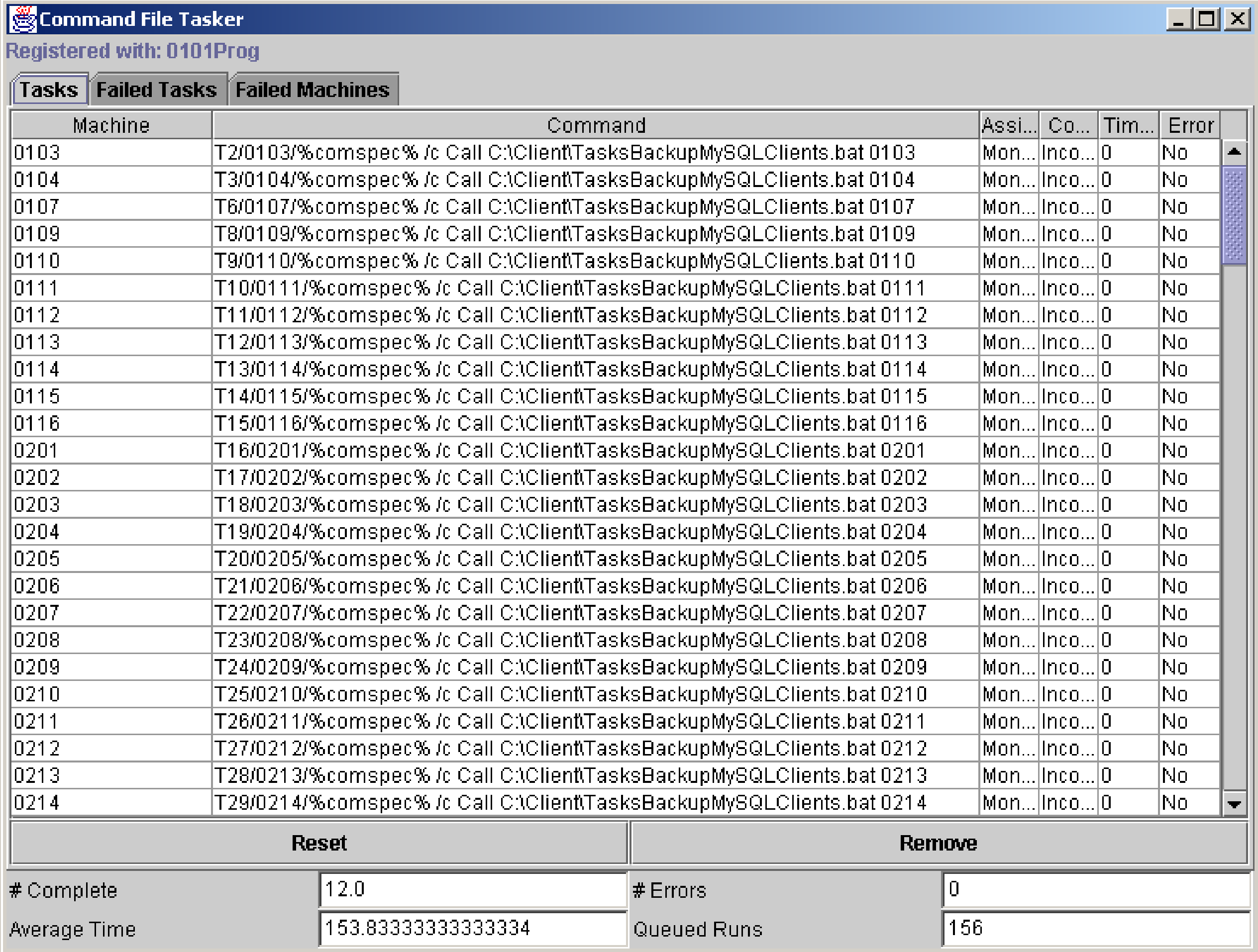
Figure 6-2. Server–Side Client Monitor Tool User Interface.



```
rem Now processing %n
copy E:\Utils\HwirParallel\jreClientFiles\run.b
```

```
0101Prog, [Server], WinNT, Power Edge 6400, (e,\hwir1), 2 Disks, 4 CPUs
0102, [Win98], Win98, Optiplex GX150, (c,\hwir), 1 Disks, 1 CPUs
0103, [Win98], Win98, Optiplex GX150, (c,\hwir), 1 Disks, 1 CPUs
0104, [Win98], Win98, Optiplex GX150, (c,\hwir), 1 Disks, 1 CPUs
0105, [Win98], Win98, Optiplex GX150, (c,\hwir), 1 Disks, 1 CPUs
0106, [Win98], Win98, Optiplex GX150, (c,\hwir), 1 Disks, 1 CPUs
0107, [Win98], Win98, Optiplex GX150, (c,\hwir), 1 Disks, 1 CPUs
0108, [Win98], Win98, Optiplex GX150, (c,\hwir), 1 Disks, 1 CPUs
0109, [Win98], Win98, Optiplex GX150, (c,\hwir), 1 Disks, 1 CPUs
0110, [Win98], Win98, Optiplex GX150, (c,\hwir), 1 Disks, 1 CPUs
0111, [Win98], Win98, Optiplex GX150, (c,\hwir), 1 Disks, 1 CPUs
0112, [Win98], Win98, Optiplex GX150, (c,\hwir), 1 Disks, 1 CPUs
```

Figure 6-3. Server–Side Update Client Tool User Interface.

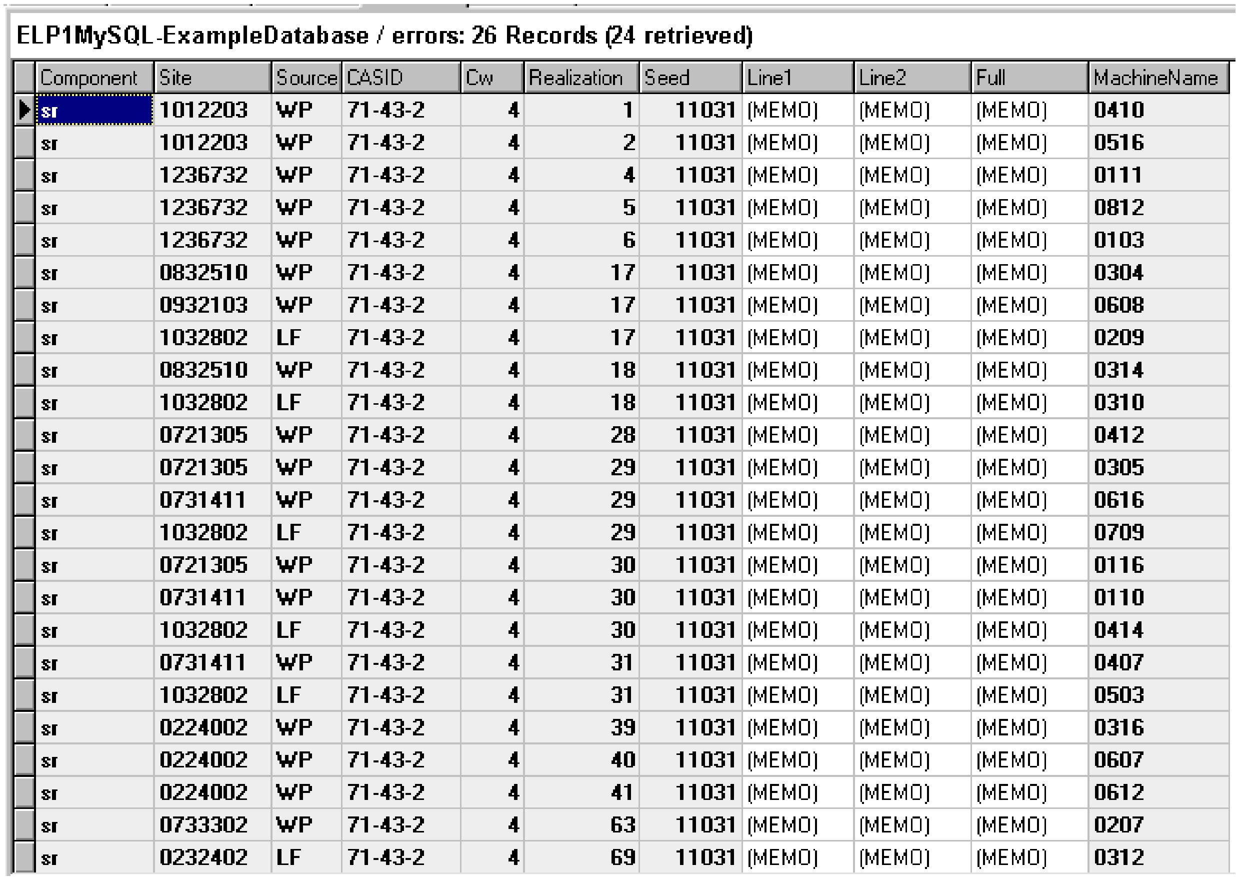


Registered with: 0101Prog

Machine	Command	Assi...	Co...	Tim...	Error
0103	T2/0103/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0103	Mon...	Inco...	0	No
0104	T3/0104/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0104	Mon...	Inco...	0	No
0107	T6/0107/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0107	Mon...	Inco...	0	No
0109	T8/0109/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0109	Mon...	Inco...	0	No
0110	T9/0110/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0110	Mon...	Inco...	0	No
0111	T10/0111/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0111	Mon...	Inco...	0	No
0112	T11/0112/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0112	Mon...	Inco...	0	No
0113	T12/0113/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0113	Mon...	Inco...	0	No
0114	T13/0114/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0114	Mon...	Inco...	0	No
0115	T14/0115/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0115	Mon...	Inco...	0	No
0116	T15/0116/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0116	Mon...	Inco...	0	No
0201	T16/0201/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0201	Mon...	Inco...	0	No
0202	T17/0202/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0202	Mon...	Inco...	0	No
0203	T18/0203/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0203	Mon...	Inco...	0	No
0204	T19/0204/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0204	Mon...	Inco...	0	No
0205	T20/0205/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0205	Mon...	Inco...	0	No
0206	T21/0206/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0206	Mon...	Inco...	0	No
0207	T22/0207/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0207	Mon...	Inco...	0	No
0208	T23/0208/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0208	Mon...	Inco...	0	No
0209	T24/0209/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0209	Mon...	Inco...	0	No
0210	T25/0210/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0210	Mon...	Inco...	0	No
0211	T26/0211/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0211	Mon...	Inco...	0	No
0212	T27/0212/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0212	Mon...	Inco...	0	No
0213	T28/0213/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0213	Mon...	Inco...	0	No
0214	T29/0214/%comspec% /c Call C:\Client\TasksBackupMySQLClients.bat 0214	Mon...	Inco...	0	No

# Complete	12.0	# Errors	0
Average Time	153.83333333333334	Queued Runs	156

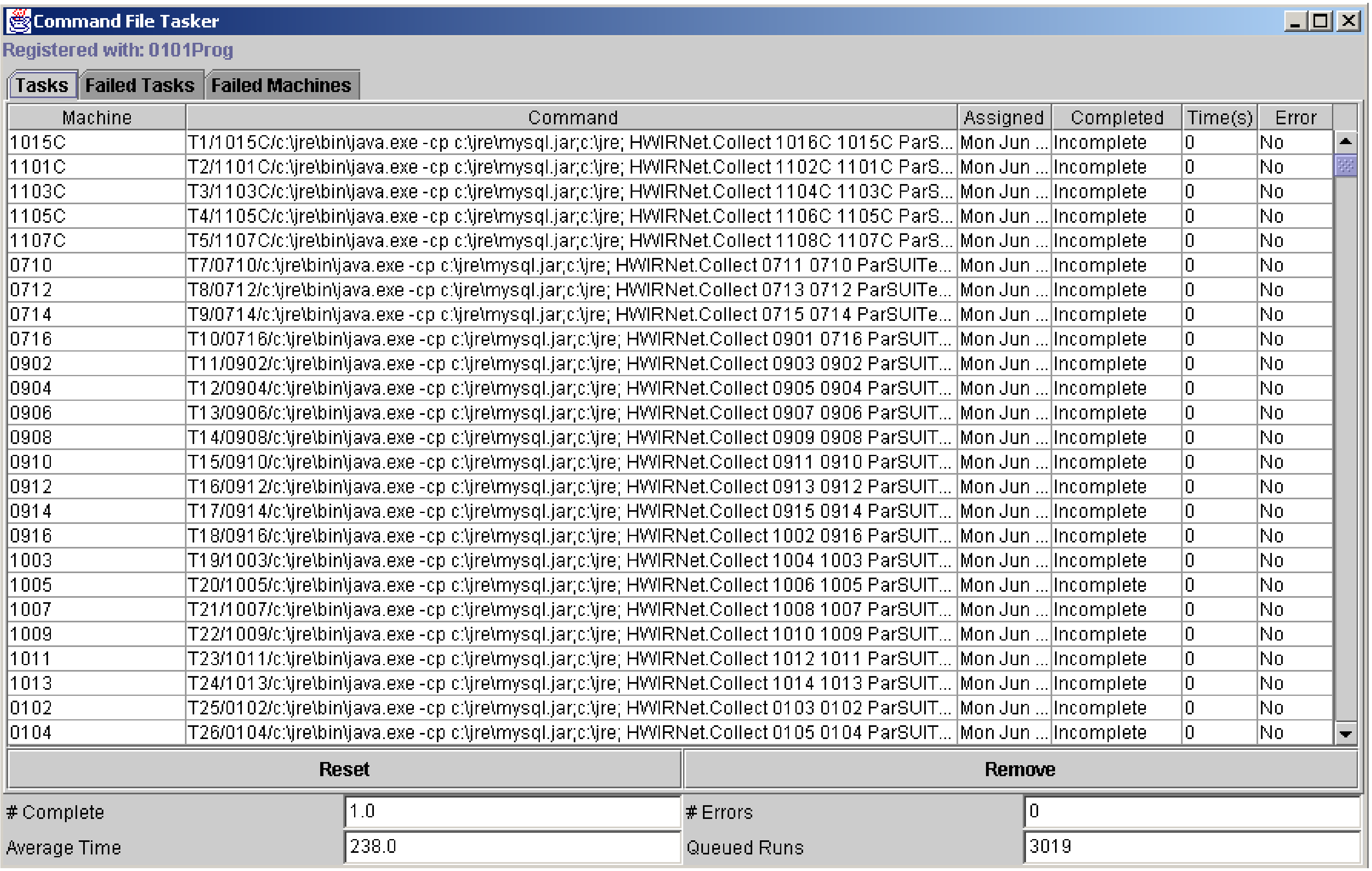
Figure 6-4. Server–Side Command Tasker Tool User Interface.
(Showing execution of a MySQL database backup routine implemented across SuperMUSE)



ELP1MySQL-ExampleDatabase / errors: 26 Records (24 retrieved)

Component	Site	Source	CASID	Cw	Realization	Seed	Line1	Line2	Full	MachineName
sr	1012203	WP	71-43-2	4	1	11031	(MEMO)	(MEMO)	(MEMO)	0410
sr	1012203	WP	71-43-2	4	2	11031	(MEMO)	(MEMO)	(MEMO)	0516
sr	1236732	WP	71-43-2	4	4	11031	(MEMO)	(MEMO)	(MEMO)	0111
sr	1236732	WP	71-43-2	4	5	11031	(MEMO)	(MEMO)	(MEMO)	0812
sr	1236732	WP	71-43-2	4	6	11031	(MEMO)	(MEMO)	(MEMO)	0103
sr	0832510	WP	71-43-2	4	17	11031	(MEMO)	(MEMO)	(MEMO)	0304
sr	0932103	WP	71-43-2	4	17	11031	(MEMO)	(MEMO)	(MEMO)	0608
sr	1032802	LF	71-43-2	4	17	11031	(MEMO)	(MEMO)	(MEMO)	0209
sr	0832510	WP	71-43-2	4	18	11031	(MEMO)	(MEMO)	(MEMO)	0314
sr	1032802	LF	71-43-2	4	18	11031	(MEMO)	(MEMO)	(MEMO)	0310
sr	0721305	WP	71-43-2	4	28	11031	(MEMO)	(MEMO)	(MEMO)	0412
sr	0721305	WP	71-43-2	4	29	11031	(MEMO)	(MEMO)	(MEMO)	0305
sr	0731411	WP	71-43-2	4	29	11031	(MEMO)	(MEMO)	(MEMO)	0616
sr	1032802	LF	71-43-2	4	29	11031	(MEMO)	(MEMO)	(MEMO)	0709
sr	0721305	WP	71-43-2	4	30	11031	(MEMO)	(MEMO)	(MEMO)	0116
sr	0731411	WP	71-43-2	4	30	11031	(MEMO)	(MEMO)	(MEMO)	0110
sr	1032802	LF	71-43-2	4	30	11031	(MEMO)	(MEMO)	(MEMO)	0414
sr	0731411	WP	71-43-2	4	31	11031	(MEMO)	(MEMO)	(MEMO)	0407
sr	1032802	LF	71-43-2	4	31	11031	(MEMO)	(MEMO)	(MEMO)	0503
sr	0224002	WP	71-43-2	4	39	11031	(MEMO)	(MEMO)	(MEMO)	0316
sr	0224002	WP	71-43-2	4	40	11031	(MEMO)	(MEMO)	(MEMO)	0607
sr	0224002	WP	71-43-2	4	41	11031	(MEMO)	(MEMO)	(MEMO)	0612
sr	0733302	WP	71-43-2	4	63	11031	(MEMO)	(MEMO)	(MEMO)	0207
sr	0232402	LF	71-43-2	4	69	11031	(MEMO)	(MEMO)	(MEMO)	0312

Figure 6-5. Example "Error" Table Created by the Process Messages Tool.



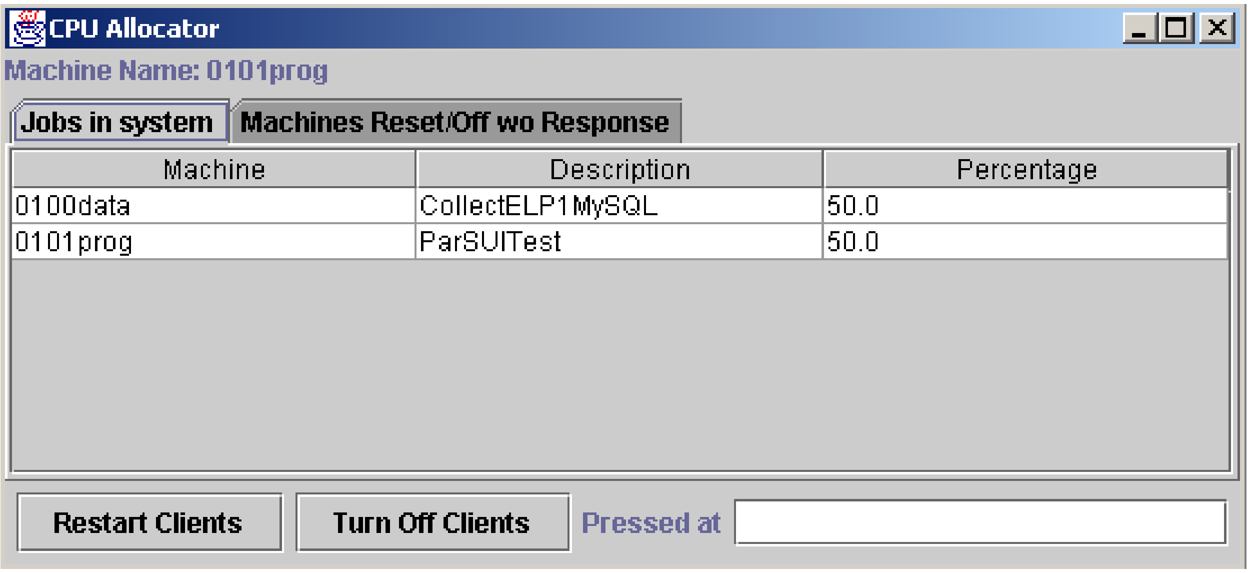
Command File Tasker

Registered with: 0101Prog

Machine	Command	Assigned	Completed	Time(s)	Error
1015C	T1/1015C/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 1016C 1015C ParS...	Mon Jun ...	Incomplete	0	No
1101C	T2/1101C/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 1102C 1101C ParS...	Mon Jun ...	Incomplete	0	No
1103C	T3/1103C/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 1104C 1103C ParS...	Mon Jun ...	Incomplete	0	No
1105C	T4/1105C/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 1106C 1105C ParS...	Mon Jun ...	Incomplete	0	No
1107C	T5/1107C/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 1108C 1107C ParS...	Mon Jun ...	Incomplete	0	No
0710	T7/0710/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0711 0710 ParSUITe...	Mon Jun ...	Incomplete	0	No
0712	T8/0712/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0713 0712 ParSUITe...	Mon Jun ...	Incomplete	0	No
0714	T9/0714/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0715 0714 ParSUITe...	Mon Jun ...	Incomplete	0	No
0716	T10/0716/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0901 0716 ParSUIT...	Mon Jun ...	Incomplete	0	No
0902	T11/0902/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0903 0902 ParSUIT...	Mon Jun ...	Incomplete	0	No
0904	T12/0904/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0905 0904 ParSUIT...	Mon Jun ...	Incomplete	0	No
0906	T13/0906/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0907 0906 ParSUIT...	Mon Jun ...	Incomplete	0	No
0908	T14/0908/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0909 0908 ParSUIT...	Mon Jun ...	Incomplete	0	No
0910	T15/0910/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0911 0910 ParSUIT...	Mon Jun ...	Incomplete	0	No
0912	T16/0912/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0913 0912 ParSUIT...	Mon Jun ...	Incomplete	0	No
0914	T17/0914/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0915 0914 ParSUIT...	Mon Jun ...	Incomplete	0	No
0916	T18/0916/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 1002 0916 ParSUIT...	Mon Jun ...	Incomplete	0	No
1003	T19/1003/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 1004 1003 ParSUIT...	Mon Jun ...	Incomplete	0	No
1005	T20/1005/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 1006 1005 ParSUIT...	Mon Jun ...	Incomplete	0	No
1007	T21/1007/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 1008 1007 ParSUIT...	Mon Jun ...	Incomplete	0	No
1009	T22/1009/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 1010 1009 ParSUIT...	Mon Jun ...	Incomplete	0	No
1011	T23/1011/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 1012 1011 ParSUIT...	Mon Jun ...	Incomplete	0	No
1013	T24/1013/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 1014 1013 ParSUIT...	Mon Jun ...	Incomplete	0	No
0102	T25/0102/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0103 0102 ParSUIT...	Mon Jun ...	Incomplete	0	No
0104	T26/0104/c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0105 0104 ParSUIT...	Mon Jun ...	Incomplete	0	No

# Complete	1.0	# Errors	0
Average Time	238.0	Queued Runs	3019

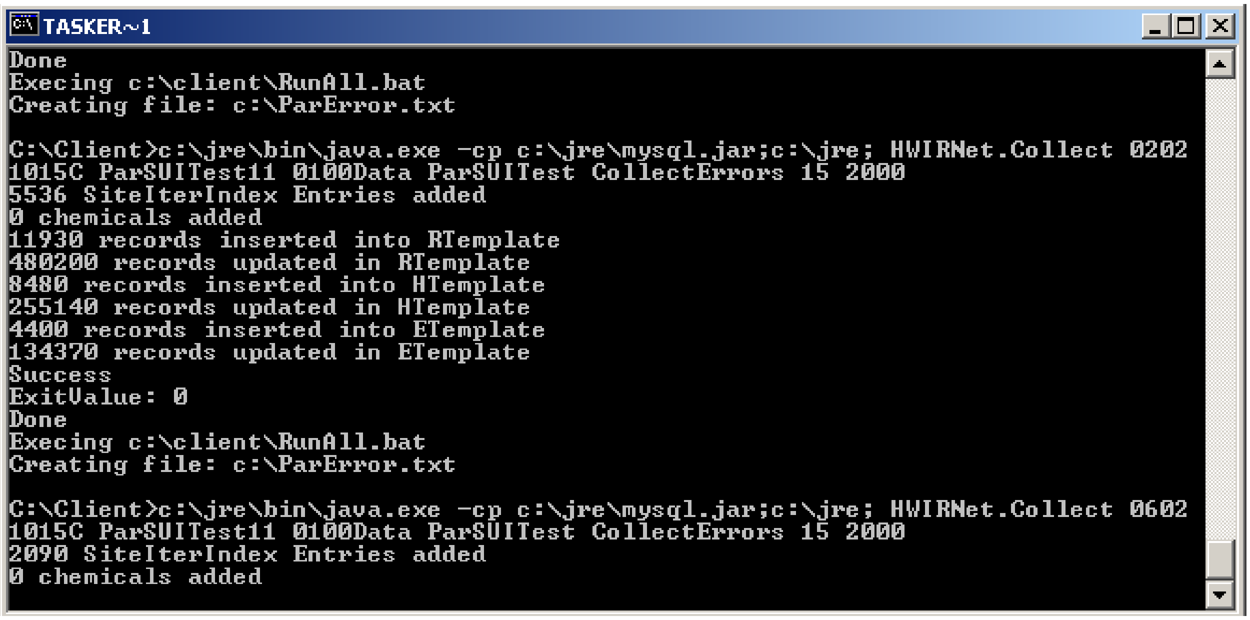
Figure 6-6a. Command Tasker Implementation of Aggregated ELP1 Client Collector Tool.



Machine Name: 0101prog

Machine	Description	Percentage
0100data	CollectELP1MySQL	50.0
0101prog	ParSUITest	50.0

Figure 6-6b. CPU Allocator With Active Command Tasker and SUITasker

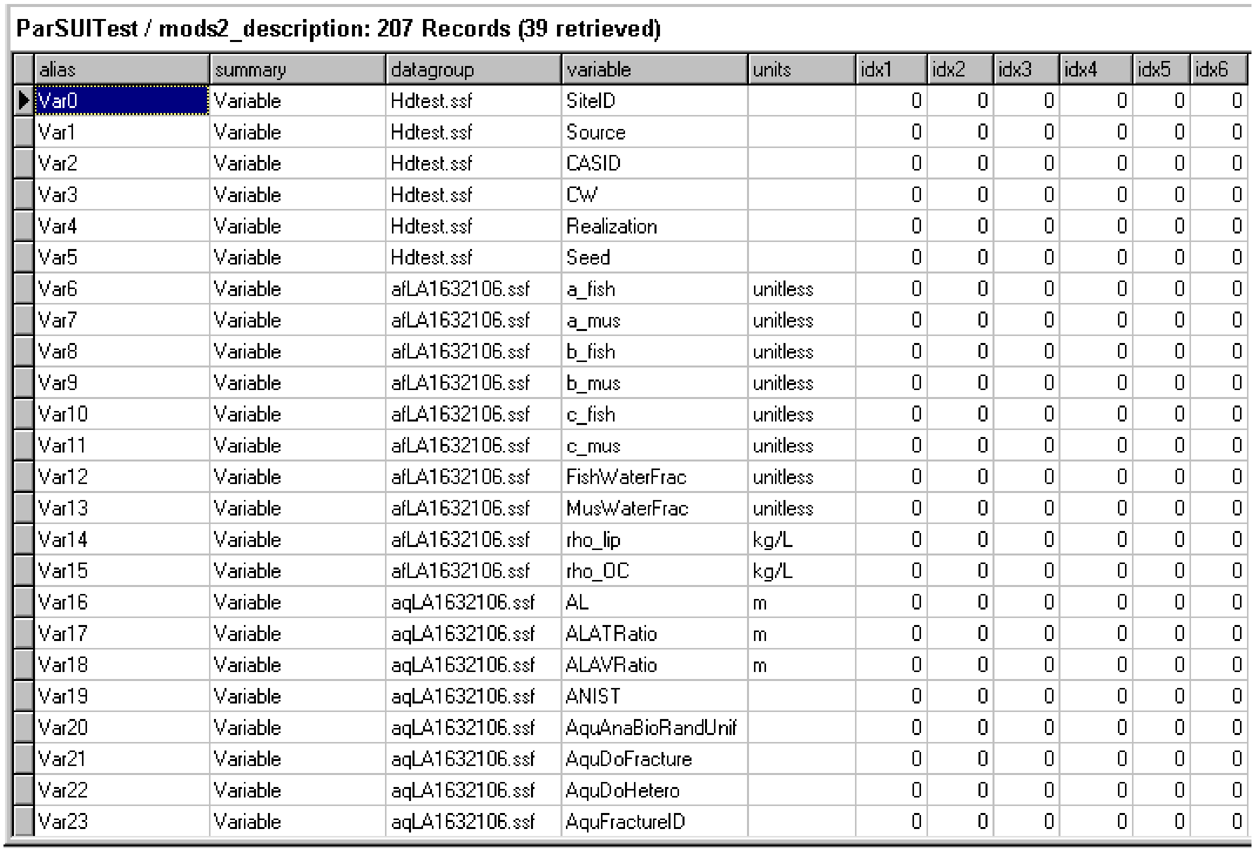


```
Done
Execing c:\client\RunAll.bat
Creating file: c:\ParError.txt

C:\Client>c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0202
1015C ParSUITest11 0100Data ParSUITest CollectErrors 15 2000
5536 SiteIterIndex Entries added
0 chemicals added
11930 records inserted into RTemplate
480200 records updated in RTemplate
8480 records inserted into HTemplate
255140 records updated in HTemplate
4400 records inserted into ETemplate
134370 records updated in ETemplate
Success
ExitValue: 0
Done
Execing c:\client\RunAll.bat
Creating file: c:\ParError.txt

C:\Client>c:\jre\bin\java.exe -cp c:\jre\mysql.jar;c:\jre; HWIRNet.Collect 0602
1015C ParSUITest11 0100Data ParSUITest CollectErrors 15 2000
2090 SiteIterIndex Entries added
0 chemicals added
```

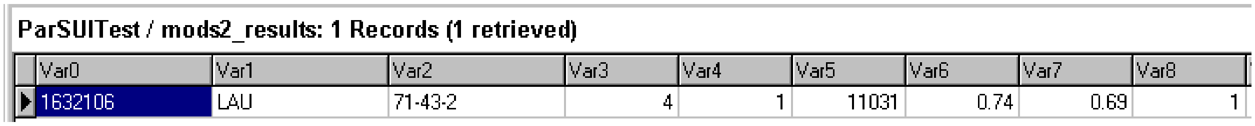
Figure 6-6c. Example Client Collector Task Implementation on Tasker Client Side.
(PC Client MachineID = 1015C Collecting ELP1MySQL records from PC Client 0602)



ParSUITest / mods2_description: 207 Records (39 retrieved)

alias	summary	datagroup	variable	units	idx1	idx2	idx3	idx4	idx5	idx6
Var0	Variable	Hdtest.ssf	SiteID		0	0	0	0	0	0
Var1	Variable	Hdtest.ssf	Source		0	0	0	0	0	0
Var2	Variable	Hdtest.ssf	CASID		0	0	0	0	0	0
Var3	Variable	Hdtest.ssf	CW		0	0	0	0	0	0
Var4	Variable	Hdtest.ssf	Realization		0	0	0	0	0	0
Var5	Variable	Hdtest.ssf	Seed		0	0	0	0	0	0
Var6	Variable	afLA1632106.ssf	a_fish	unitless	0	0	0	0	0	0
Var7	Variable	afLA1632106.ssf	a_mus	unitless	0	0	0	0	0	0
Var8	Variable	afLA1632106.ssf	b_fish	unitless	0	0	0	0	0	0
Var9	Variable	afLA1632106.ssf	b_mus	unitless	0	0	0	0	0	0
Var10	Variable	afLA1632106.ssf	c_fish	unitless	0	0	0	0	0	0
Var11	Variable	afLA1632106.ssf	c_mus	unitless	0	0	0	0	0	0
Var12	Variable	afLA1632106.ssf	FishWaterFrac	unitless	0	0	0	0	0	0
Var13	Variable	afLA1632106.ssf	MusWaterFrac	unitless	0	0	0	0	0	0
Var14	Variable	afLA1632106.ssf	rho_lip	kg/L	0	0	0	0	0	0
Var15	Variable	afLA1632106.ssf	rho_OC	kg/L	0	0	0	0	0	0
Var16	Variable	aqLA1632106.ssf	AL	m	0	0	0	0	0	0
Var17	Variable	aqLA1632106.ssf	ALATRatio	m	0	0	0	0	0	0
Var18	Variable	aqLA1632106.ssf	ALAVRatio	m	0	0	0	0	0	0
Var19	Variable	aqLA1632106.ssf	ANIST		0	0	0	0	0	0
Var20	Variable	aqLA1632106.ssf	AquAnaBioRandUnif		0	0	0	0	0	0
Var21	Variable	aqLA1632106.ssf	AquDoFracture		0	0	0	0	0	0
Var22	Variable	aqLA1632106.ssf	AquDoHetero		0	0	0	0	0	0
Var23	Variable	aqLA1632106.ssf	AquFractureID		0	0	0	0	0	0

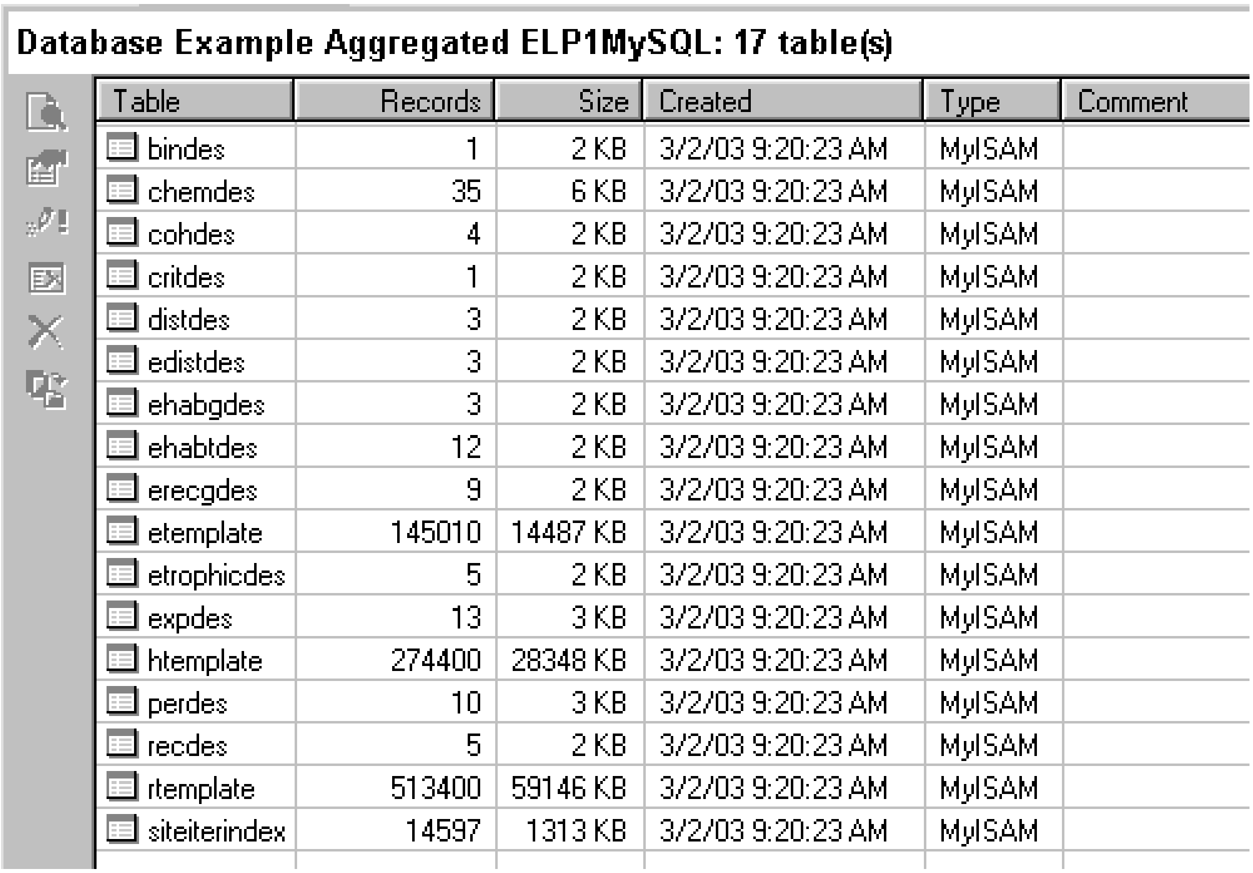
Figure 6-7a. Example "Description" Table Created by the Site Summary Tool.



ParSUITest / mods2_results: 1 Records (1 retrieved)

Var0	Var1	Var2	Var3	Var4	Var5	Var6	Var7	Var8
1632106	LAU	71-43-2	4	1	11031	0.74	0.69	1

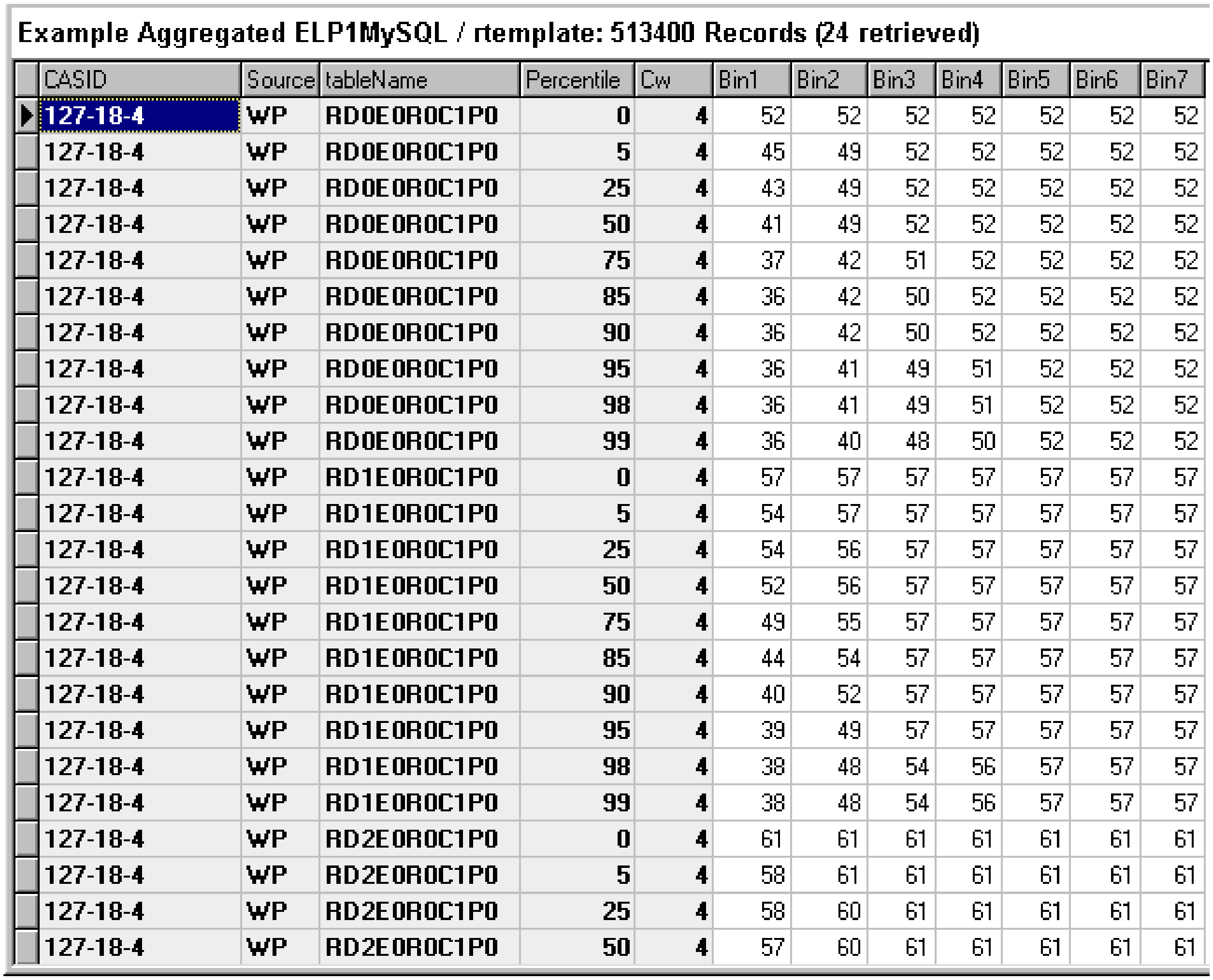
Figure 6-7b. Example "Result" Table Created by the Site Summary Tool.



Database Example Aggregated ELP1MySQL: 17 table(s)

Table	Records	Size	Created	Type	Comment
bindes	1	2 KB	3/2/03 9:20:23 AM	MyISAM	
chemdes	35	6 KB	3/2/03 9:20:23 AM	MyISAM	
cohdes	4	2 KB	3/2/03 9:20:23 AM	MyISAM	
critdes	1	2 KB	3/2/03 9:20:23 AM	MyISAM	
distdes	3	2 KB	3/2/03 9:20:23 AM	MyISAM	
edistdes	3	2 KB	3/2/03 9:20:23 AM	MyISAM	
ehabgdes	3	2 KB	3/2/03 9:20:23 AM	MyISAM	
ehabtdes	12	2 KB	3/2/03 9:20:23 AM	MyISAM	
erecgdes	9	2 KB	3/2/03 9:20:23 AM	MyISAM	
etemplate	145010	14487 KB	3/2/03 9:20:23 AM	MyISAM	
etrophicdes	5	2 KB	3/2/03 9:20:23 AM	MyISAM	
expdes	13	3 KB	3/2/03 9:20:23 AM	MyISAM	
htemplate	274400	28348 KB	3/2/03 9:20:23 AM	MyISAM	
perdes	10	3 KB	3/2/03 9:20:23 AM	MyISAM	
recdes	5	2 KB	3/2/03 9:20:23 AM	MyISAM	
rtemplate	513400	59146 KB	3/2/03 9:20:23 AM	MyISAM	
siteiterindex	14597	1313 KB	3/2/03 9:20:23 AM	MyISAM	

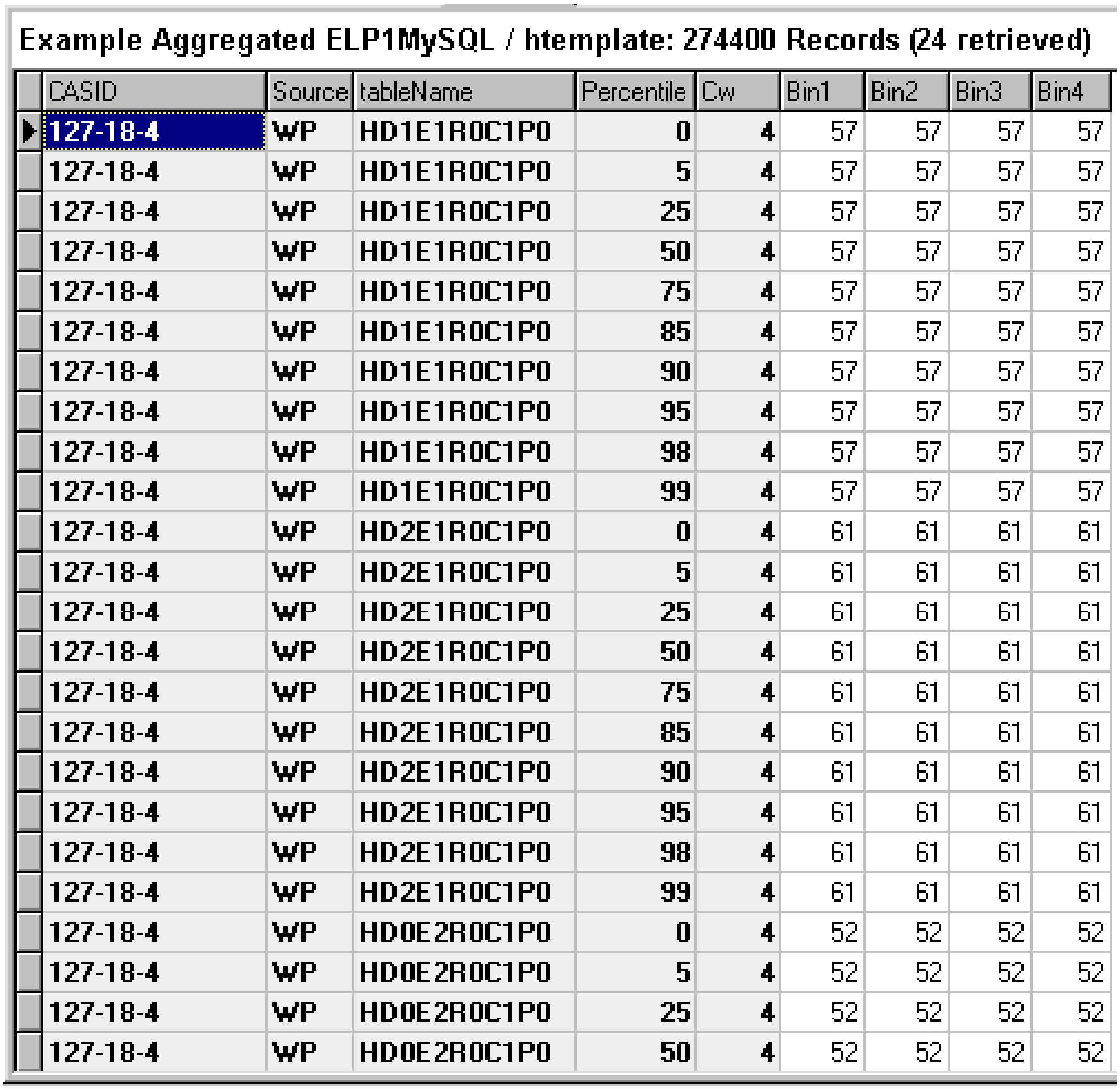
Figure 6-8a. Example Table Summary For Aggregated ELP1 MySQL Database.



Example Aggregated ELP1MySQL / rtemplate: 513400 Records (24 retrieved)

CASID	Source	tableName	Percentile	Cw	Bin1	Bin2	Bin3	Bin4	Bin5	Bin6	Bin7
127-18-4	WP	RD0E0R0C1P0	0	4	52	52	52	52	52	52	52
127-18-4	WP	RD0E0R0C1P0	5	4	45	49	52	52	52	52	52
127-18-4	WP	RD0E0R0C1P0	25	4	43	49	52	52	52	52	52
127-18-4	WP	RD0E0R0C1P0	50	4	41	49	52	52	52	52	52
127-18-4	WP	RD0E0R0C1P0	75	4	37	42	51	52	52	52	52
127-18-4	WP	RD0E0R0C1P0	85	4	36	42	50	52	52	52	52
127-18-4	WP	RD0E0R0C1P0	90	4	36	42	50	52	52	52	52
127-18-4	WP	RD0E0R0C1P0	95	4	36	41	49	51	52	52	52
127-18-4	WP	RD0E0R0C1P0	98	4	36	41	49	51	52	52	52
127-18-4	WP	RD0E0R0C1P0	99	4	36	40	48	50	52	52	52
127-18-4	WP	RD1E0R0C1P0	0	4	57	57	57	57	57	57	57
127-18-4	WP	RD1E0R0C1P0	5	4	54	57	57	57	57	57	57
127-18-4	WP	RD1E0R0C1P0	25	4	54	56	57	57	57	57	57
127-18-4	WP	RD1E0R0C1P0	50	4	52	56	57	57	57	57	57
127-18-4	WP	RD1E0R0C1P0	75	4	49	55	57	57	57	57	57
127-18-4	WP	RD1E0R0C1P0	85	4	44	54	57	57	57	57	57
127-18-4	WP	RD1E0R0C1P0	90	4	40	52	57	57	57	57	57
127-18-4	WP	RD1E0R0C1P0	95	4	39	49	57	57	57	57	57
127-18-4	WP	RD1E0R0C1P0	98	4	38	48	54	56	57	57	57
127-18-4	WP	RD1E0R0C1P0	99	4	38	48	54	56	57	57	57
127-18-4	WP	RD2E0R0C1P0	0	4	61	61	61	61	61	61	61
127-18-4	WP	RD2E0R0C1P0	5	4	58	61	61	61	61	61	61
127-18-4	WP	RD2E0R0C1P0	25	4	58	60	61	61	61	61	61
127-18-4	WP	RD2E0R0C1P0	50	4	57	60	61	61	61	61	61

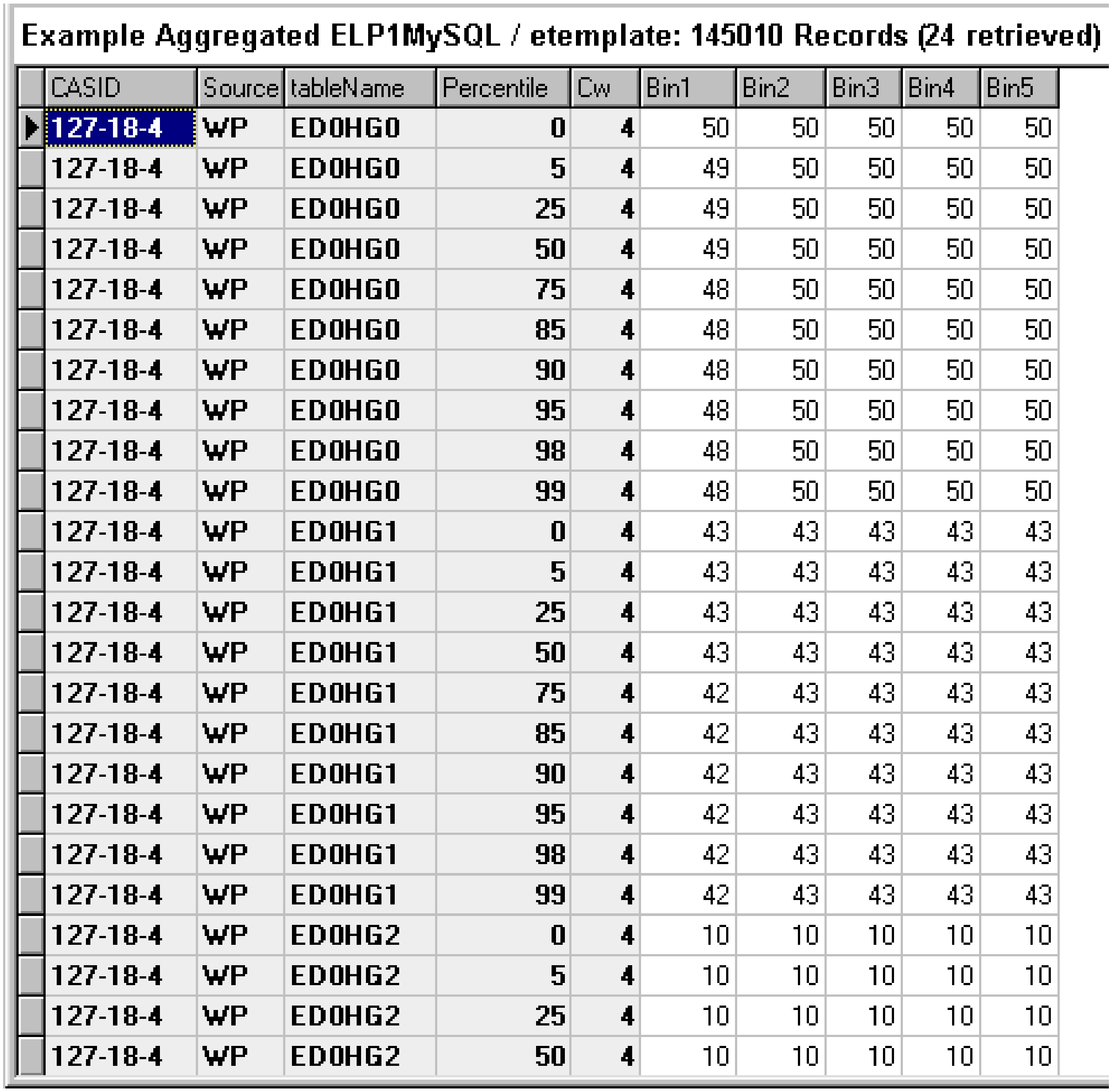
Figure 6-8b. Example RTemplate Table in Aggregated ELP1 MySQL Database.
(Human Cancer Risk Summary)



Example Aggregated ELP1MySQL / htemplate: 274400 Records (24 retrieved)

CASID	Source	tableName	Percentile	Cw	Bin1	Bin2	Bin3	Bin4
127-18-4	WP	HD1E1R0C1P0	0	4	57	57	57	57
127-18-4	WP	HD1E1R0C1P0	5	4	57	57	57	57
127-18-4	WP	HD1E1R0C1P0	25	4	57	57	57	57
127-18-4	WP	HD1E1R0C1P0	50	4	57	57	57	57
127-18-4	WP	HD1E1R0C1P0	75	4	57	57	57	57
127-18-4	WP	HD1E1R0C1P0	85	4	57	57	57	57
127-18-4	WP	HD1E1R0C1P0	90	4	57	57	57	57
127-18-4	WP	HD1E1R0C1P0	95	4	57	57	57	57
127-18-4	WP	HD1E1R0C1P0	98	4	57	57	57	57
127-18-4	WP	HD1E1R0C1P0	99	4	57	57	57	57
127-18-4	WP	HD2E1R0C1P0	0	4	61	61	61	61
127-18-4	WP	HD2E1R0C1P0	5	4	61	61	61	61
127-18-4	WP	HD2E1R0C1P0	25	4	61	61	61	61
127-18-4	WP	HD2E1R0C1P0	50	4	61	61	61	61
127-18-4	WP	HD2E1R0C1P0	75	4	61	61	61	61
127-18-4	WP	HD2E1R0C1P0	85	4	61	61	61	61
127-18-4	WP	HD2E1R0C1P0	90	4	61	61	61	61
127-18-4	WP	HD2E1R0C1P0	95	4	61	61	61	61
127-18-4	WP	HD2E1R0C1P0	98	4	61	61	61	61
127-18-4	WP	HD2E1R0C1P0	99	4	61	61	61	61
127-18-4	WP	HD0E2R0C1P0	0	4	52	52	52	52
127-18-4	WP	HD0E2R0C1P0	5	4	52	52	52	52
127-18-4	WP	HD0E2R0C1P0	25	4	52	52	52	52
127-18-4	WP	HD0E2R0C1P0	50	4	52	52	52	52

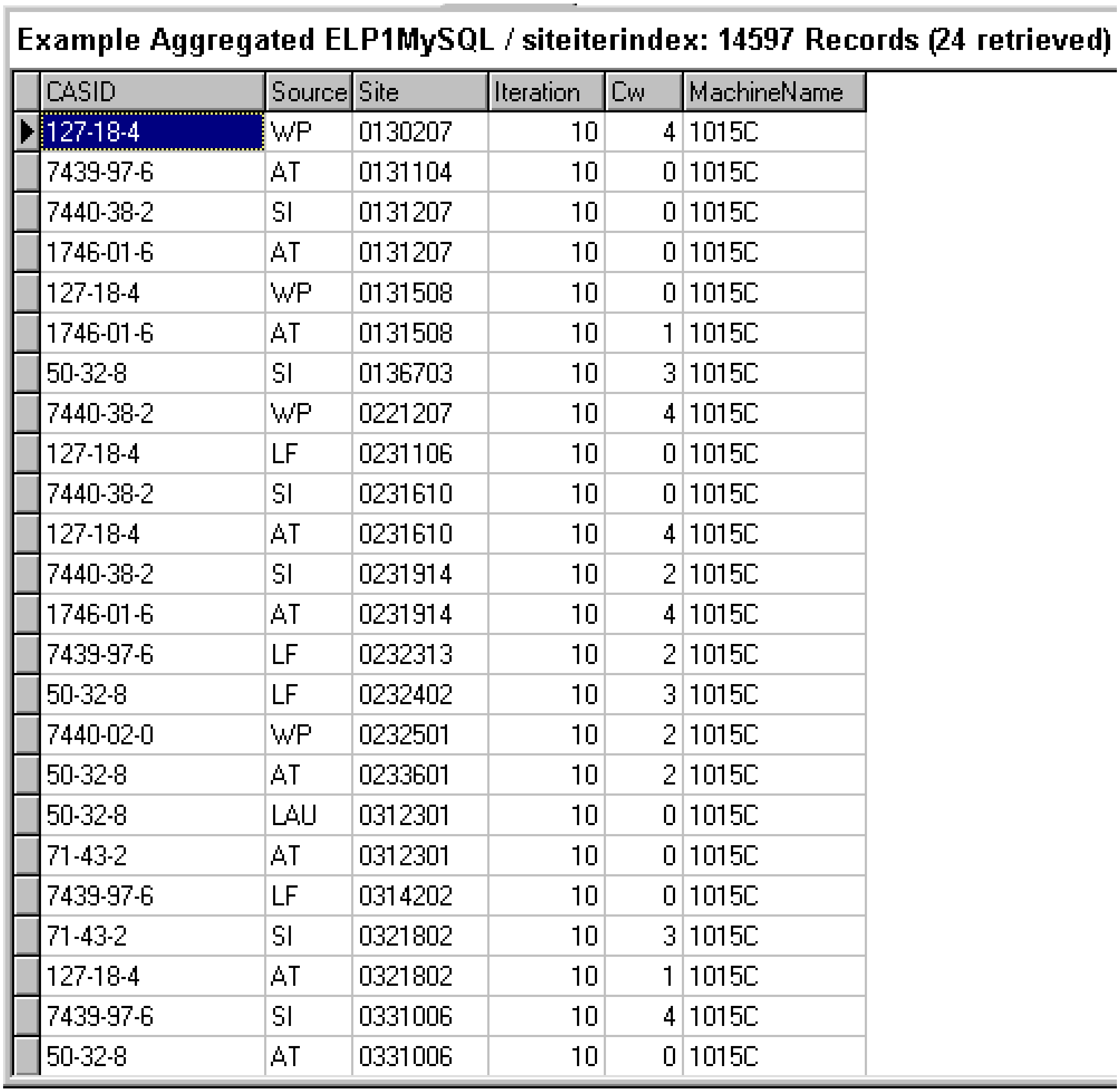
Figure 6-8c. Example HTemplate Table in Aggregated ELP1 MySQL Database.
(Human Hazard Risk Summary)



Example Aggregated ELP1MySQL / etemplate: 145010 Records (24 retrieved)

CASID	Source	tableName	Percentile	Cw	Bin1	Bin2	Bin3	Bin4	Bin5
127-18-4	WP	ED0HG0	0	4	50	50	50	50	50
127-18-4	WP	ED0HG0	5	4	49	50	50	50	50
127-18-4	WP	ED0HG0	25	4	49	50	50	50	50
127-18-4	WP	ED0HG0	50	4	49	50	50	50	50
127-18-4	WP	ED0HG0	75	4	48	50	50	50	50
127-18-4	WP	ED0HG0	85	4	48	50	50	50	50
127-18-4	WP	ED0HG0	90	4	48	50	50	50	50
127-18-4	WP	ED0HG0	95	4	48	50	50	50	50
127-18-4	WP	ED0HG0	98	4	48	50	50	50	50
127-18-4	WP	ED0HG0	99	4	48	50	50	50	50
127-18-4	WP	ED0HG1	0	4	43	43	43	43	43
127-18-4	WP	ED0HG1	5	4	43	43	43	43	43
127-18-4	WP	ED0HG1	25	4	43	43	43	43	43
127-18-4	WP	ED0HG1	50	4	43	43	43	43	43
127-18-4	WP	ED0HG1	75	4	42	43	43	43	43
127-18-4	WP	ED0HG1	85	4	42	43	43	43	43
127-18-4	WP	ED0HG1	90	4	42	43	43	43	43
127-18-4	WP	ED0HG1	95	4	42	43	43	43	43
127-18-4	WP	ED0HG1	98	4	42	43	43	43	43
127-18-4	WP	ED0HG1	99	4	42	43	43	43	43
127-18-4	WP	ED0HG2	0	4	10	10	10	10	10
127-18-4	WP	ED0HG2	5	4	10	10	10	10	10
127-18-4	WP	ED0HG2	25	4	10	10	10	10	10
127-18-4	WP	ED0HG2	50	4	10	10	10	10	10

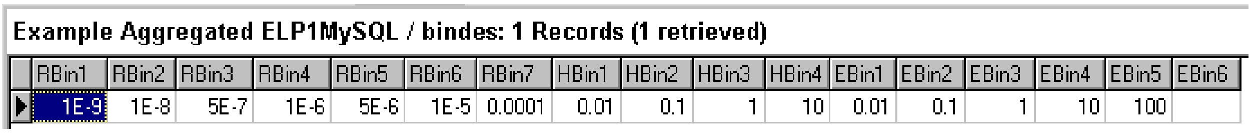
Figure 6-8d. Example ETemplate Table in Aggregated ELP1 MySQL Database.
(Ecological Hazard Risk Summary)



Example Aggregated ELP1MySQL / siteiterindex: 14597 Records (24 retrieved)

CASID	Source	Site	Iteration	Cw	MachineName
127-18-4	WP	0130207	10	4	1015C
7439-97-6	AT	0131104	10	0	1015C
7440-38-2	SI	0131207	10	0	1015C
1746-01-6	AT	0131207	10	0	1015C
127-18-4	WP	0131508	10	0	1015C
1746-01-6	AT	0131508	10	1	1015C
50-32-8	SI	0136703	10	3	1015C
7440-38-2	WP	0221207	10	4	1015C
127-18-4	LF	0231106	10	0	1015C
7440-38-2	SI	0231610	10	0	1015C
127-18-4	AT	0231610	10	4	1015C
7440-38-2	SI	0231914	10	2	1015C
1746-01-6	AT	0231914	10	4	1015C
7439-97-6	LF	0232313	10	2	1015C
50-32-8	LF	0232402	10	3	1015C
7440-02-0	WP	0232501	10	2	1015C
50-32-8	AT	0233601	10	2	1015C
50-32-8	LAU	0312301	10	0	1015C
71-43-2	AT	0312301	10	0	1015C
7439-97-6	LF	0314202	10	0	1015C
71-43-2	SI	0321802	10	3	1015C
127-18-4	AT	0321802	10	1	1015C
7439-97-6	SI	0331006	10	4	1015C
50-32-8	AT	0331006	10	0	1015C

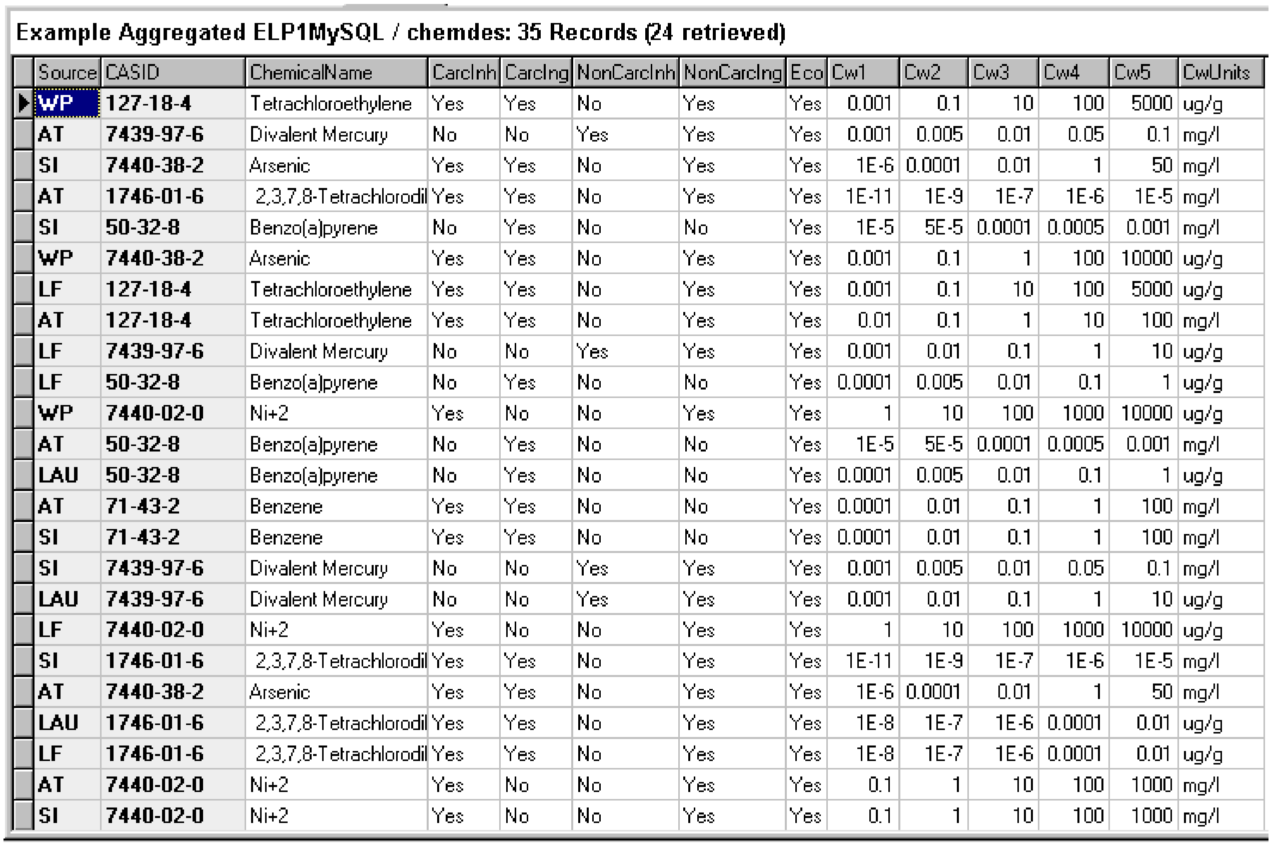
Figure 6-8e. Example SiteIterIndex Table in Aggregated ELP1 MySQL Database.
(Site Iteration Index Summary; Representing a List of Model Runs in the Scenario Set)



Example Aggregated ELP1MySQL / bindes: 1 Records (1 retrieved)

RBin1	RBin2	RBin3	RBin4	RBin5	RBin6	RBin7	HBin1	HBin2	HBin3	HBin4	EBin1	EBin2	EBin3	EBin4	EBin5	EBin6
1E-9	1E-8	5E-7	1E-6	5E-6	1E-5	0.0001	0.01	0.1	1	10	0.01	0.1	1	10	100	

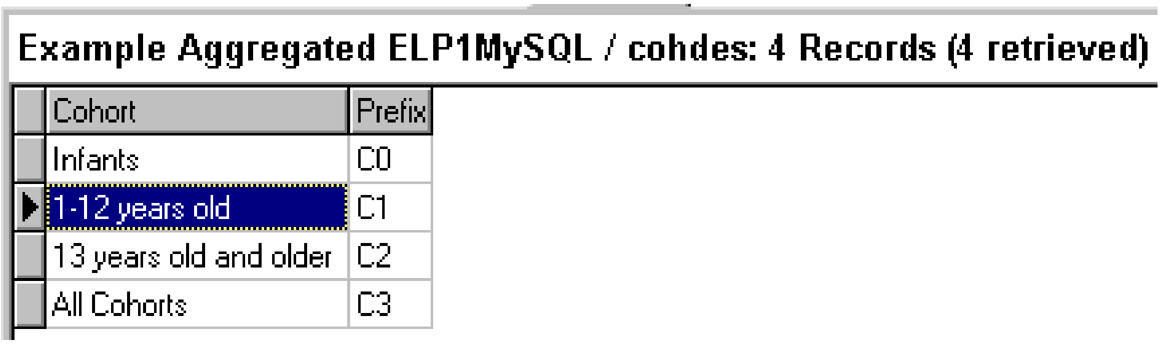
Figure 6-8f. Example BinDes Table in Aggregated ELP1 MySQL Database.
(Cancer/Hazard Risk Bins)



Example Aggregated ELP1MySQL / chemdes: 35 Records (24 retrieved)

Source	CASID	ChemicalName	CarcInh	CarcIng	NonCarcInh	NonCarcIng	Eco	Cw1	Cw2	Cw3	Cw4	Cw5	CwUnits
WP	127-18-4	Tetrachloroethylene	Yes	Yes	No	Yes	Yes	0.001	0.1	10	100	5000	ug/g
AT	7439-97-6	Divalent Mercury	No	No	Yes	Yes	Yes	0.001	0.005	0.01	0.05	0.1	mg/l
SI	7440-38-2	Arsenic	Yes	Yes	No	Yes	Yes	1E-6	0.0001	0.01	1	50	mg/l
AT	1746-01-6	2,3,7,8-Tetrachlorodil	Yes	Yes	No	Yes	Yes	1E-11	1E-9	1E-7	1E-6	1E-5	mg/l
SI	50-32-8	Benzo(a)pyrene	No	Yes	No	No	Yes	1E-5	5E-5	0.0001	0.0005	0.001	mg/l
WP	7440-38-2	Arsenic	Yes	Yes	No	Yes	Yes	0.001	0.1	1	100	10000	ug/g
LF	127-18-4	Tetrachloroethylene	Yes	Yes	No	Yes	Yes	0.001	0.1	10	100	5000	ug/g
AT	127-18-4	Tetrachloroethylene	Yes	Yes	No	Yes	Yes	0.01	0.1	1	10	100	mg/l
LF	7439-97-6	Divalent Mercury	No	No	Yes	Yes	Yes	0.001	0.01	0.1	1	10	ug/g
LF	50-32-8	Benzo(a)pyrene	No	Yes	No	No	Yes	0.0001	0.005	0.01	0.1	1	ug/g
WP	7440-02-0	Ni+2	Yes	No	No	Yes	Yes	1	10	100	1000	10000	ug/g
AT	50-32-8	Benzo(a)pyrene	No	Yes	No	No	Yes	1E-5	5E-5	0.0001	0.0005	0.001	mg/l
LAU	50-32-8	Benzo(a)pyrene	No	Yes	No	No	Yes	0.0001	0.005	0.01	0.1	1	ug/g
AT	71-43-2	Benzene	Yes	Yes	No	No	Yes	0.0001	0.01	0.1	1	100	mg/l
SI	71-43-2	Benzene	Yes	Yes	No	No	Yes	0.0001	0.01	0.1	1	100	mg/l
SI	7439-97-6	Divalent Mercury	No	No	Yes	Yes	Yes	0.001	0.005	0.01	0.05	0.1	mg/l
LAU	7439-97-6	Divalent Mercury	No	No	Yes	Yes	Yes	0.001	0.01	0.1	1	10	ug/g
LF	7440-02-0	Ni+2	Yes	No	No	Yes	Yes	1	10	100	1000	10000	ug/g
SI	1746-01-6	2,3,7,8-Tetrachlorodil	Yes	Yes	No	Yes	Yes	1E-11	1E-9	1E-7	1E-6	1E-5	mg/l
AT	7440-38-2	Arsenic	Yes	Yes	No	Yes	Yes	1E-6	0.0001	0.01	1	50	mg/l
LAU	1746-01-6	2,3,7,8-Tetrachlorodil	Yes	Yes	No	Yes	Yes	1E-8	1E-7	1E-6	0.0001	0.01	ug/g
LF	1746-01-6	2,3,7,8-Tetrachlorodil	Yes	Yes	No	Yes	Yes	1E-8	1E-7	1E-6	0.0001	0.01	ug/g
AT	7440-02-0	Ni+2	Yes	No	No	Yes	Yes	0.1	1	10	100	1000	mg/l
SI	7440-02-0	Ni+2	Yes	No	No	Yes	Yes	0.1	1	10	100	1000	mg/l

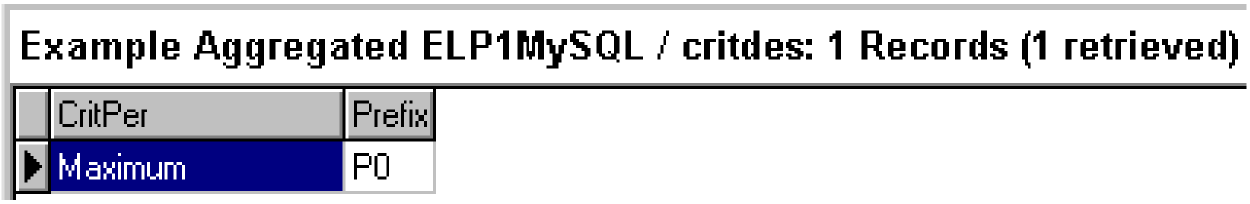
Figure 6-8g. Example ChemDes Table in Aggregated ELP1 MySQL Database.
(Chemical Descriptions)



Example Aggregated ELP1MySQL / cohdes: 4 Records (4 retrieved)

Cohort	Prefix
Infants	C0
1-12 years old	C1
13 years old and older	C2
All Cohorts	C3

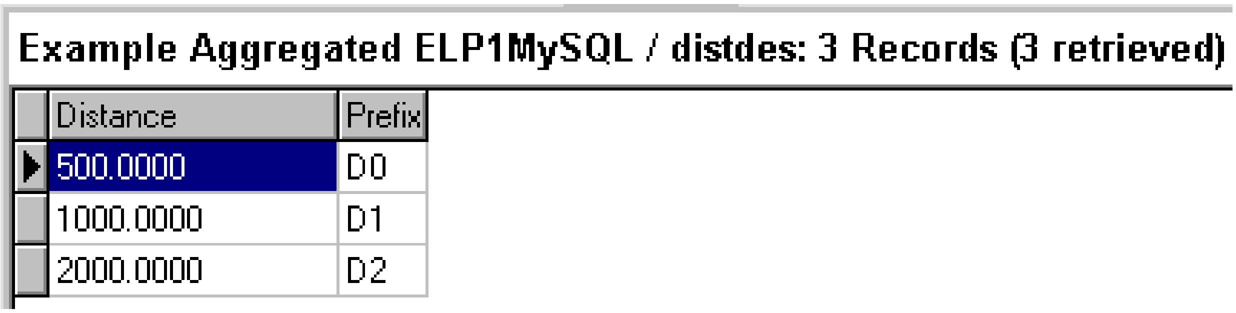
Figure 6-8h. Example CohDes Table in Aggregated ELP1 MySQL Database.
(Cohort Descriptions)



Example Aggregated ELP1MySQL / critdes: 1 Records (1 retrieved)

CritPer	Prefix
Maximum	P0

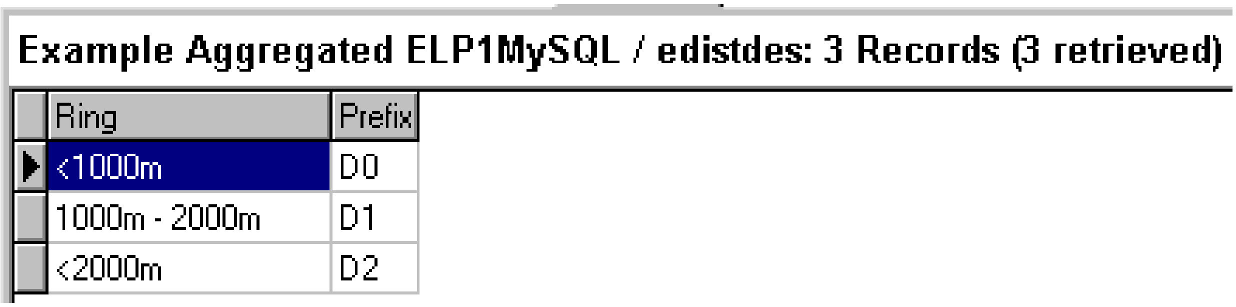
Figure 6-8i. Example CritDes Table in Aggregated ELP1 MySQL Database.
(Critical Time Period Descriptions)



Example Aggregated ELP1MySQL / distdes: 3 Records (3 retrieved)

Distance	Prefix
500.0000	D0
1000.0000	D1
2000.0000	D2

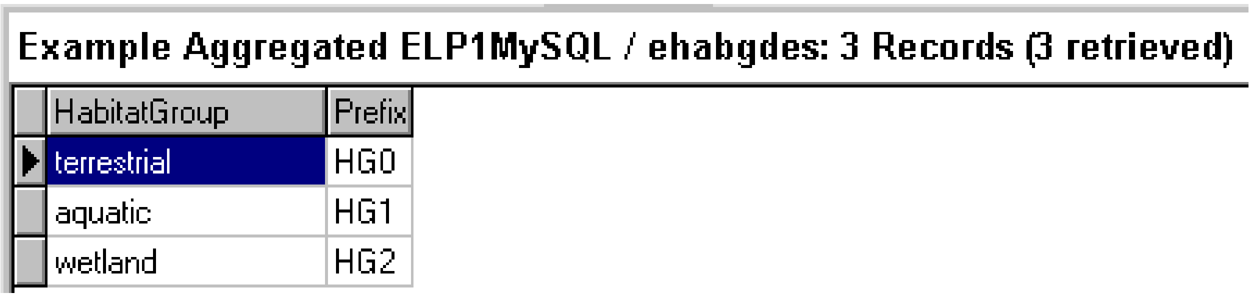
Figure 6-8j. Example DistDes Table in Aggregated ELP1 MySQL Database.
(Ring Distance Descriptions for Humans)



Example Aggregated ELP1MySQL / edistdes: 3 Records (3 retrieved)

Ring	Prefix
<1000m	D0
1000m - 2000m	D1
<2000m	D2

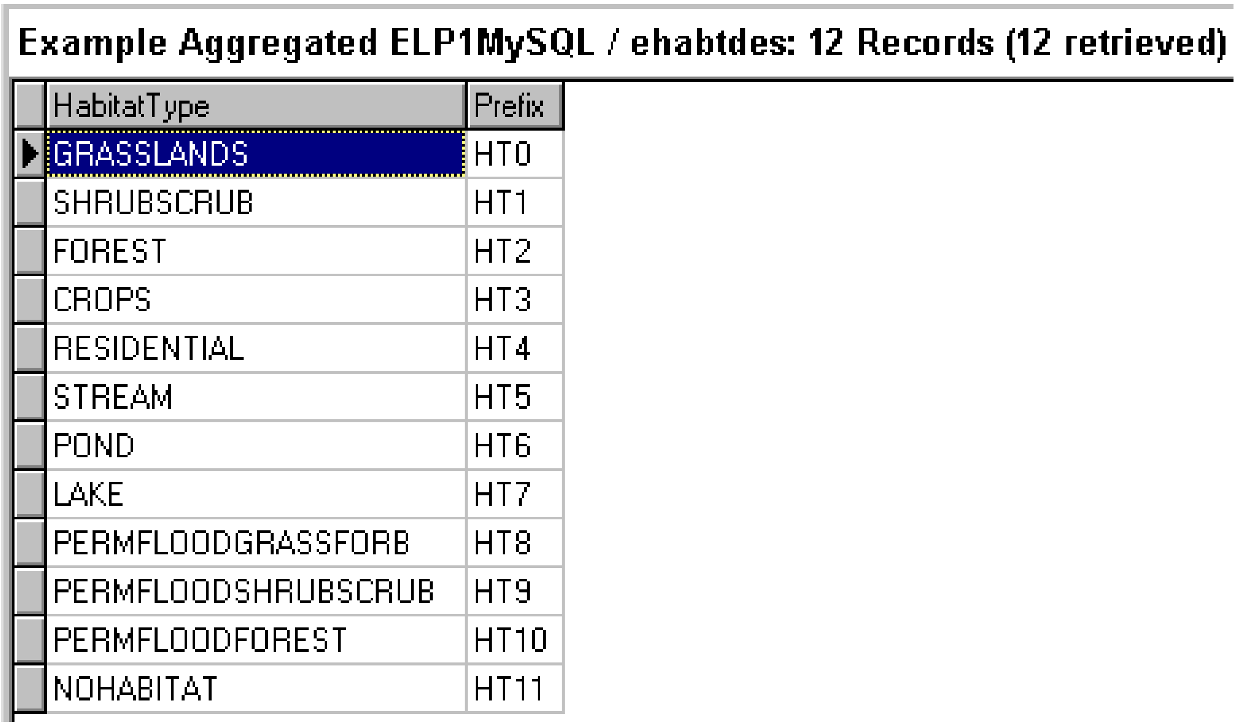
Figure 6-8k. Example EDistDes Table in Aggregated ELP1 MySQL Database.
(Ring Distance Descriptions for Ecology)



Example Aggregated ELP1MySQL / ehabgdes: 3 Records (3 retrieved)

HabitatGroup	Prefix
terrestrial	HG0
aquatic	HG1
wetland	HG2

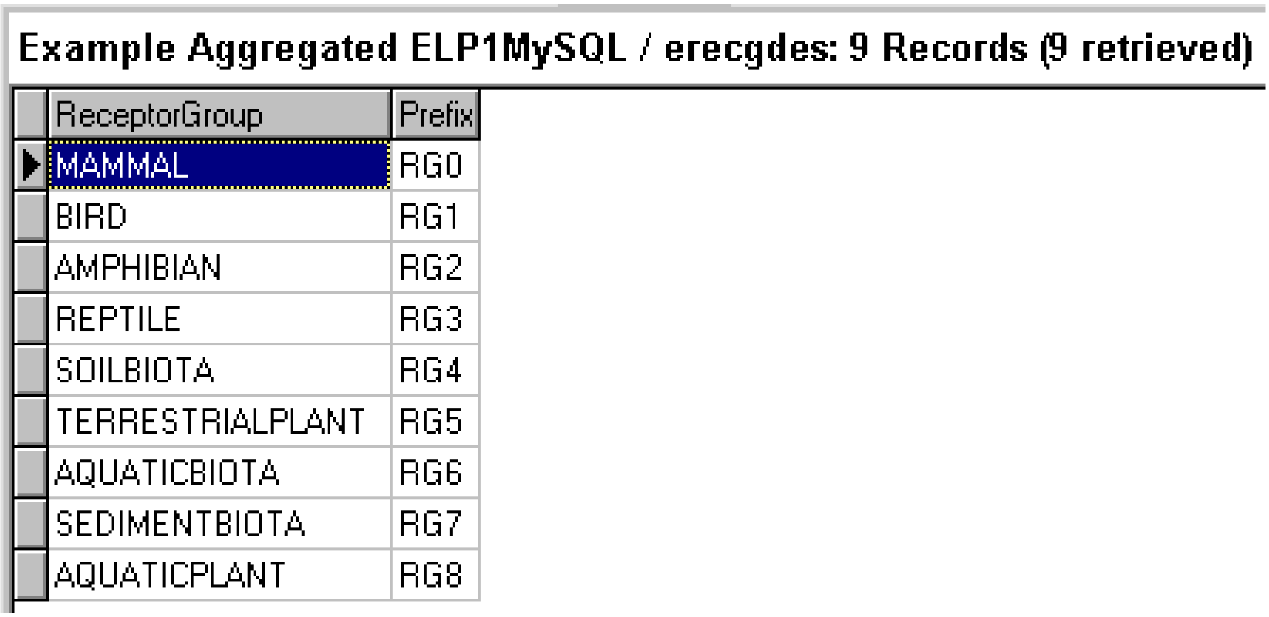
Figure 6-8l. Example EHabGDes Table in Aggregated ELP1 MySQL Database.
(Ecological Habitat Group Descriptions)



Example Aggregated ELP1MySQL / ehabtdes: 12 Records (12 retrieved)

HabitatType	Prefix
GRASSLANDS	HT0
SHRUBSCRUB	HT1
FOREST	HT2
CROPS	HT3
RESIDENTIAL	HT4
STREAM	HT5
POND	HT6
LAKE	HT7
PERMFLOODGRASSFORB	HT8
PERMFLOODSHRUBSCRUB	HT9
PERMFLOODFOREST	HT10
NOHABITAT	HT11

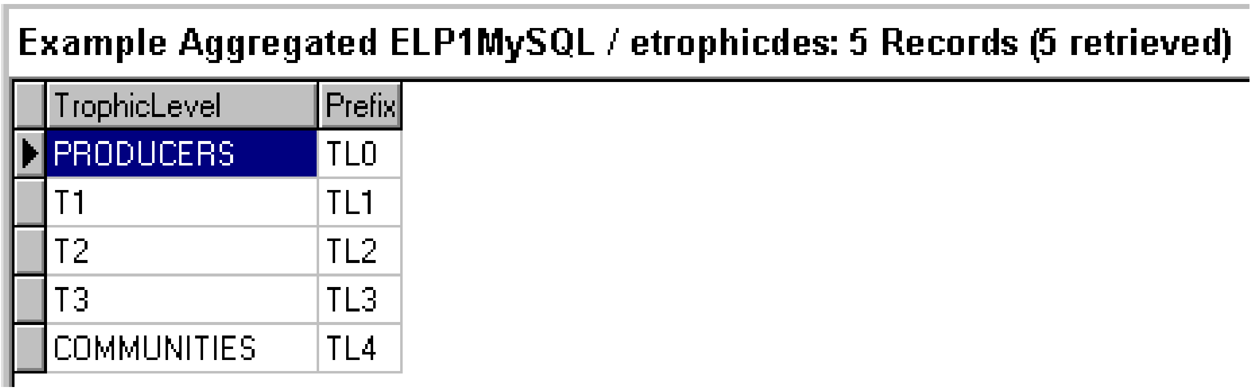
Figure 6-8m Example EHabTDes Table in Aggregated ELP1 MySQL Database.
(Ecological Habitat Type Descriptions)



Example Aggregated ELP1MySQL / erecgdes: 9 Records (9 retrieved)

ReceptorGroup	Prefix
MAMMAL	RG0
BIRD	RG1
AMPHIBIAN	RG2
REPTILE	RG3
SOILBIOTA	RG4
TERRESTRIALPLANT	RG5
AQUATICBIOTA	RG6
SEDIMENTBIOTA	RG7
AQUATICPLANT	RG8

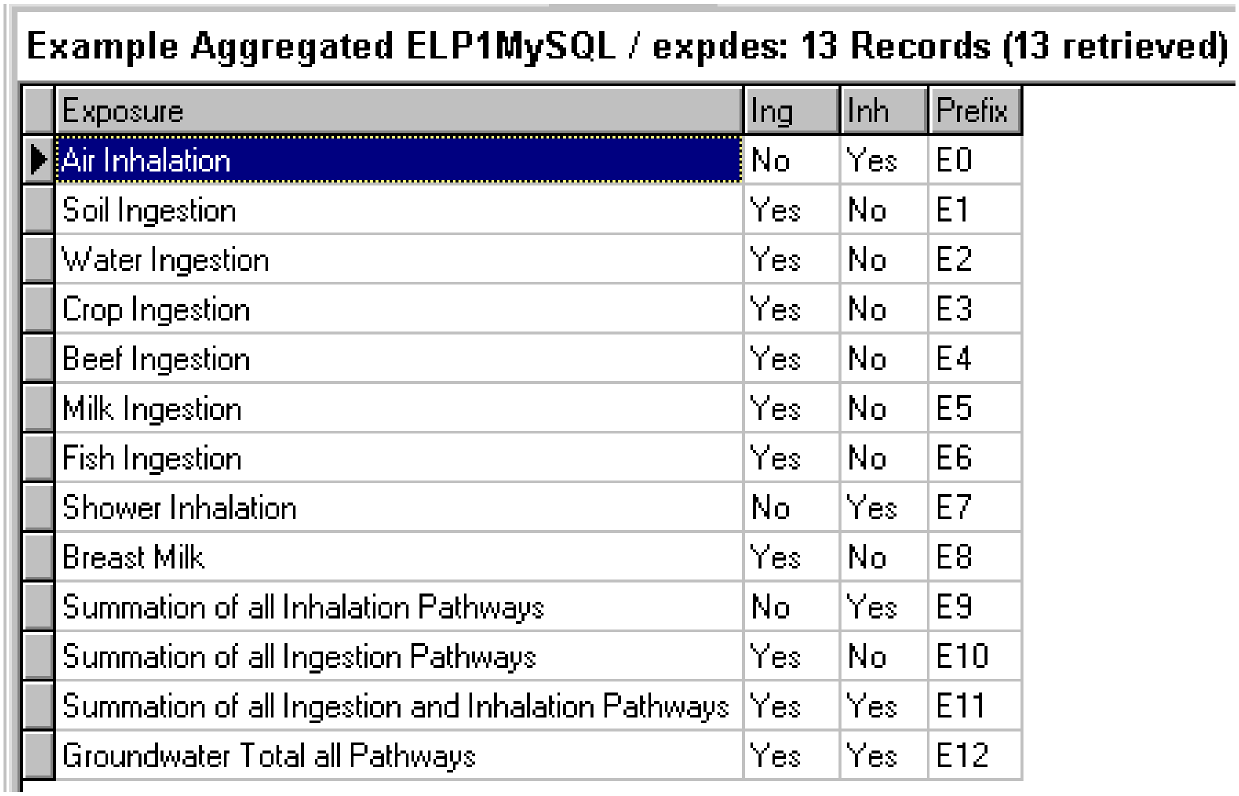
Figure 6-8n. Example ERecGDes Table in Aggregated ELP1 MySQL Database.
(Ecological Receptor Group Descriptions)



Example Aggregated ELP1MySQL / etrophicdes: 5 Records (5 retrieved)

TrophicLevel	Prefix
PRODUCERS	TL0
T1	TL1
T2	TL2
T3	TL3
COMMUNITIES	TL4

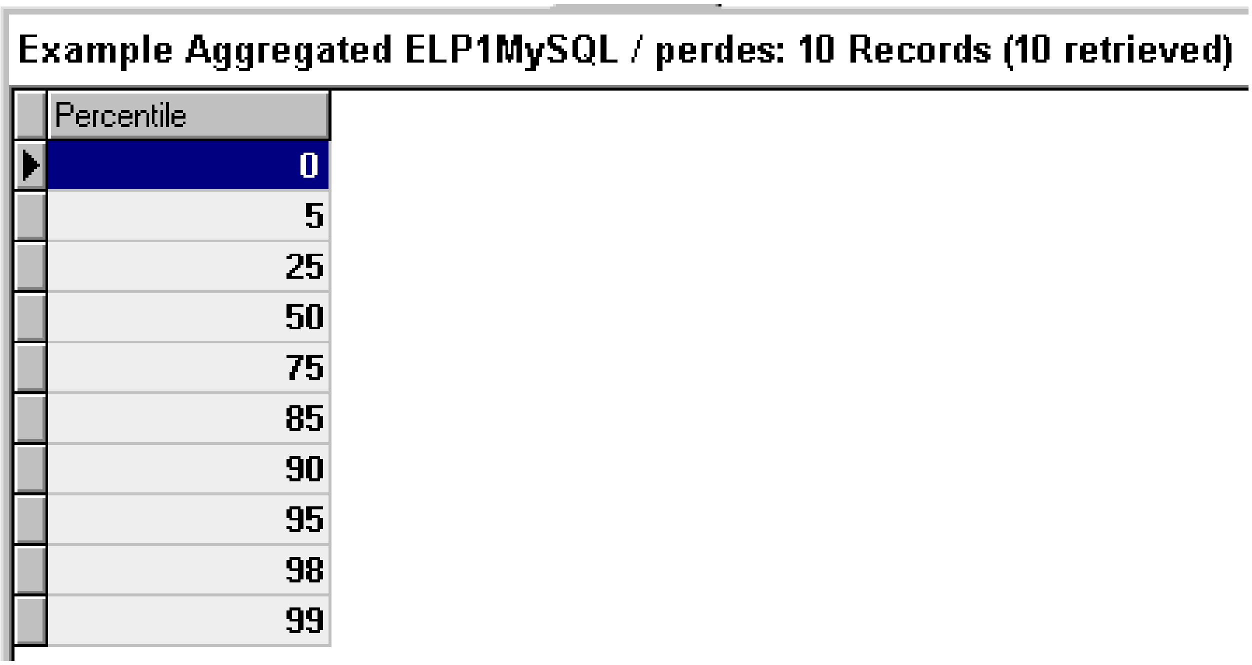
Figure 6-8o. Example ETrophicDes Table in Aggregated ELP1 MySQL Database.
(Ecological Trophic Level Descriptions)



Example Aggregated ELP1MySQL / expdes: 13 Records (13 retrieved)

Exposure	Ing	Inh	Prefix
Air Inhalation	No	Yes	E0
Soil Ingestion	Yes	No	E1
Water Ingestion	Yes	No	E2
Crop Ingestion	Yes	No	E3
Beef Ingestion	Yes	No	E4
Milk Ingestion	Yes	No	E5
Fish Ingestion	Yes	No	E6
Shower Inhalation	No	Yes	E7
Breast Milk	Yes	No	E8
Summation of all Inhalation Pathways	No	Yes	E9
Summation of all Ingestion Pathways	Yes	No	E10
Summation of all Ingestion and Inhalation Pathways	Yes	Yes	E11
Groundwater Total all Pathways	Yes	Yes	E12

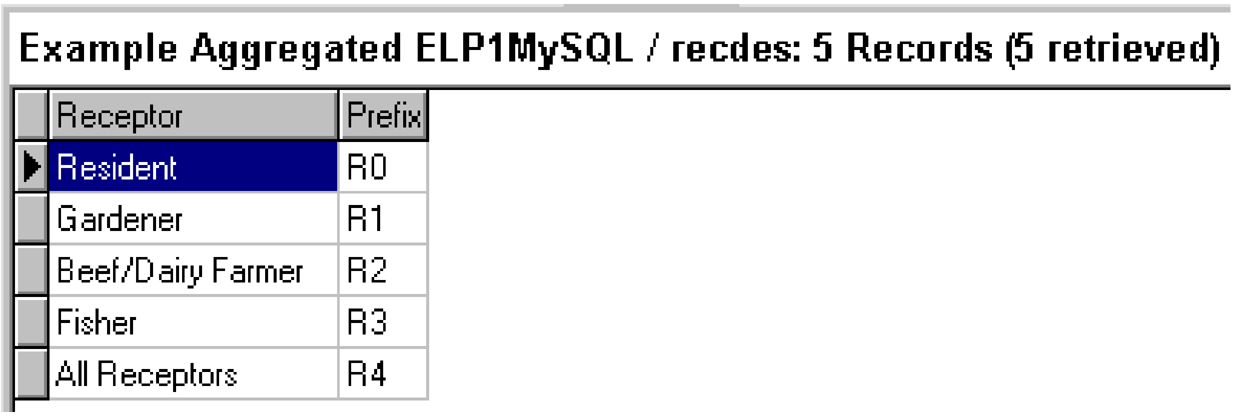
Figure 6-8p. Example ExpDes Table in Aggregated ELP1 MySQL Database.
(Exposure Pathway Descriptions)



Example Aggregated ELP1MySQL / perdes: 10 Records (10 retrieved)

Percentile
0
5
25
50
75
85
90
95
98
99

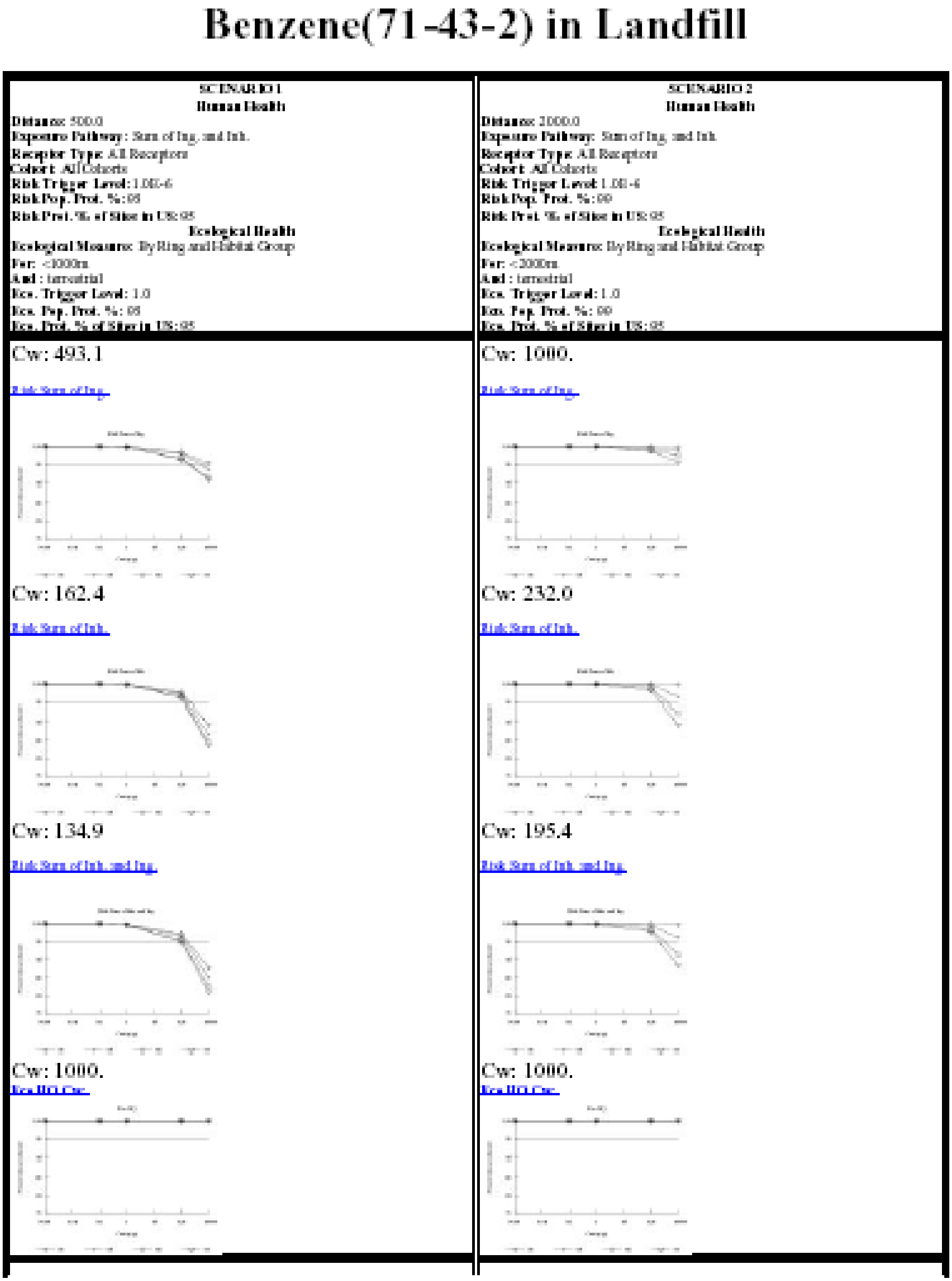
Figure 6-8q. Example PerDes Table in Aggregated ELP1 MySQL Database.
(Population Percentile Descriptions)



Example Aggregated ELP1MySQL / recdes: 5 Records (5 retrieved)

Receptor	Prefix
Resident	R0
Gardener	R1
Beef/Dairy Farmer	R2
Fisher	R3
All Receptors	R4

Figure 6-8r Example RecDes Table in Aggregated ELP1 MySQL Database.
(Human Receptor Group Descriptions)



Benzene(71-43-2) in Landfill

SCENARIO 1
Human Health
Distance: 500.0
Exposure Pathway: Sum of Ing. and Inh.
Receptor Type: All Receptors
Cohort: All Cohorts
Risk Trigger Level: 1.0E-6
Risk Pop. Prot. %: 95
Risk Prot. % of Sites in US: 95
Ecological Health
Ecological Measure: By Ring and Habitat Group
For: <1000m
And : terrestrial
Eco. Trigger Level: 1.0
Eco. Pop. Prot. %: 95
Eco. Prot. % of Sites in US: 95

Cw: 493.1
Risk Sum of Ing.

Cw: 162.4
Risk Sum of Inh.

Cw: 134.9
Risk Sum of Inh. and Ing.

Cw: 1000.
Eco HQ Cw.

SCENARIO 2
Human Health
Distance: 2000.0
Exposure Pathway: Sum of Ing. and Inh.
Receptor Type: All Receptors
Cohort: All Cohorts
Risk Trigger Level: 1.0E-6
Risk Pop. Prot. %: 99
Risk Prot. % of Sites in US: 95
Ecological Health
Ecological Measure: By Ring and Habitat Group
For: <2000m
And : terrestrial
Eco. Trigger Level: 1.0
Eco. Pop. Prot. %: 99
Eco. Prot. % of Sites in US: 95

Cw: 1000.
Risk Sum of Ing.

Cw: 232.0
Risk Sum of Inh.

Cw: 195.4
Risk Sum of Inh. and Ing.

Cw: 1000.
Eco HQ Cw.

Figure 6-9 Benzene-Landfill Example ELP2Vis Output.

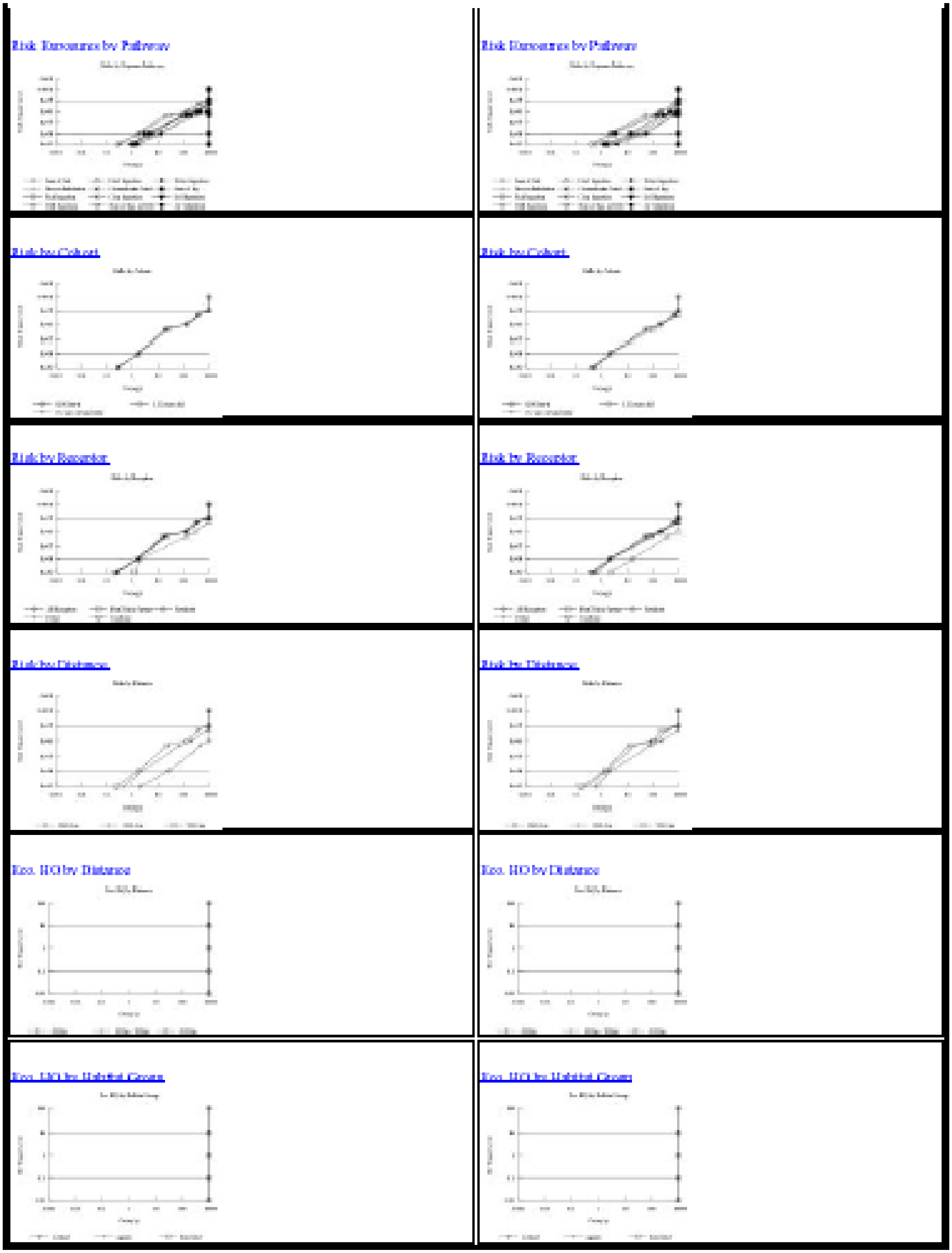
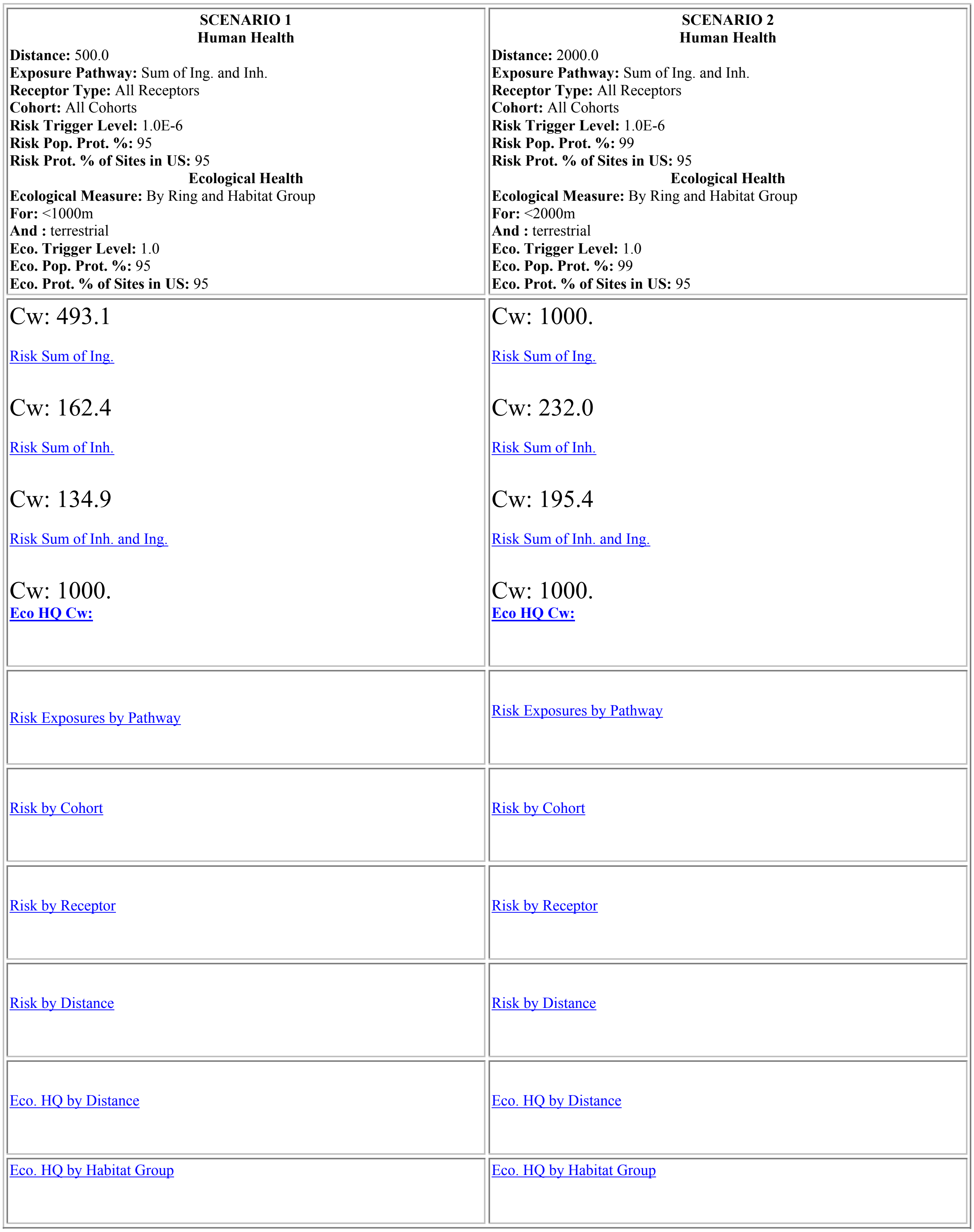


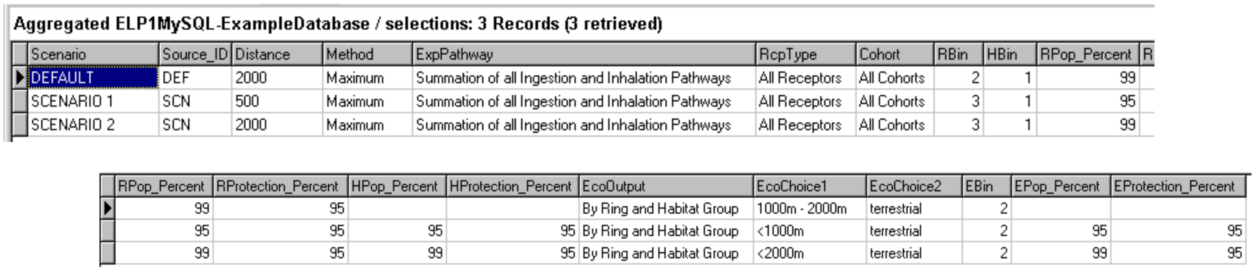
Figure 6-9 Benzene-Landfill Example ELP2Vis Output (Continued).

Benzene(71-43-2) in Landfill



SCENARIO 1 **Human Health** **Distance:** 500.0 **Exposure Pathway:** Sum of Ing. and Inh. **Receptor Type:** All Receptors **Cohort:** All Cohorts **Risk Trigger Level:** 1.0E-6 **Risk Pop. Prot. %:** 95 **Risk Prot. % of Sites in US:** 95 **Ecological Health** **Ecological Measure:** By Ring and Habitat Group **For:** <1000m **And :** terrestrial **Eco. Trigger Level:** 1.0 **Eco. Pop. Prot. %:** 95 **Eco. Prot. % of Sites in US:** 95	**SCENARIO 2** **Human Health** **Distance:** 2000.0 **Exposure Pathway:** Sum of Ing. and Inh. **Receptor Type:** All Receptors **Cohort:** All Cohorts **Risk Trigger Level:** 1.0E-6 **Risk Pop. Prot. %:** 99 **Risk Prot. % of Sites in US:** 95 **Ecological Health** **Ecological Measure:** By Ring and Habitat Group **For:** <2000m **And :** terrestrial **Eco. Trigger Level:** 1.0 **Eco. Pop. Prot. %:** 99 **Eco. Prot. % of Sites in US:** 95
Cw: 493.1 Risk Sum of Ing. Cw: 162.4 Risk Sum of Inh. Cw: 134.9 Risk Sum of Inh. and Ing. Cw: 1000. **Eco HQ Cw:**	Cw: 1000. Risk Sum of Ing. Cw: 232.0 Risk Sum of Inh. Cw: 195.4 Risk Sum of Inh. and Ing. Cw: 1000. **Eco HQ Cw:**
Risk Exposures by Pathway	Risk Exposures by Pathway
Risk by Cohort	Risk by Cohort
Risk by Receptor	Risk by Receptor
Risk by Distance	Risk by Distance
Eco. HQ by Distance	Eco. HQ by Distance
Eco. HQ by Habitat Group	Eco. HQ by Habitat Group

Figure 6-10 Benzene-Landfill Example ELP2Vis Output (Less Thumbnail Graphics).



Aggregated ELP1MySQL-ExampleDatabase / selections: 3 Records (3 retrieved)

Scenario	Source_ID	Distance	Method	ExpPathway	RcpType	Cohort	RBin	HBin	RPop_Percent	R
DEFAULT	DEF	2000	Maximum	Summation of all Ingestion and Inhalation Pathways	All Receptors	All Cohorts	2	1	99	
SCENARIO 1	SCN	500	Maximum	Summation of all Ingestion and Inhalation Pathways	All Receptors	All Cohorts	3	1	95	
SCENARIO 2	SCN	2000	Maximum	Summation of all Ingestion and Inhalation Pathways	All Receptors	All Cohorts	3	1	99	

RPop_Percent	RProtection_Percent	HPop_Percent	HProtection_Percent	EcoOutput	EcoChoice1	EcoChoice2	EBin	EPop_Percent	EProtection_Percent
99	95			By Ring and Habitat Group	1000m - 2000m	terrestrial	2		
95	95	95	95	By Ring and Habitat Group	<1000m	terrestrial	2	95	95
99	95	99	95	By Ring and Habitat Group	<2000m	terrestrial	2	99	95

Figure 6-11. Example Scenario "Selections" Table in Aggregated ELP1 MySQL Database.
(Each Scenario record is used by ELP2Vis in defining html output columns)

Benzene(71-43-2) in Landfill
SCENARIO 1
Human Health
Distance: 500.0
Exposure Pathway: Sum of Ing.
Receptor Type: All Receptors
Cohort: All Cohorts
Risk Trigger Level: 1.0E-6
Risk Pop. Prot. %: 95
Risk Prot. % of Sites in US: 95

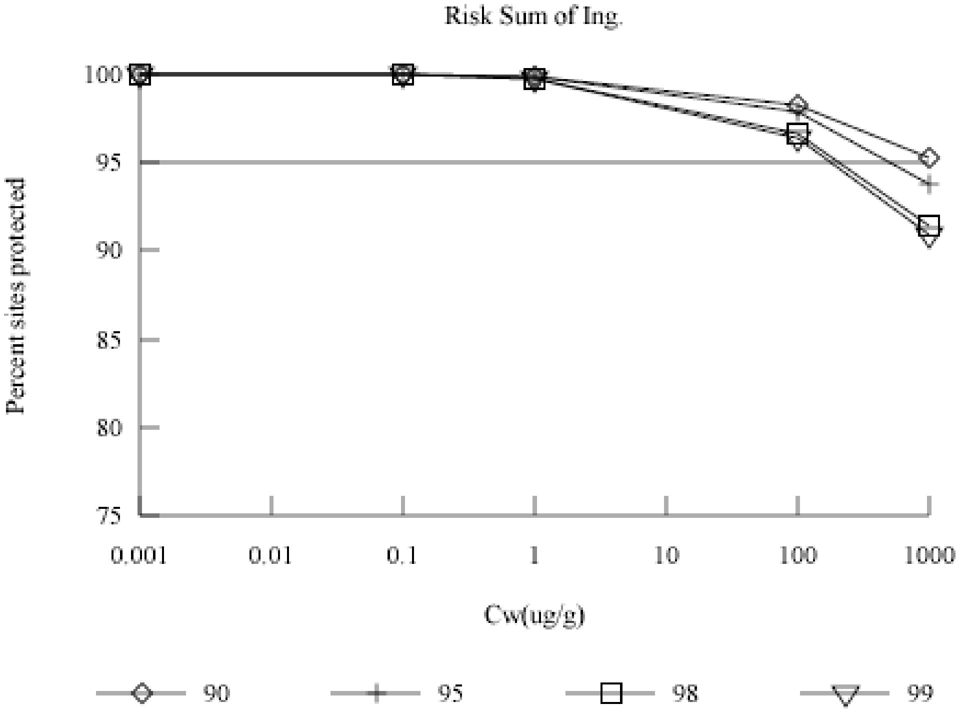


Figure 6-12a. ELP2Vis Output - Graphic Detail for Human Risk Summary of Ingestion.

Benzene(71-43-2) in Landfill
SCENARIO 1
Human Health
Distance: 500.0
Exposure Pathway: Sum of Inh.
Receptor Type: All Receptors
Cohort: All Cohorts
Risk Trigger Level: 1.0E-6
Risk Pop. Prot. %: 95
Risk Prot. % of Sites in US: 95

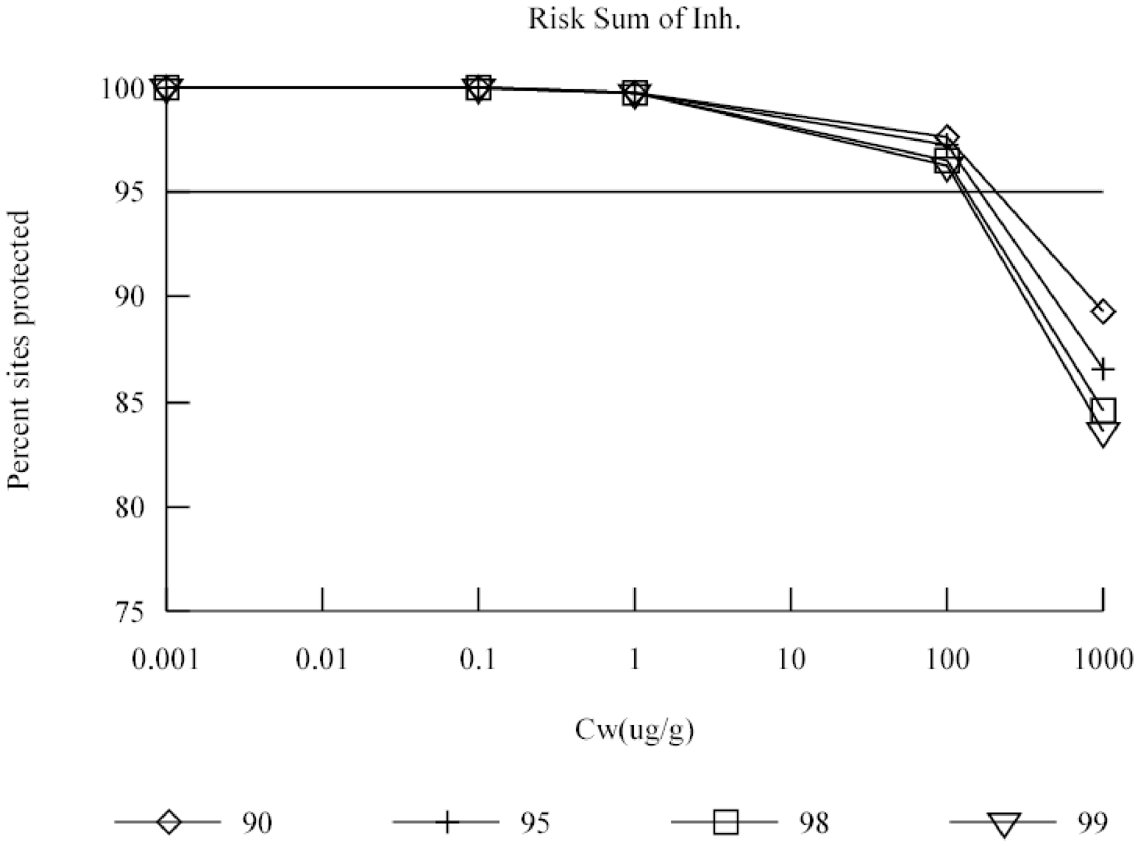


Figure 6-12b. ELP2Vis Output - Graphic Detail for Human Risk Summary of Inhalation.

Benzene(71-43-2) in Landfill
SCENARIO 1
Human Health
Distance: 500.0
Exposure Pathway: Sum of Ing. and Inh.
Receptor Type: All Receptors
Cohort: All Cohorts
Risk Trigger Level: 1.0E-6
Risk Pop. Prot. %: 95
Risk Prot. % of Sites in US: 95

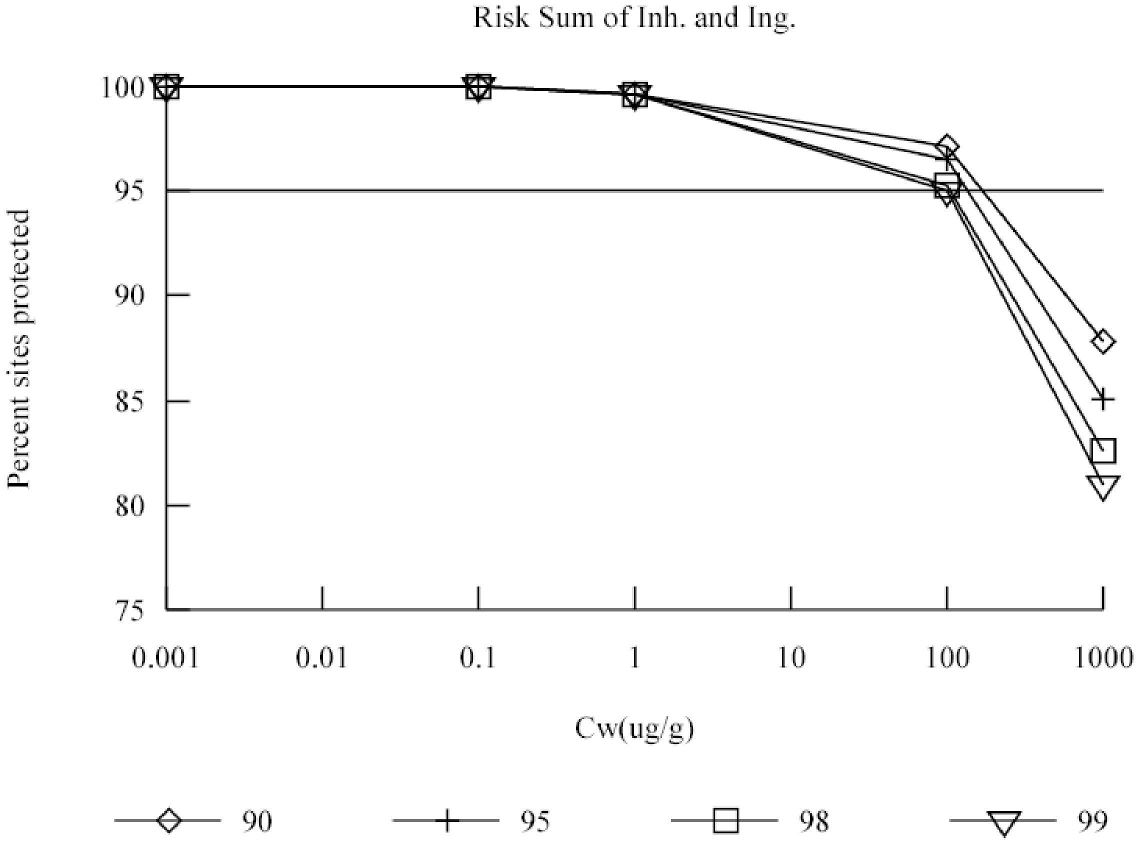


Figure 6-12c. ELP2Vis Output - Graphic Detail for Human Risk Summary of Inhalation Plus Ingestion.

Benzene(71-43-2) in Landfill
SCENARIO 1
Ecological Health
Ecological Measure: By Ring and Habitat Group
For: <1000m
And : terrestrial
Eco. Trigger Level: 1.0
Eco. Pop. Prot. %: 95
Eco. Prot. % of Sites in US: 95

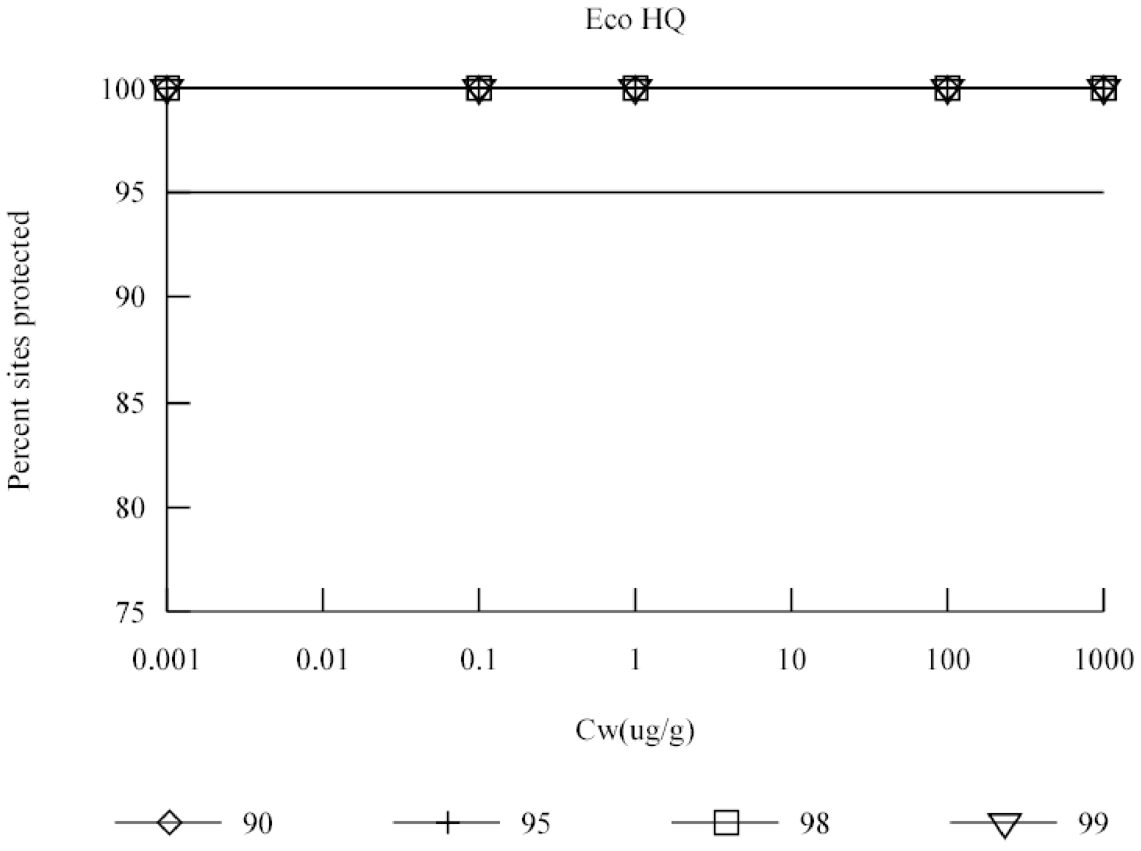


Figure 6-12d. ELP2Vis Output - Graphic Detail for Summary of Ecological Risk.

Benzene(71-43-2) in Landfill
SCENARIO 1
Human Health
Distance: 500.0
Exposure Pathway: Sum of Ing. and Inh.
Receptor Type: All Receptors
Cohort: All Cohorts
Risk Trigger Level: 1.0E-6
Risk Pop. Prot. %: 95
Risk Prot. % of Sites in US: 95

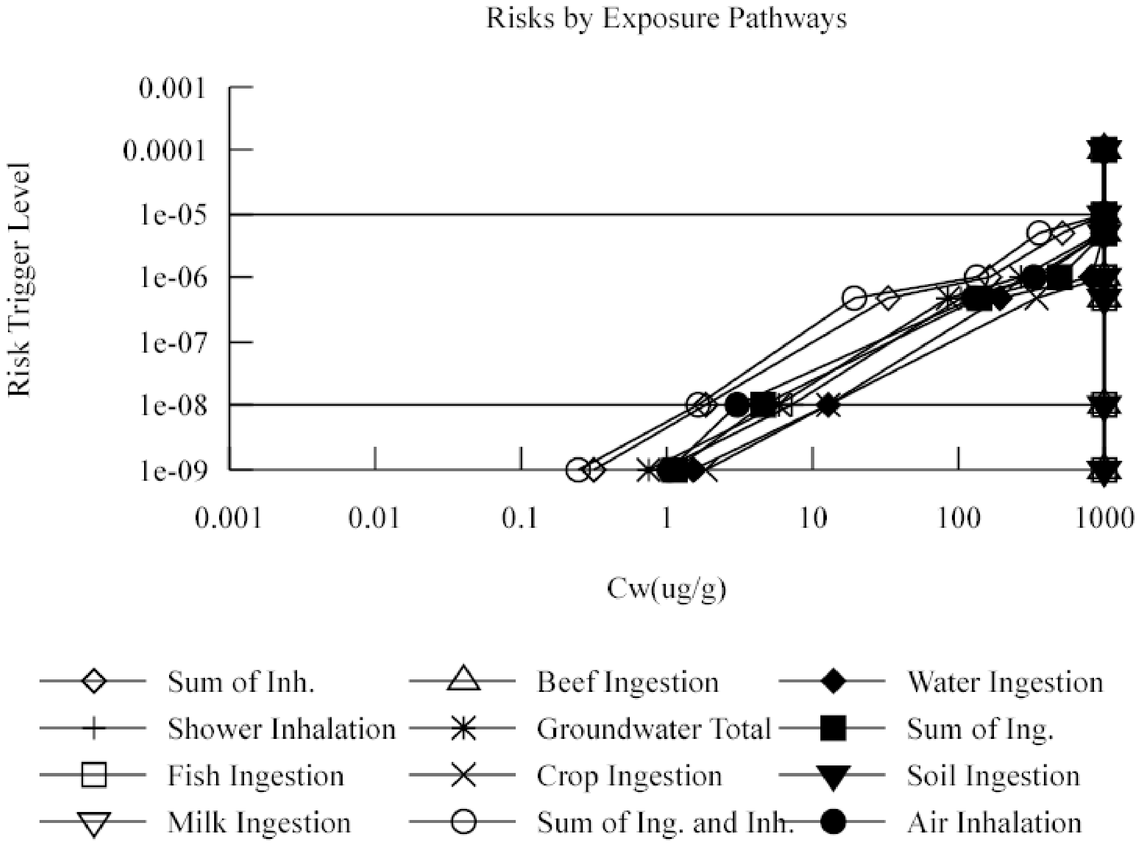


Figure 6-12e. ELP2Vis Output - Graphic Detail for Human Risk by Exposure Pathway.

Benzene(71-43-2) in Landfill
SCENARIO 1
Human Health
Distance: 500.0
Exposure Pathway: Sum of Ing. and Inh.
Receptor Type: All Receptors
Cohort: All Cohorts
Risk Trigger Level: 1.0E-6
Risk Pop. Prot. %: 95
Risk Prot. % of Sites in US: 95

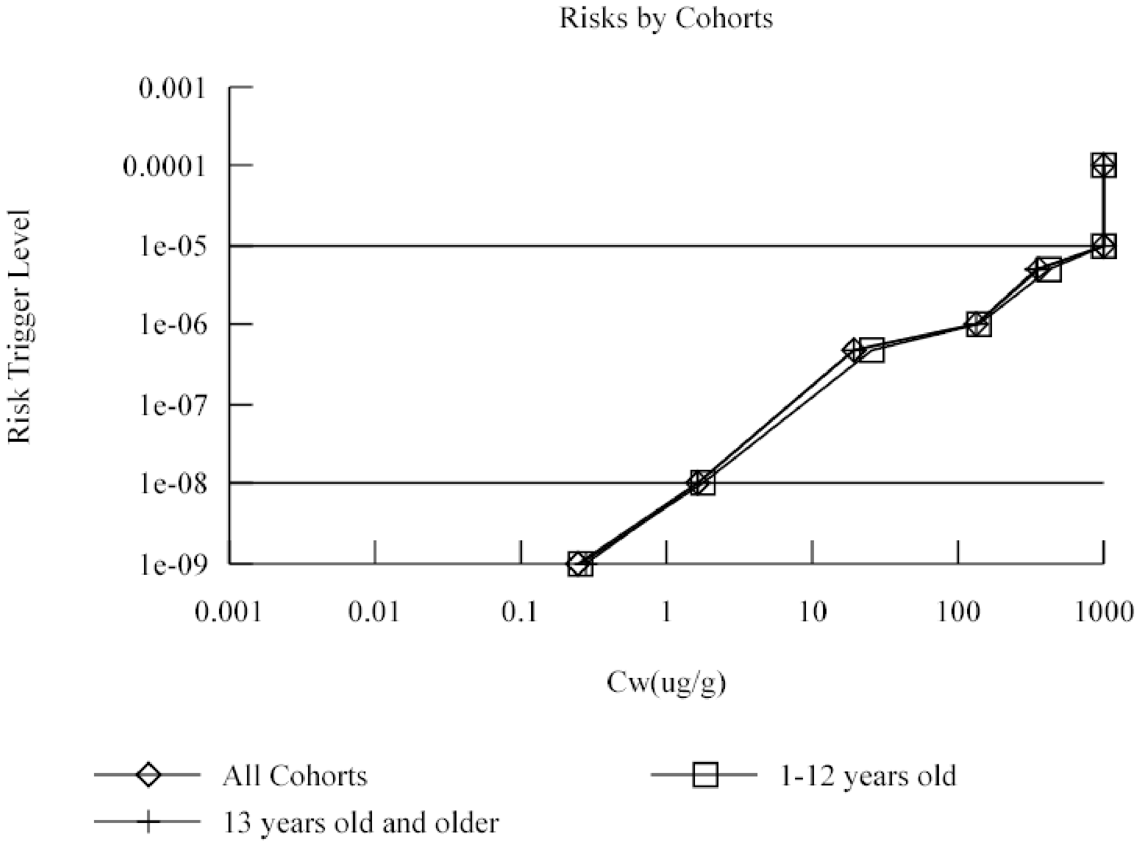


Figure 6-12f. ELP2Vis Output - Graphic Detail for Human Risk By Cohort Group.

Benzene(71-43-2) in Landfill
SCENARIO 1
Human Health
Distance: 500.0
Exposure Pathway: Sum of Ing. and Inh.
Receptor Type: All Receptors
Cohort: All Cohorts
Risk Trigger Level: 1.0E-6
Risk Pop. Prot. %: 95
Risk Prot. % of Sites in US: 95

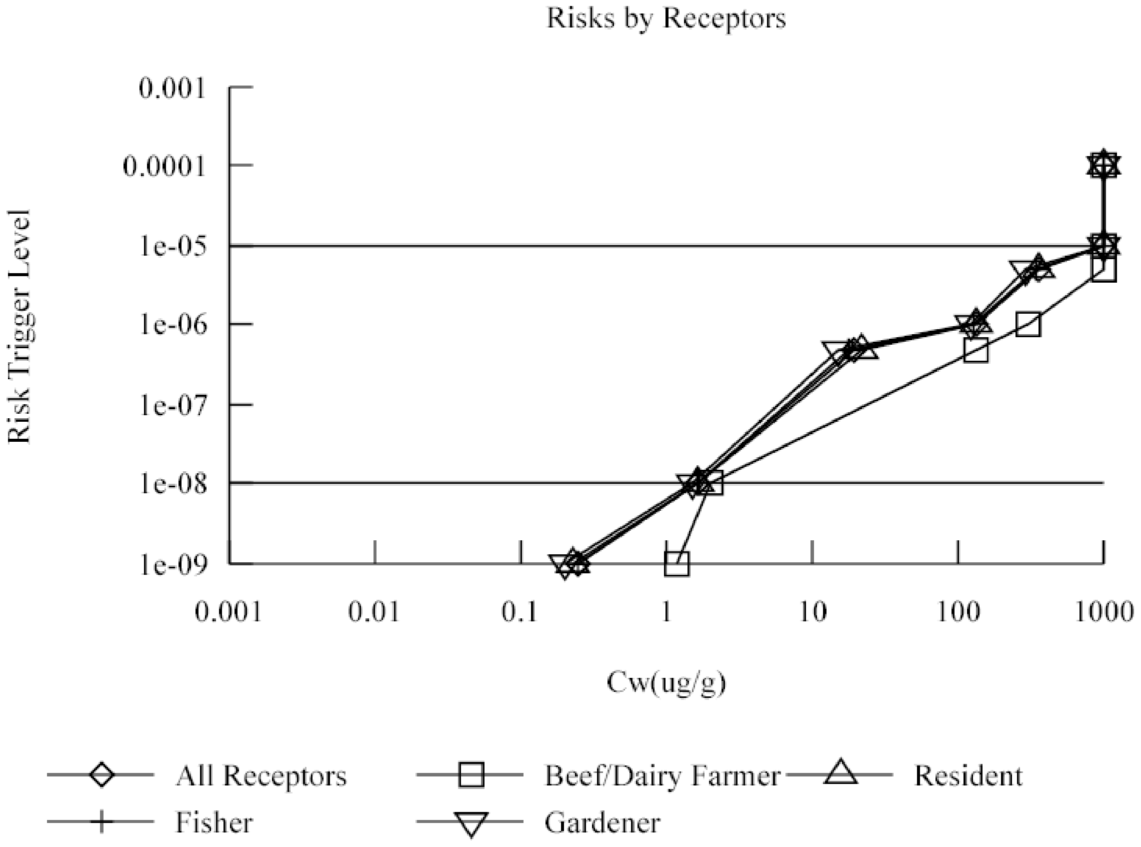


Figure 6-12g. ELP2Vis Output - Graphic Detail for Human Risk By Receptor Group.

Benzene(71-43-2) in Landfill
SCENARIO 1
Human Health
Distance: 500.0
Exposure Pathway: Sum of Ing. and Inh.
Receptor Type: Resident
Cohort: All Cohorts
Risk Trigger Level: 1.0E-6
Risk Pop. Prot. %: 95
Risk Prot. % of Sites in US: 95

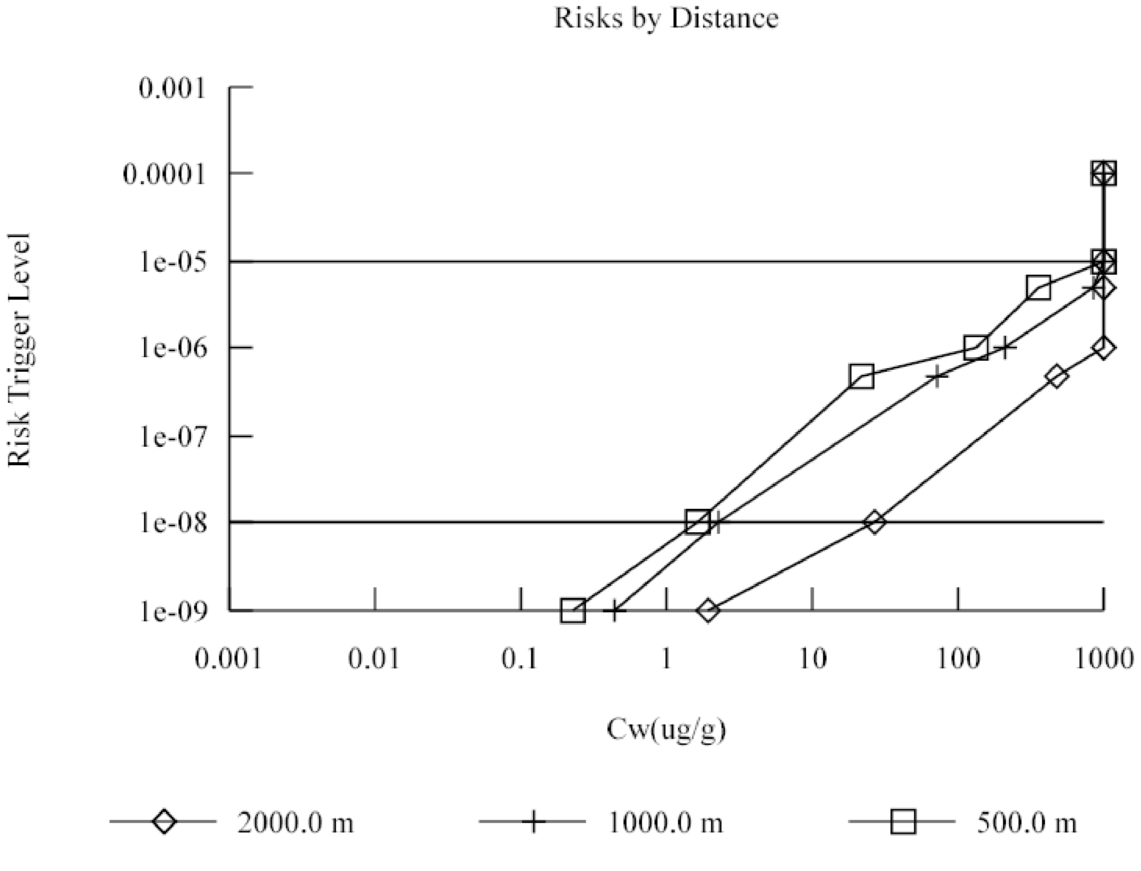


Figure 6-12h. ELP2Vis Output - Graphic Detail for Human Risk By Distance.

Benzene(71-43-2) in Landfill
SCENARIO 1
Ecological Health
Ecological Measure: By Ring and Habitat Group
For: <1000m
And : terrestrial
Eco. Trigger Level: 1.0
Eco. Pop. Prot. %: 95
Eco. Prot. % of Sites in US: 95

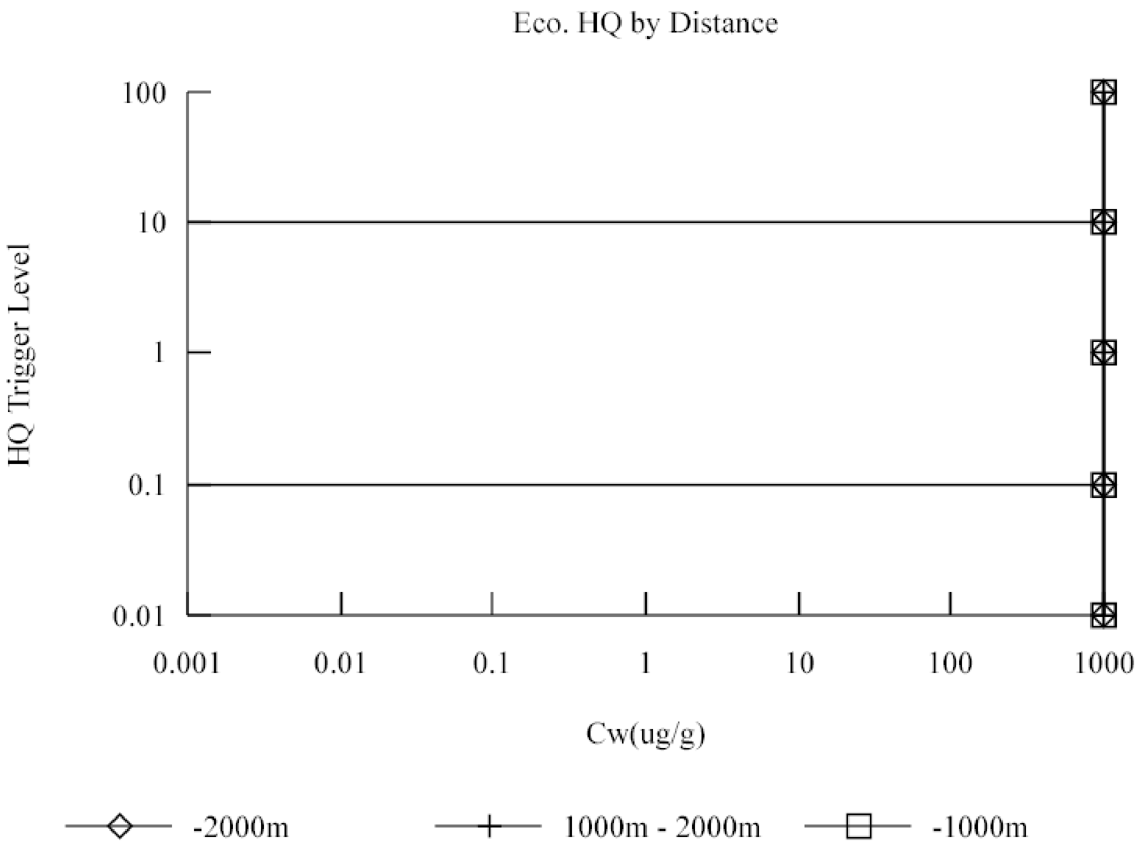


Figure 6-12i. ELP2Vis Output - Graphic Detail for Ecological Risk By Distance.

Benzene(71-43-2) in Landfill
SCENARIO 1
Ecological Health
Ecological Measure: By Ring and Habitat Group
For: <1000m
And : terrestrial
Eco. Trigger Level: 1.0
Eco. Pop. Prot. %: 95
Eco. Prot. % of Sites in US: 95

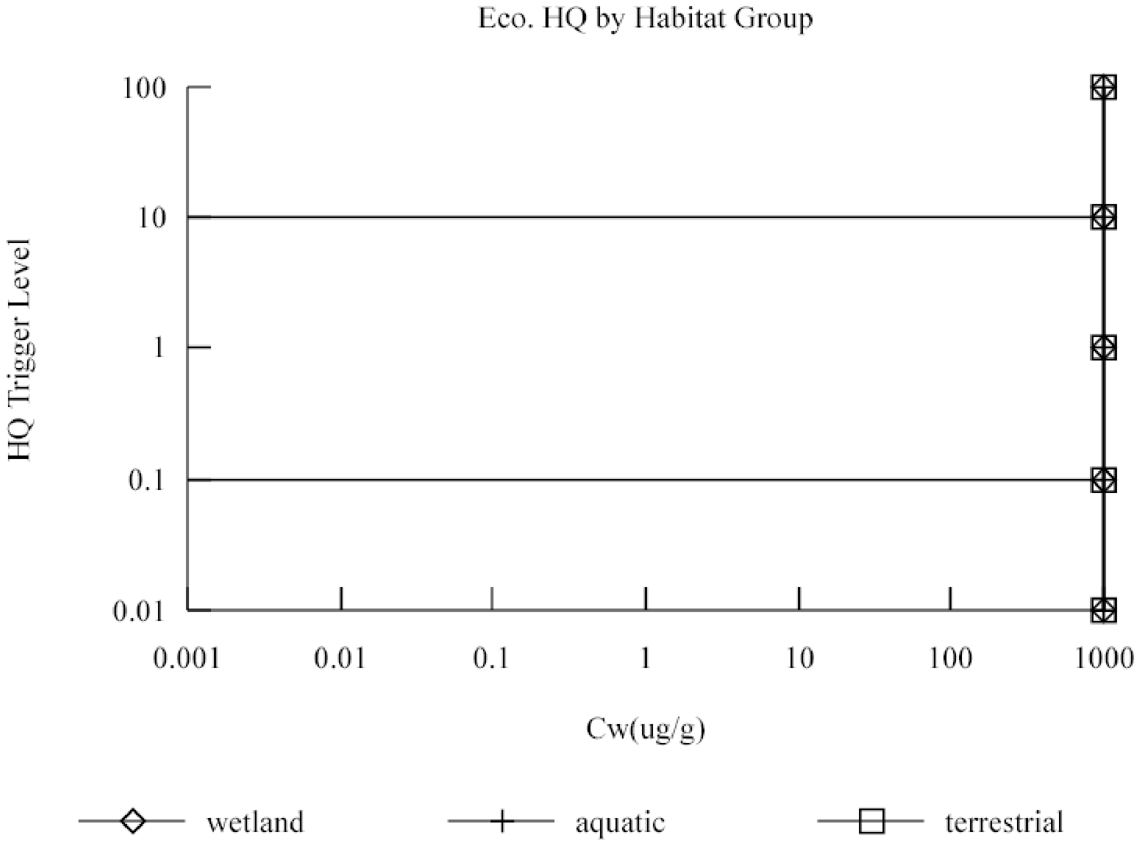
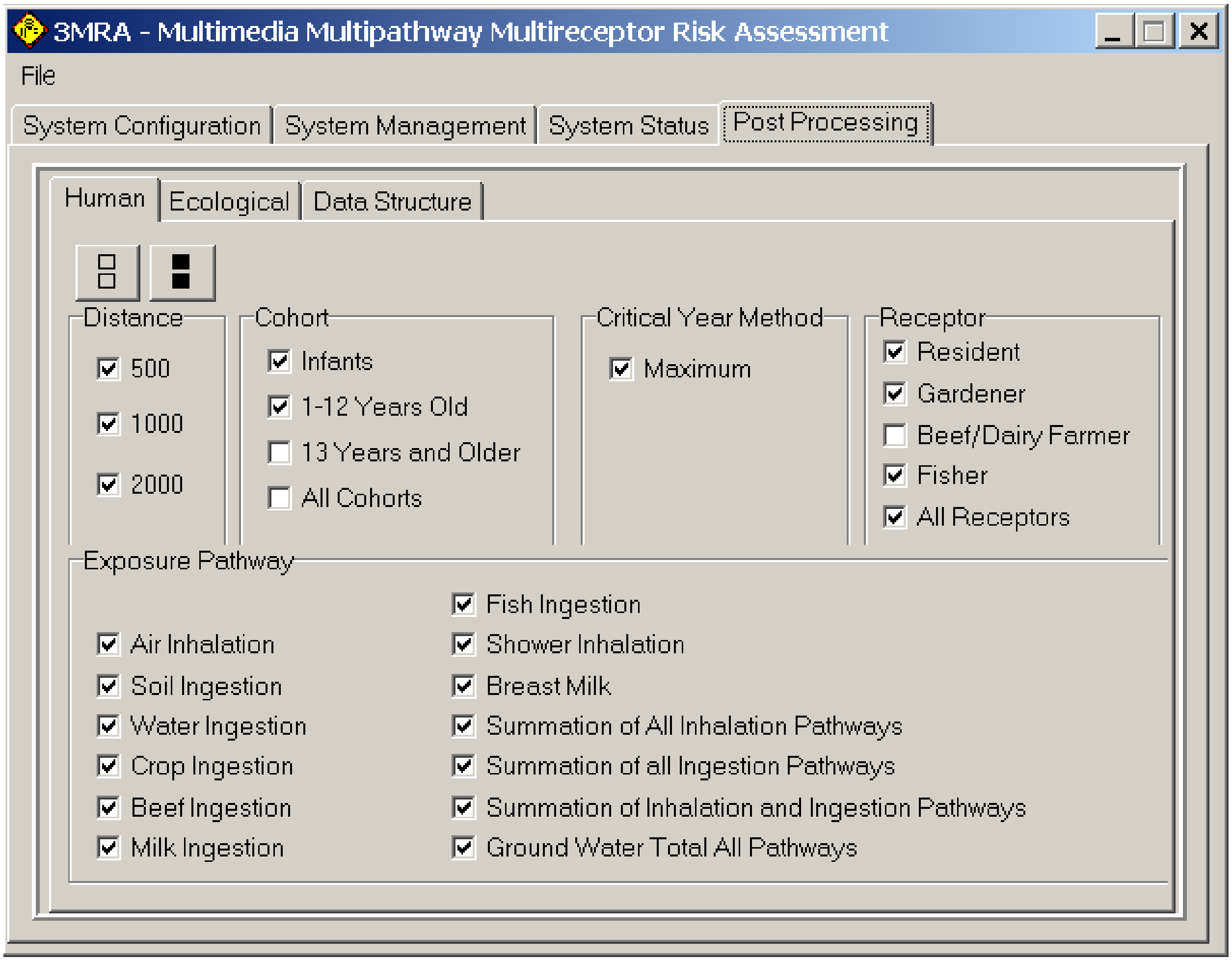


Figure 6-12j. ELP2Vis Output - Graphic Detail for Ecological Risk By Receptor Group.



Distance
☑ 500
☑ 1000
☑ 2000

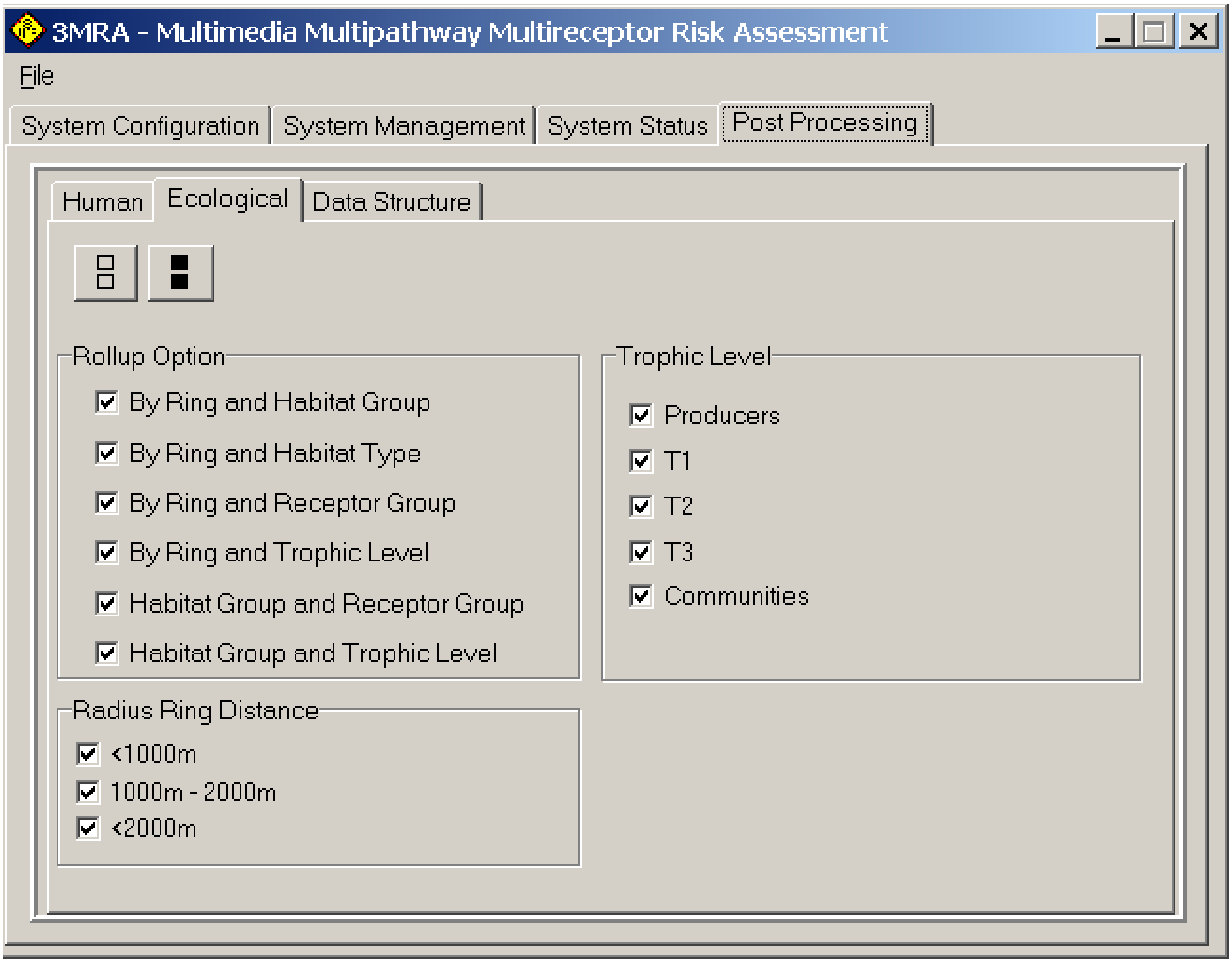
Cohort
☑ Infants
☑ 1-12 Years Old
□ 13 Years and Older
□ All Cohorts

Critical Year Method
☑ Maximum

Receptor
☑ Resident
☑ Gardener
□ Beef/Dairy Farmer
☑ Fisher
☑ All Receptors

Exposure Pathway
☑ Air Inhalation
☑ Soil Ingestion
☑ Water Ingestion
☑ Crop Ingestion
☑ Beef Ingestion
☑ Milk Ingestion
☑ Fish Ingestion
☑ Shower Inhalation
☑ Breast Milk
☑ Summation of All Inhalation Pathways
☑ Summation of all Ingestion Pathways
☑ Summation of Inhalation and Ingestion Pathways
☑ Ground Water Total All Pathways

Figure 6-13a. Enhanced SUI panel for Disaggregated ELP Processing (Human).



Rollup Option

☑ By Ring and Habitat Group

☑ By Ring and Habitat Type

☑ By Ring and Receptor Group

☑ By Ring and Trophic Level

☑ Habitat Group and Receptor Group

☑ Habitat Group and Trophic Level

Trophic Level

☑ Producers

☑ T1

☑ T2

☑ T3

☑ Communities

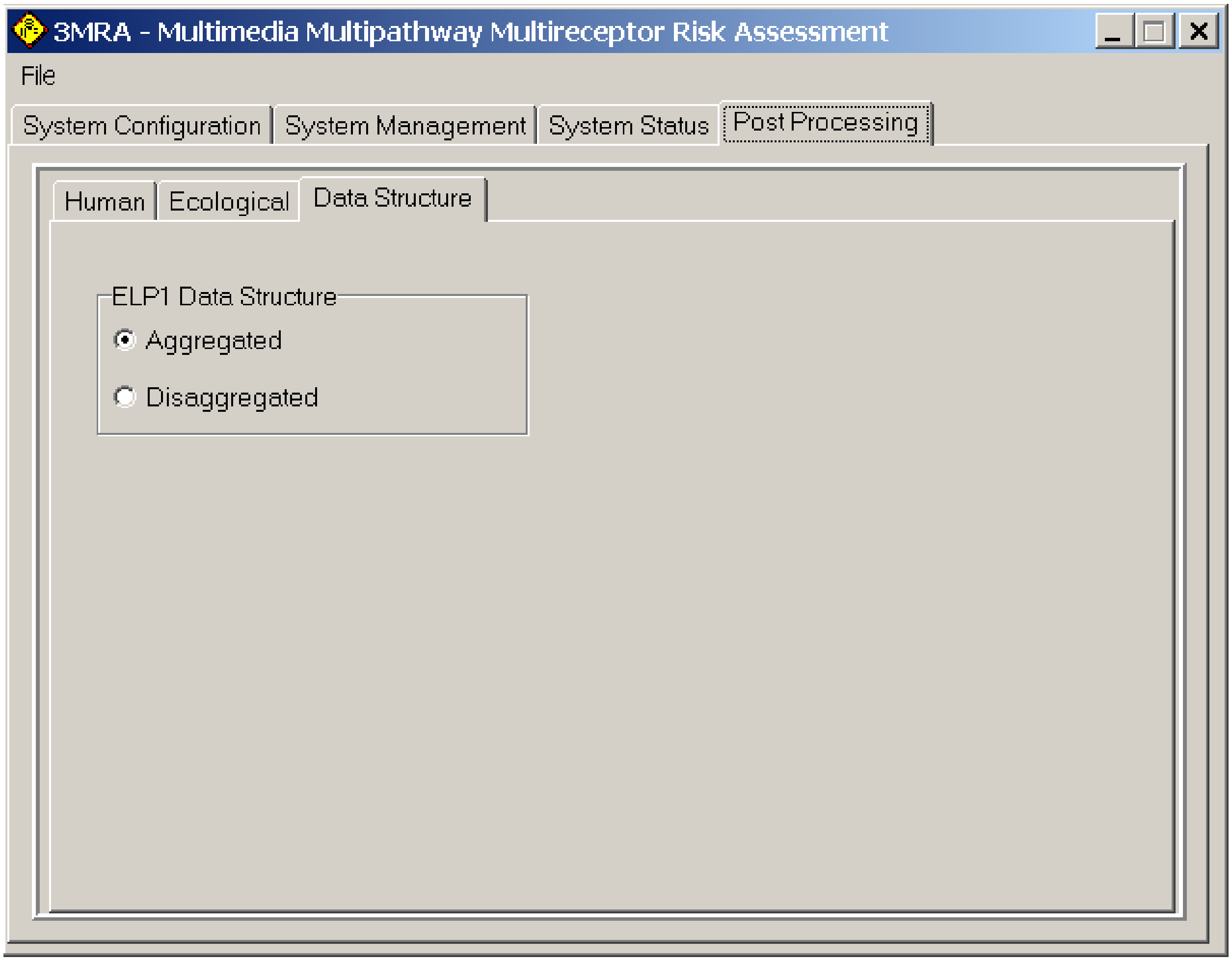
Radius Ring Distance

☑ <1000m

☑ 1000m - 2000m

☑ <2000m

Figure 6-13b. Enhanced SUI panel for Disaggregated ELP Processing (Ecological).

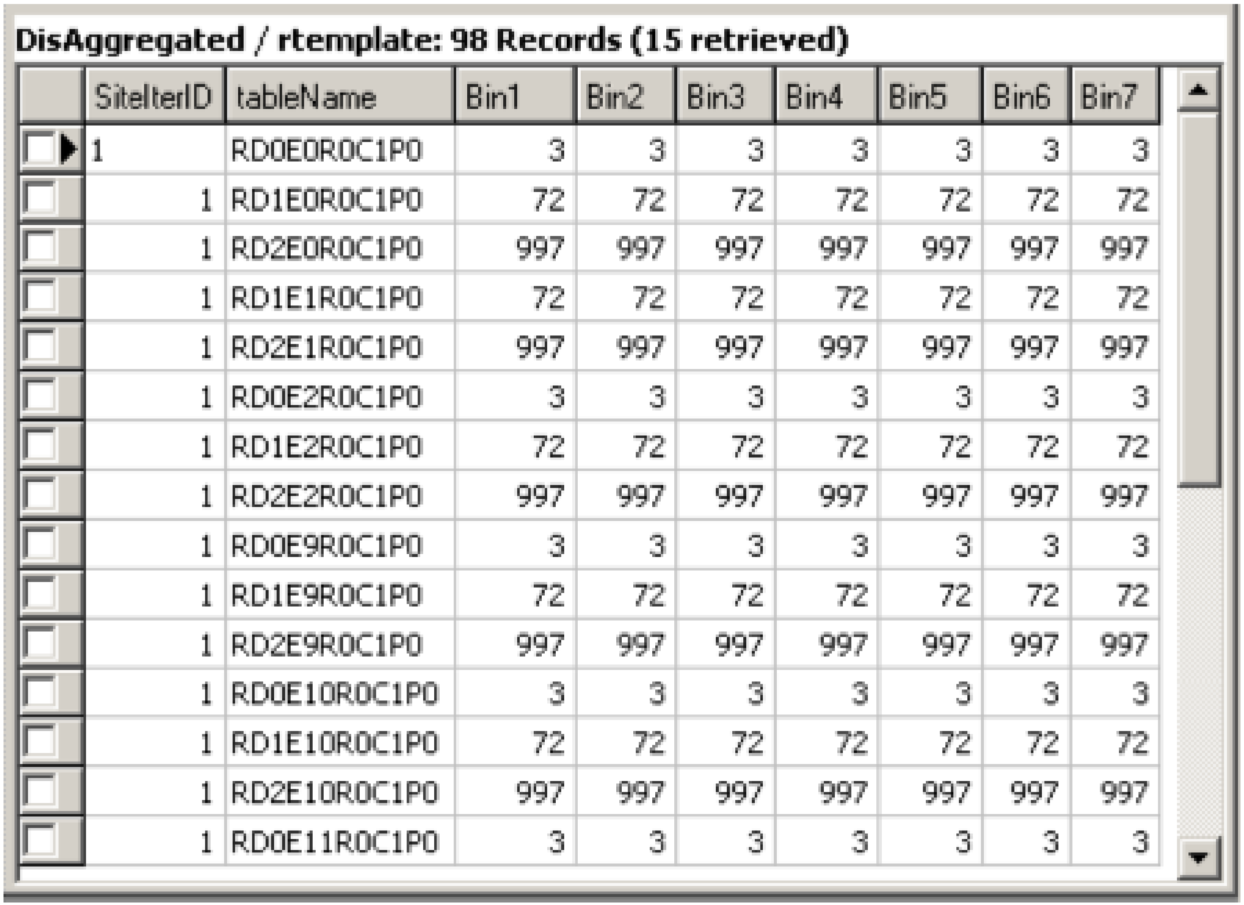


ELP1 Data Structure

● Aggregated

○ Disaggregated

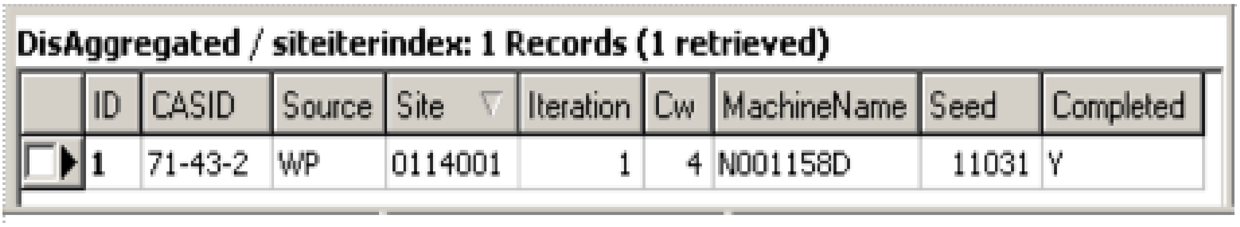
Figure 6-13c. Enhanced SUI panel for Disaggregated ELP Processing (Data Structure).



DisAggregated / rtemplate: 98 Records (15 retrieved)

	SiteIterID	tableName	Bin1	Bin2	Bin3	Bin4	Bin5	Bin6	Bin7
□ ▶	1	RD0E0R0C1P0	3	3	3	3	3	3	3
□	1	RD1E0R0C1P0	72	72	72	72	72	72	72
□	1	RD2E0R0C1P0	997	997	997	997	997	997	997
□	1	RD1E1R0C1P0	72	72	72	72	72	72	72
□	1	RD2E1R0C1P0	997	997	997	997	997	997	997
□	1	RD0E2R0C1P0	3	3	3	3	3	3	3
□	1	RD1E2R0C1P0	72	72	72	72	72	72	72
□	1	RD2E2R0C1P0	997	997	997	997	997	997	997
□	1	RD0E9R0C1P0	3	3	3	3	3	3	3
□	1	RD1E9R0C1P0	72	72	72	72	72	72	72
□	1	RD2E9R0C1P0	997	997	997	997	997	997	997
□	1	RD0E10R0C1P0	3	3	3	3	3	3	3
□	1	RD1E10R0C1P0	72	72	72	72	72	72	72
□	1	RD2E10R0C1P0	997	997	997	997	997	997	997
□	1	RD0E11R0C1P0	3	3	3	3	3	3	3

Figure 6-14a Example RTemplate Table in Disaggregated ELP1 MySQL Database.
(Human Cancer Risk Summary)



DisAggregated / siteiterindex: 1 Records (1 retrieved)

	ID	CASID	Source	Site	Iteration	Cw	MachineName	Seed	Completed
□ ▶	1	71-43-2	WP	0114001	1	4	N001158D	11031	Y

Figure 6-14b. Example SiteIterIndex Table in Disaggregated ELP1 MySQL Database.
(Site Iteration Index Summary)

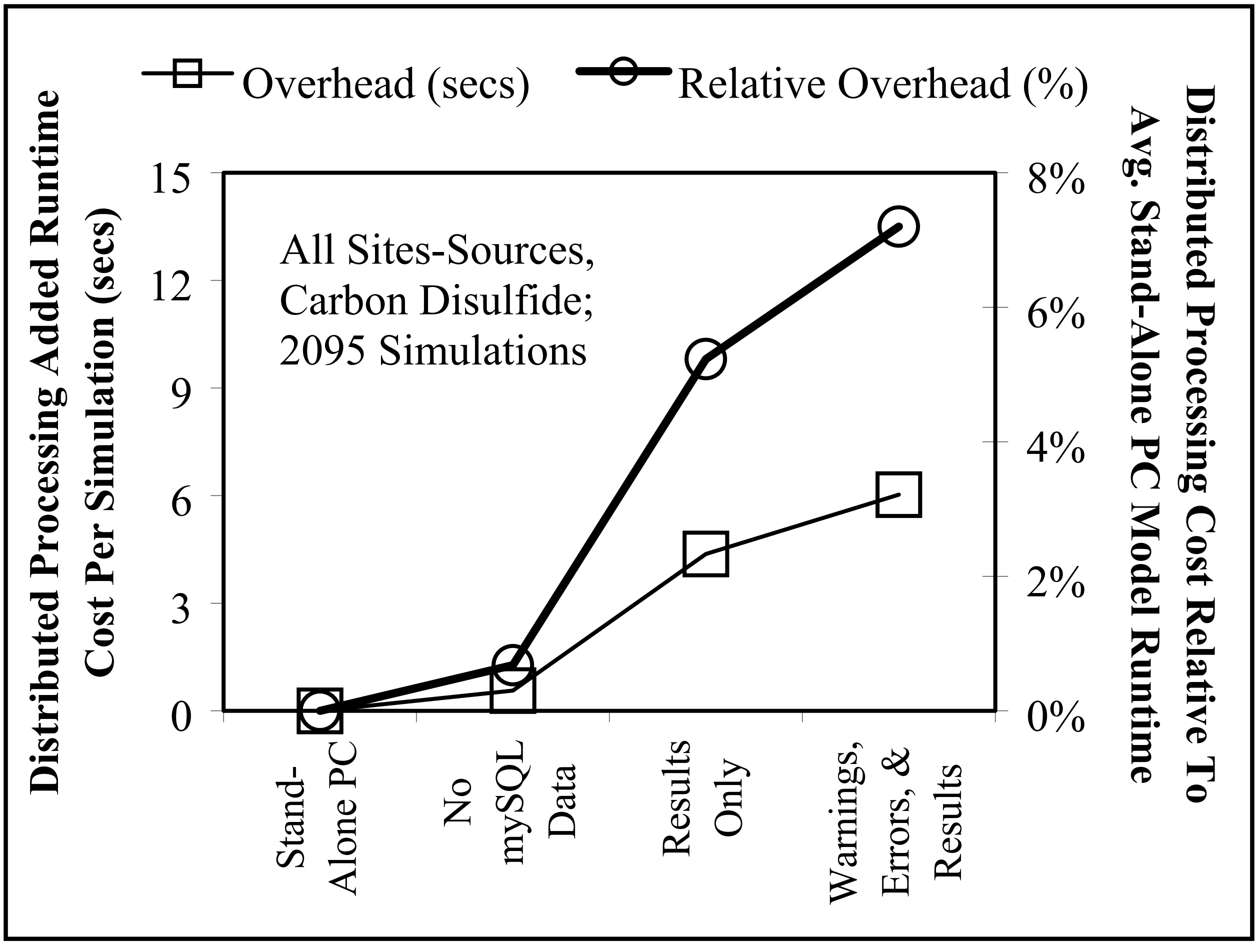


Figure 6-15. Comparison of Average 3MRA Runtimes: SuperMUSE Parallel Vs. Stand-Alone Execution Modes.

Table 6-1. Spearman-Rank Correlations for 3MRA Version 1.0 "Num" Type Parameters.

Num Parameter*	#Air	#Aqu	#AquWell**	#Farm	#Hab	#HabType	#HumRcp	#Vad	#WBN	#WSSub	WBN#Rch
#Air	1										
#Aqu	0.08	1									
#AquWell**	0.29	0.79	1								
#Farm	0.04	0.02	0.17	1							
#Hab	0.32	0.05	0.18	0.18	1						
#HabType	0.29	0.02	0.12	0.18	**0.84**	1					
#HumRcp	0.67	0.04	0.25	-0.16	0.00	-0.03	1				
#Vad	0.08	**1.00**	0.79	0.02	0.05	0.02	0.04	1			
#WBN	-0.07	0.00	0.06	-0.06	0.08	0.10	-0.08	0.00	1		
#WSSub	0.15	0.03	0.13	0.15	0.34	0.27	-0.02	0.03	0.16	1	
WBN#Rch	0.06	0.05	0.12	0.16	0.37	0.34	-0.13	0.05	-0.07	0.62	1

n = 419; for parameter <WSSubNumSubArea> no variation observed among 419 Site-WMU combinations.

* - All parameters except <NumVad> & <NumHabType> selected as candidate regressor for evaluation of model runtime.

** - Pearson Correlation coefficient = 0.55 for NumAqu; retained for analysis as candidate regressor.

Table 6-2. R^2 Values for Best Multiple Regressions Among "Num"Type Parameters.
(Based on 201 national sites; 419 site-WMU combinations)

Group	Parameter	1-Variable*		2-Variable		3-Variable		Predictability
VOCs	Benzene	0.33	#HumRcp	0.47	#WBN	0.50	WBN#Rch	Fair
	PCE	0.15	#WBN	0.30	#HumRcp	0.31	#Aqu	Fair
SVOCs	TCDD (Dioxin)	0.61	#HumRcp	0.86	#WBN	0.90	#Air	Good
	Benzo(a)pyrene	0.61	#HumRcp	0.88	#WBN	0.92	#Hab	Good
Metals	Arsenic	0.08	#Aqu	0.14	#HumRcp	0.15	#WBN	Poor
	Mercury*	0.12	#NumAqu	0.22	#WBN	0.30	#Air	poor

Table 6-3. 3MRA Version 1.0 Database: Organic Chemicals and Metals.

CASID	Chemical Name
106-93-4	Ethylene dibromide [1,2-Dibromoethane]
107-13-1	Acrylonitrile [2-Propenenitrile]
108-88-3	Toluene [Methylbenzene]
108-90-7	Chlorobenzene
108-95-2	Phenol
110-86-1	Pyridine
117-81-7	Bis-(2-ethylhexyl) phthalate [Di-2-ethylhexyl phthalate]
127-18-4	Tetrachloroethylene [Perchloroethylene]
137-26-8	Thiram [Thiuram][Tetramethylthiuram disulfide]
16065-83-1	Chromium III (insoluble salts)
1746-01-6	2,3,7,8-Tetrachlorodibenzo-p-dioxin [2,3,7,8-TCDD]
18540-29-9	Chromium VI
50-32-8	Benzo(a)pyrene
53-70-3	Dibenz[a,h]anthracene
62-53-3	Aniline
67-66-3	Chloroform
71-43-2	Benzene
71-55-6	1,1,1-Trichloroethane [Methyl chloroform]
72-43-5	Methoxychlor
7439-92-1	Lead
7439-97-6	Divalent Mercury
7439-97-6e	Elemental Mercury
7439-97-6m	Methyl Mercury
7440-02-0	Ni+2
7440-22-4	Silver
7440-36-0	Antimony
7440-38-2	Arsenic
7440-39-3	Barium
7440-41-7	Beryllium
7440-43-9	Cadmium
7440-62-2	Vanadium
7440-66-6	Zinc
7446-18-6	Thallium
75-01-4	Vinyl chloride [Chloroethylene][Ethylene chloride]
75-05-8	Acetonitrile [Methyl cyanide]
75-09-2	Methylene chloride [Dichloromethane]
75-15-0	Carbon disulfide
7783-79-1	Se+6
78-93-3	Methyl ethyl ketone [2-Butanone][MEK]
79-01-6	Trichloroethylene
80-62-6	Methyl methacrylate
87-68-3	Hexachloro-1,3-butadiene [Hexachlorobutadiene]
87-86-5	Pentachlorophenol [PCP]
94-75-7	2,4-D [2,4-Dichlorophenoxyacetic acid][2,4-D salts and esters]
98-95-3	Nitrobenzene

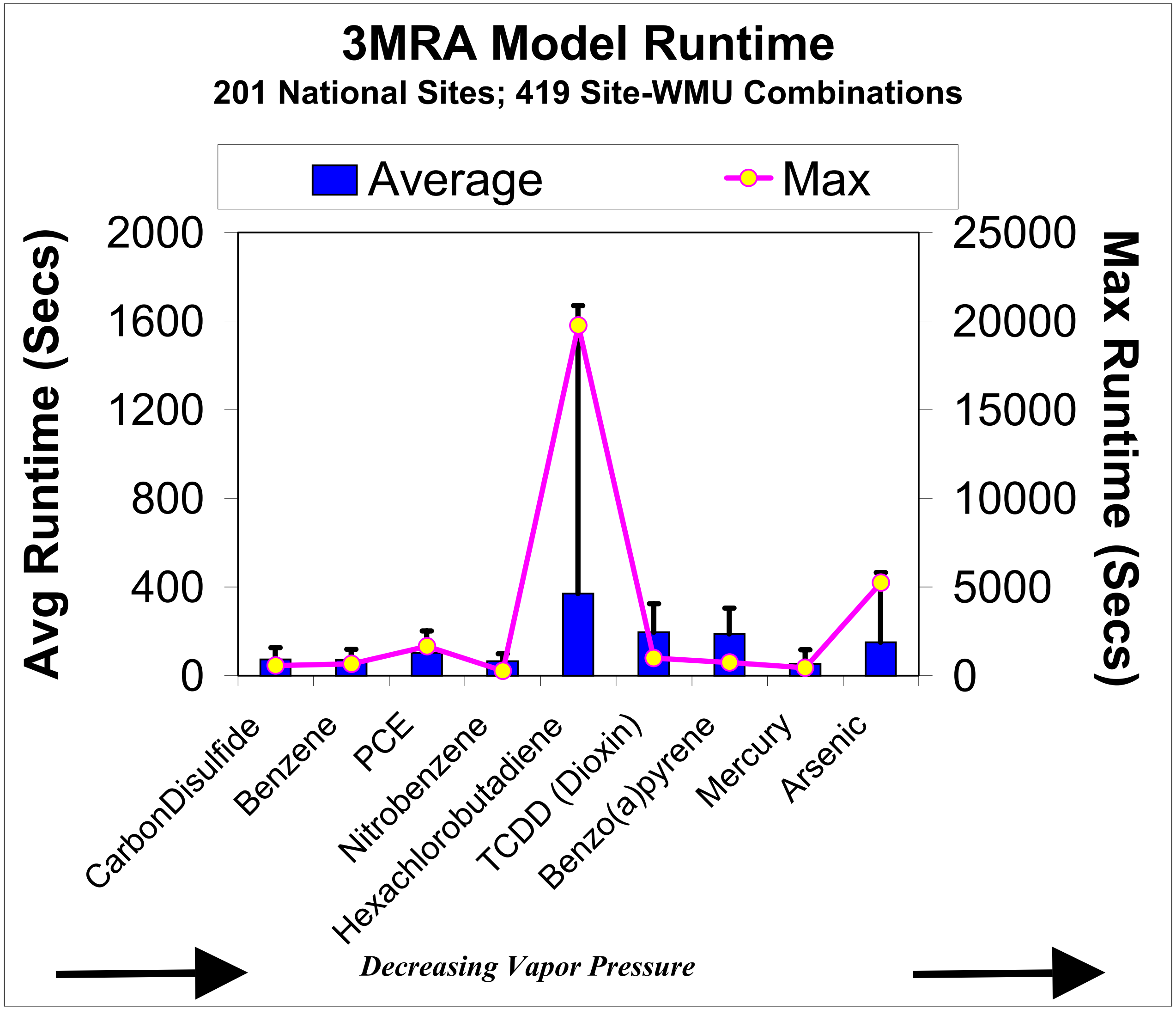


Figure 6-16. Model Runtimes: Initial 3MRA Time Trial Analysis for Selected Chemicals.

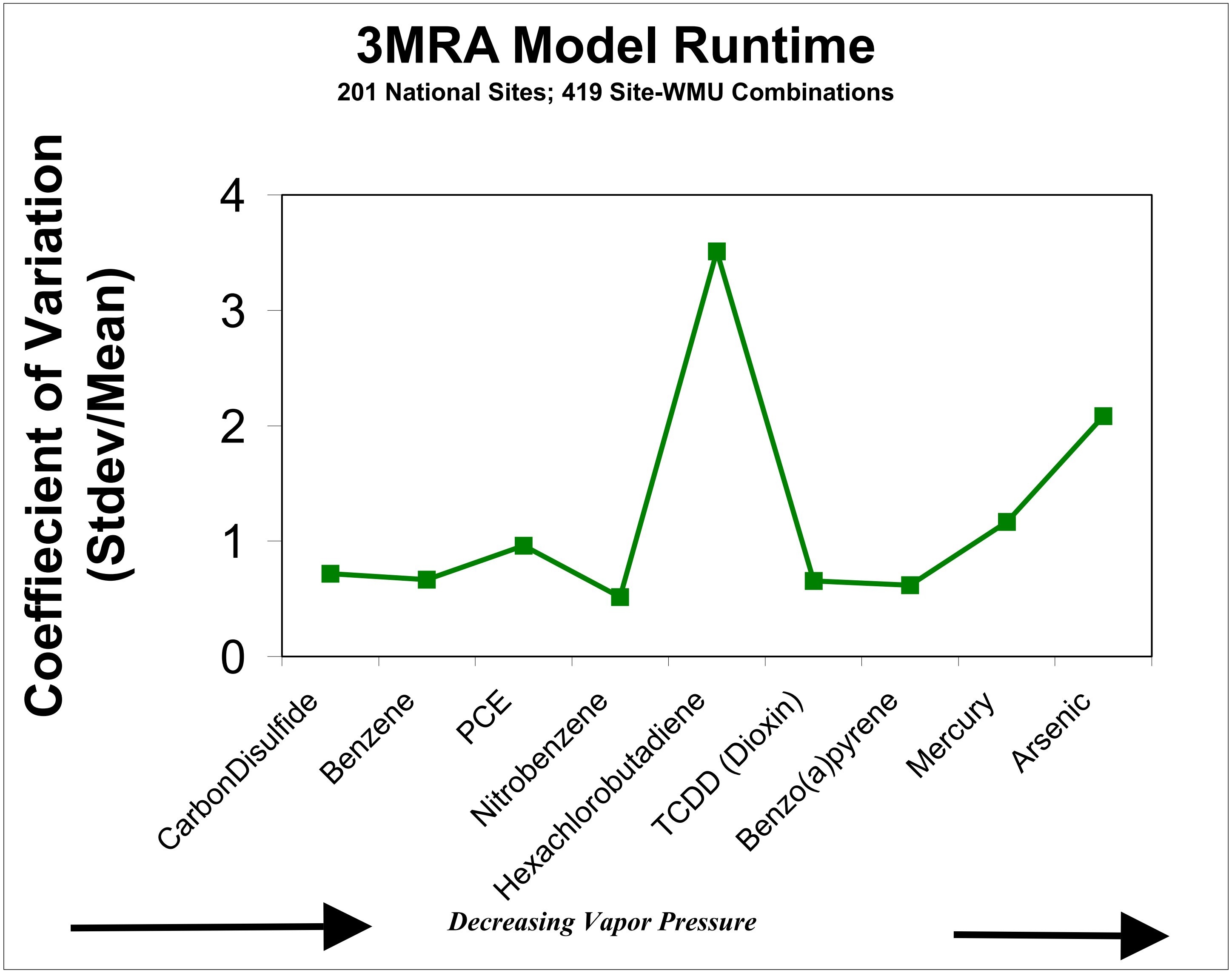


Figure 6-17. Model Runtime CVs: Initial 3MRA Time Trial Analysis for Selected Chemicals.

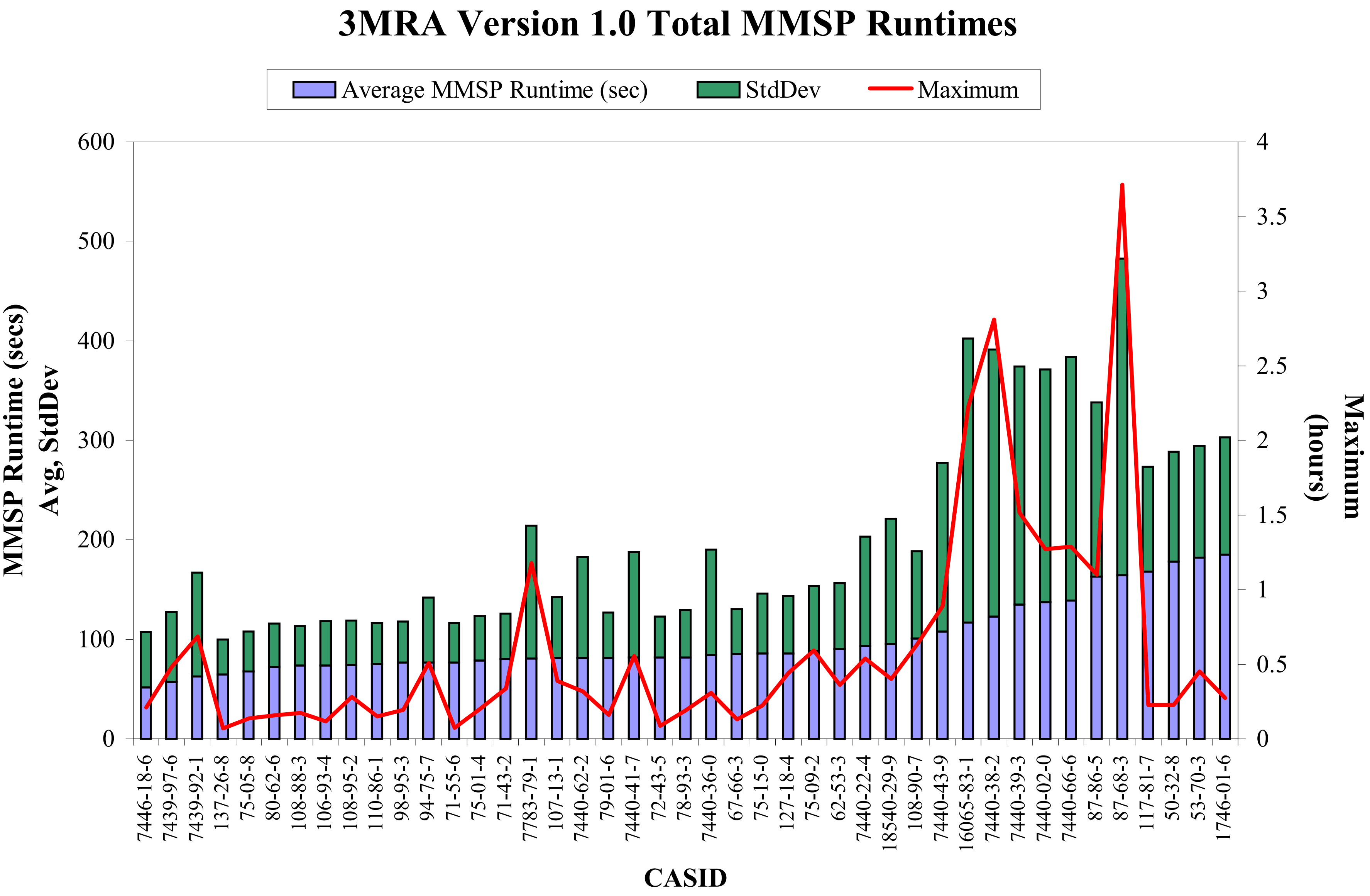


Figure 6-18. 3MRA Version 1.0 Model Runtimes By Chemical (447,284 simulations).

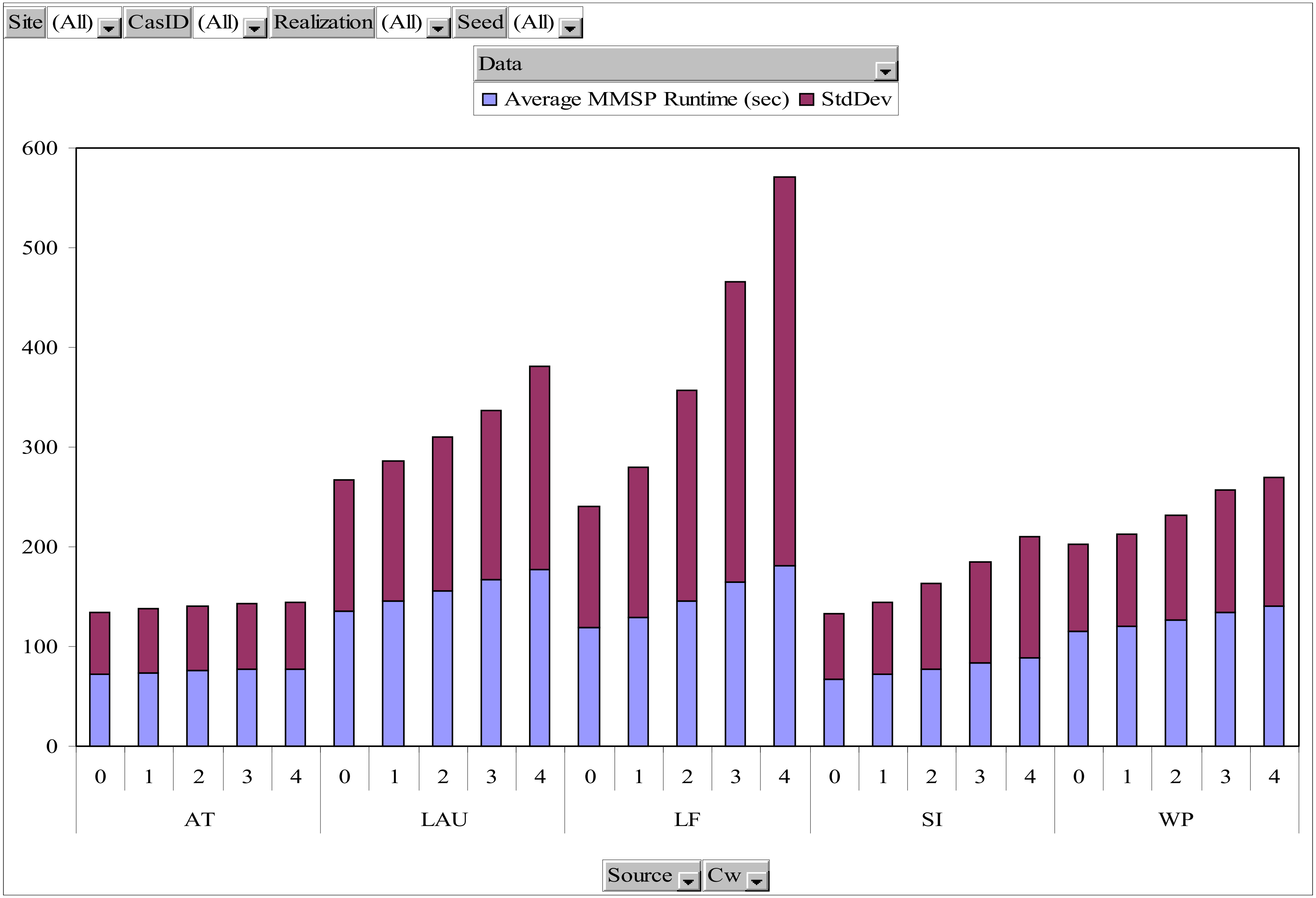


Figure 6-19. 3MRA Version 1.0 Model Runtimes By Source Type (447,284 simulations).

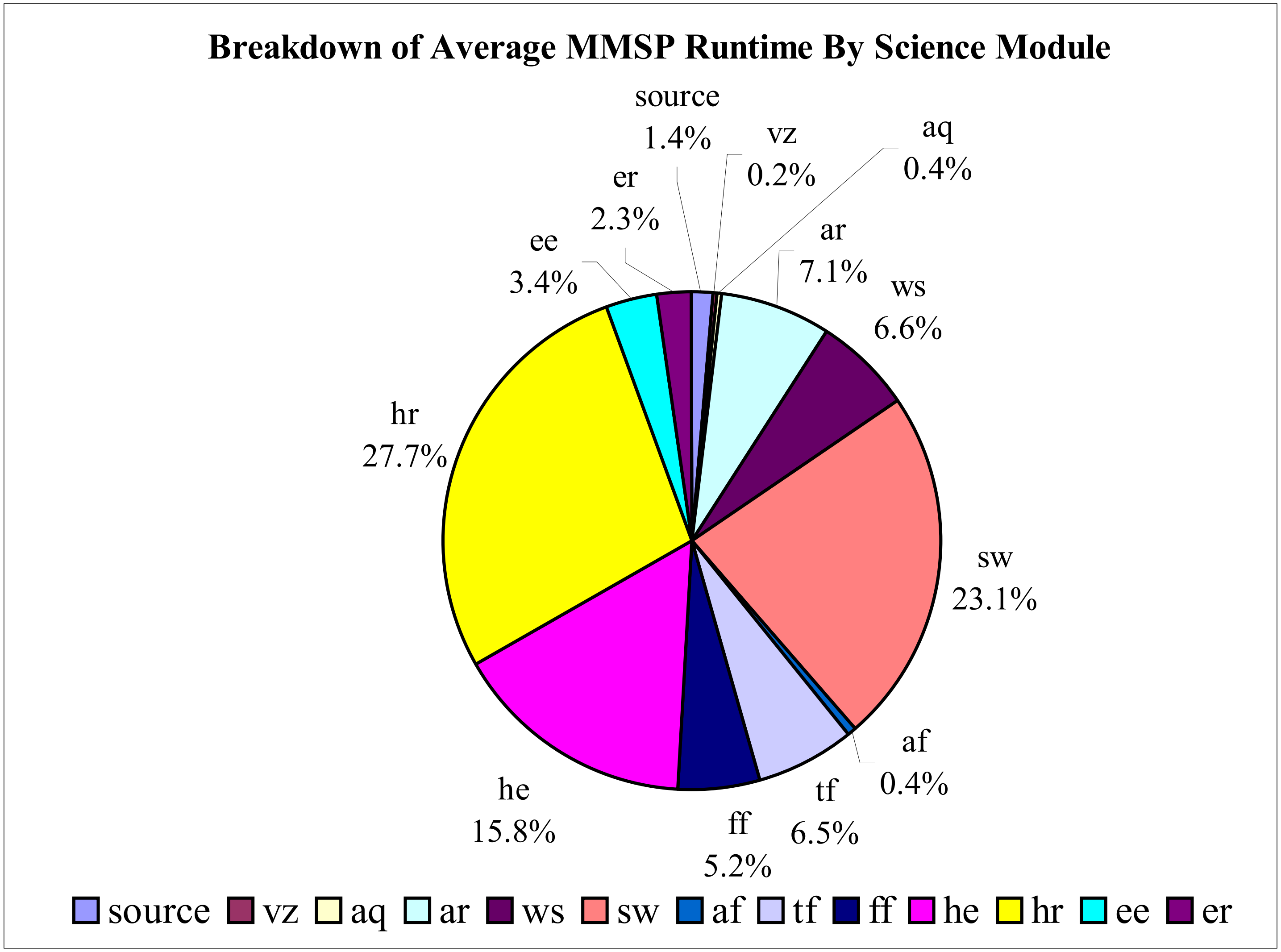


Figure 6-20. Breakdown of Average 3MRA MMSP Runtime by Science Module (447,284 simulations).

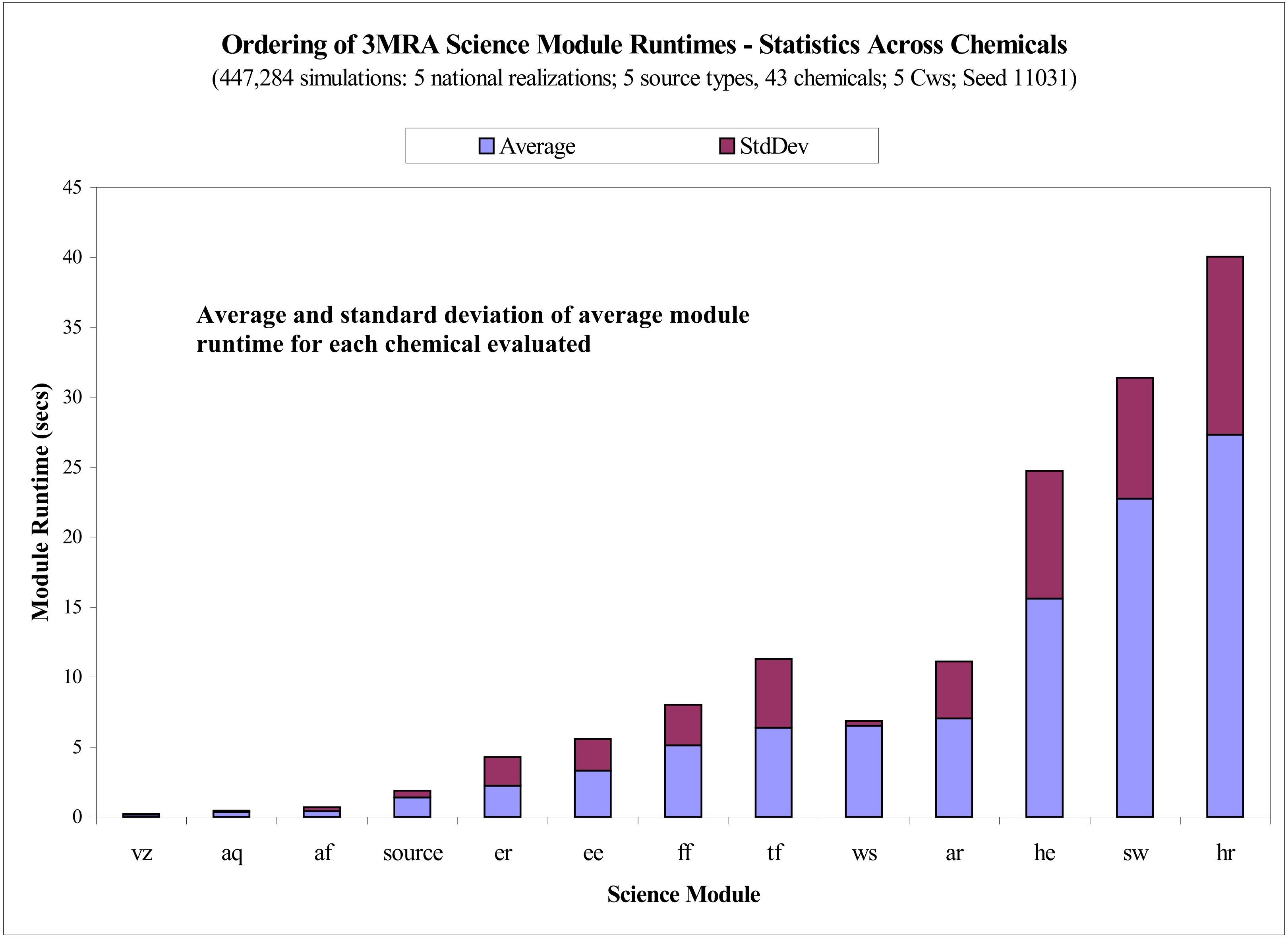


Figure 6-21. Ordering of 3MRA Science Module Runtimes for 43 Chemicals.

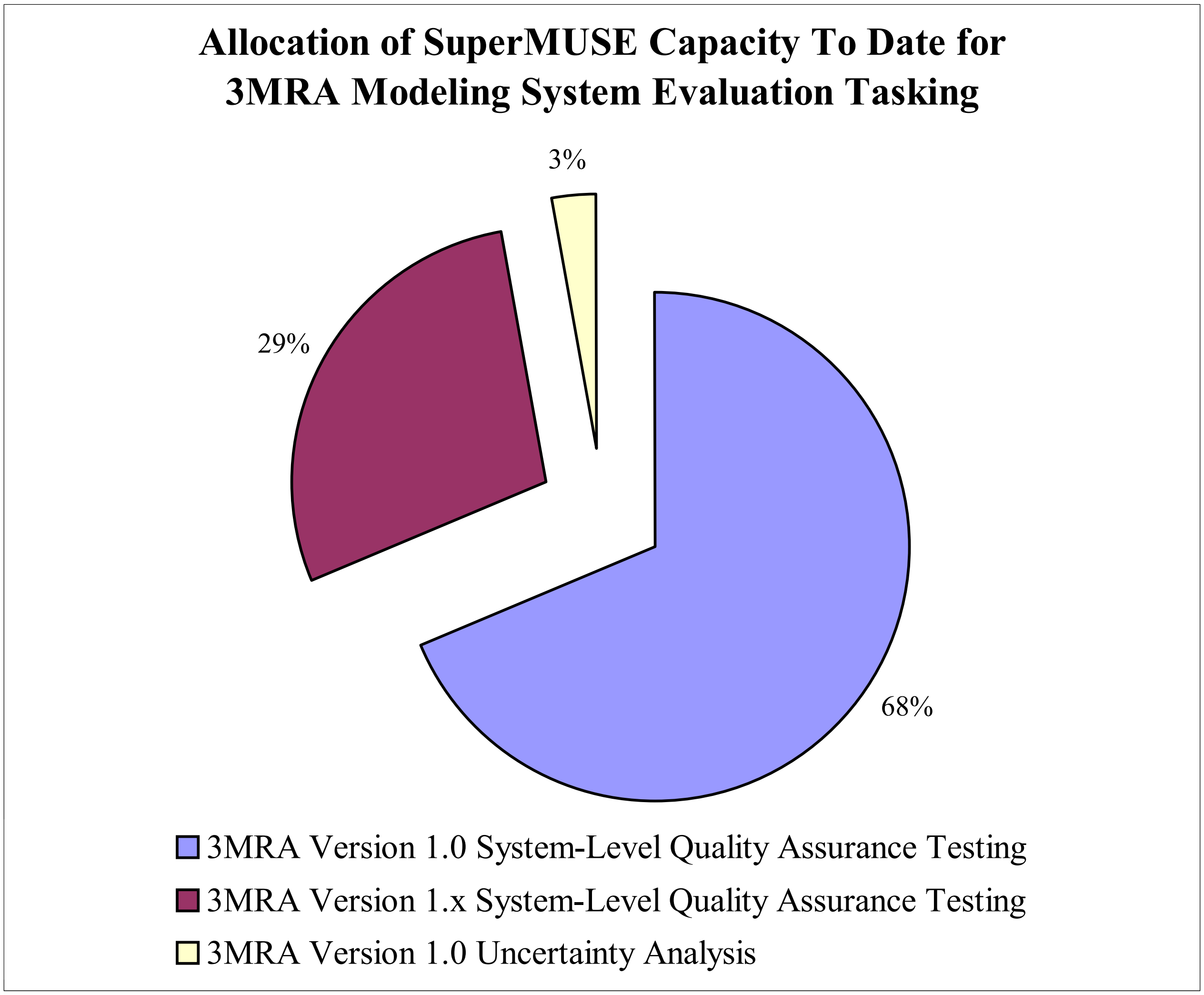


Figure 6-22. Allocation of SuperMUSE Computing Resources To Date.
(Represents over 36 million 3MRA simulations conducted since 2001)