US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

Documentation for the FRAMES-3MRA Technology Software System, Parallel Processor Modifications

K. J. Castleton
J. E. Babendreier
R. Y. Taira
L. K. Williams

May 2003

Prepared for
Office of Research and Development
National Environmental Research Laboratory
U.S. Environmental Protection Agency
and
Office of Solid Waste
Economics, Methods, and Risk Analysis Division
U.S. Environmental Protection Agency
under Contract DE-AC06-76RLO 1830

PNNL-Draft

US EPA ARCHIVE DOCUMENT

DISCLAIMER

This report was prepared as an account of work sponsored by the U.S. Environmental Protection Agency. Neither Battelle Memorial Institute, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RLO 1830

This document was printed on recycled paper.
(9/97)

US EPA ARCHIVE DOCUMENT

Documentation for the FRAMES-3MRA Technology Software System, Parallel Processor Modifications

K. J. Castleton
J.E. Babendreier
R. Y. Taira
L. K. Williams

October 2002

Prepared for
Office of Research and Development
National Environmental Research Laboratory
U.S. Environmental Protection Agency
and
Office of Solid Waste
Economics, Methods, and Risk Analysis Division
U.S. Environmental Protection Agency
under **Contract DE-AC06-76RLO 1830**

Pacific Northwest National Laboratory
Richland, Washington 99352

US EPA ARCHIVE DOCUMENT

Summary

The U.S. Environmental Protection Agency (EPA) is developing a comprehensive environmental exposure and risk analysis software system for agency-wide application. The software system will be applied to the technical assessment of exposures and risks relevant to the Hazardous Waste Identification Rule (HWIR). The software system adapted to automate this assessment is the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES), developed by the Pacific Northwest National Laboratory. The process used to develop the FRAMES-3MRA Technology Software System includes steps for requirements analysis, design, specification, and development with testing and quality assurance composing a critical portion of each step. This report documents that process for the software that was created or modified to implement a parallel distributed processing version of HWIR.

Acronyms and Abbreviations

ASCII	American Standard Code for Information Interchange
DIC	Dictionary File
DLL	dynamic link library
ELP I	Exit Level Processor I
ELP II	Exit Level Processor II
EPA	U.S. Environmental Protection Agency
FRAMES	Framework for Risk Analysis in Multimedia Environmental Systems
GRF	Global Results Files
HWIR	Hazardous Waste Identification Rule
MMSP	Multimedia Multipathway Simulation Processor
OSW	Office of Solid Waste
PNNL	Pacific Northwest National Laboratory
RCRA	Resource Conservation and Recovery Act
SDP	Site Definition Processor
SSF	Site Simulation Files
Stat DLL	Statistical Dynamic Link Library
SUI	System User Interface

US EPA ARCHIVE DOCUMENT

Terminology

CPU Allocator	Matches Tasker Clients with Taskers for the purpose of assigning tasks
Job	A list of tasks to be completed
Process Errors Program	A program used to keep track of which components of an HWIR simulation have succeeded or failed
Site Summary Tool	A program that reads input and output files created by the HWIRIO.dll and summarizes them in tables
SUI Tasker	A specialized Tasker that reads an HWIR header file and generates tasks for the simulations specified in the header file
Task	A component of a job
Tasker	A program that generates a list of jobs to be completed
Tasker Client	A machine that performs a task
Update Client Program	A program that facilitates the copying of files to a large number (hundreds) of machines that make up a cluster

US EPA ARCHIVE DOCUMENT

Contents

Summary iii

Acronyms and Abbreviations iv

Terminology v

Figures viii

Tables viii

1.0 Introduction 1.1

2.0 Testing Approach 2.1

3.0 CPU Allocator 2.1
3.1 Requirements 2.1
3.2 Design 2.1
3.3 Specifications 2.2
3.4 Testing Approach and Results 2.3

4.0 Tasker Client 2.11
4.1 Requirements 2.11
4.2 Design 2.11
4.3 Specifications 2.12
4.4 Testing Approach and Results 2.13

5.0 SUI Tasker 2.20
5.1 Requirements 2.20
5.2 Design 2.21
5.3 Specifications 2.23
5.4 Testing Approach and Results 2.24

6.0 Process Error Program 2.42
6.1 Requirements 2.43
6.2 Design 2.43
6.3 Specifications 2.44
6.4 Testing Approach and Results 2.45

7.0 Update Client Program 2.54
7.1 Requirements 2.55
7.2 Design 2.55
7.3 Specifications 2.56
7.4 Testing Approach and Results 2.57

US EPA ARCHIVE DOCUMENT

8.0 Site Summary Tool 2.63
8.1 Requirements 2.64
8.2 Design 2.64
8.3 Specifications 2.64
8.4 Testing Approach and Results 2.67

9.0 Quality Assurance Program 9.1

10.0 References 10.1

US EPA ARCHIVE DOCUMENT

Figures

1.1 Overview of the FRAMES-3MRA Technology Software System 1.2
5.1 HWIR System User Interface (SUI) 2.20
5.2 SUI Tasker User Interface 2.21
7.1 Update Client User Interface 2.55
9.1 Ensuring Quality in the Environmental Software Development Process 9.2
9.2 Quality Assurance Implementation Checklist for the Module Execution Manager 9.4

Tables

3.1 Fundamental Requirements for Testing the CPU Allocator 2.3
3.2 Relationship Between Test Cases and Fundamental Requirements for the CPU Allocator 2.3
4.1 Fundamental Requirements for Testing the Tasker Client 2.13
4.2 Relationship Between Test Cases and Fundamental Requirements for the Tasker Client 2.13
5.1 Fundamental Requirements for Testing the SUI Tasker 2.24
5.2 Relationship Between Test Cases and Fundamental Requirements for the SUI Tasker 2.24
6.1 Field Definitions for Errors and Warnings Table 2.44
6.2 Fundamental Requirements for Testing the PEP 2.45
6.3 Relationship Between Test Cases and Fundamental Requirements for the PEP 2.46
7.1 Fundamental Requirements for Testing the Update Client Tool 2.57
7.2 Relationship Between Test Cases and Fundamental Requirements for the Update Client Tool . . 2.57
8.1 Fundamental Requirements for Testing the Site Summary Tool 2.67
8.2 Relationship Between Test Cases and Fundamental Requirements for the Site Summary Tool . . 2.67
9.1 Relationship of PNNL Environmental Software Development Process to Quality Assurance Requirements 9.3

US EPA ARCHIVE DOCUMENT

1.0 Introduction

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

The software system adapted to automate this assessment is the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES), developed by the Pacific Northwest National Laboratory (PNNL). The FRAMES-3MRA Technology Software System consists of a series of components within a system framework (Figure 1.1). The process used to develop the FRAMES-3MRA Technology Software System includes steps for requirements analysis, design, specification, and development, with testing and quality assurance composing a critical portion of each step.

This report documents the six processors that were developed in order to implement the HWIR software system in a distributed computing mode. Each processor's requirements, design, specifications, testing plans and results are included as well as a description of the quality assurance program implemented. References cited in the text are listed in Section 10.0. Other components developed by PNNL are described in companion documents, which are listed in the reference list; the system itself is documented in a summary report entitled *Overview of the FRAMES-HWIR Technology Software System*.

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

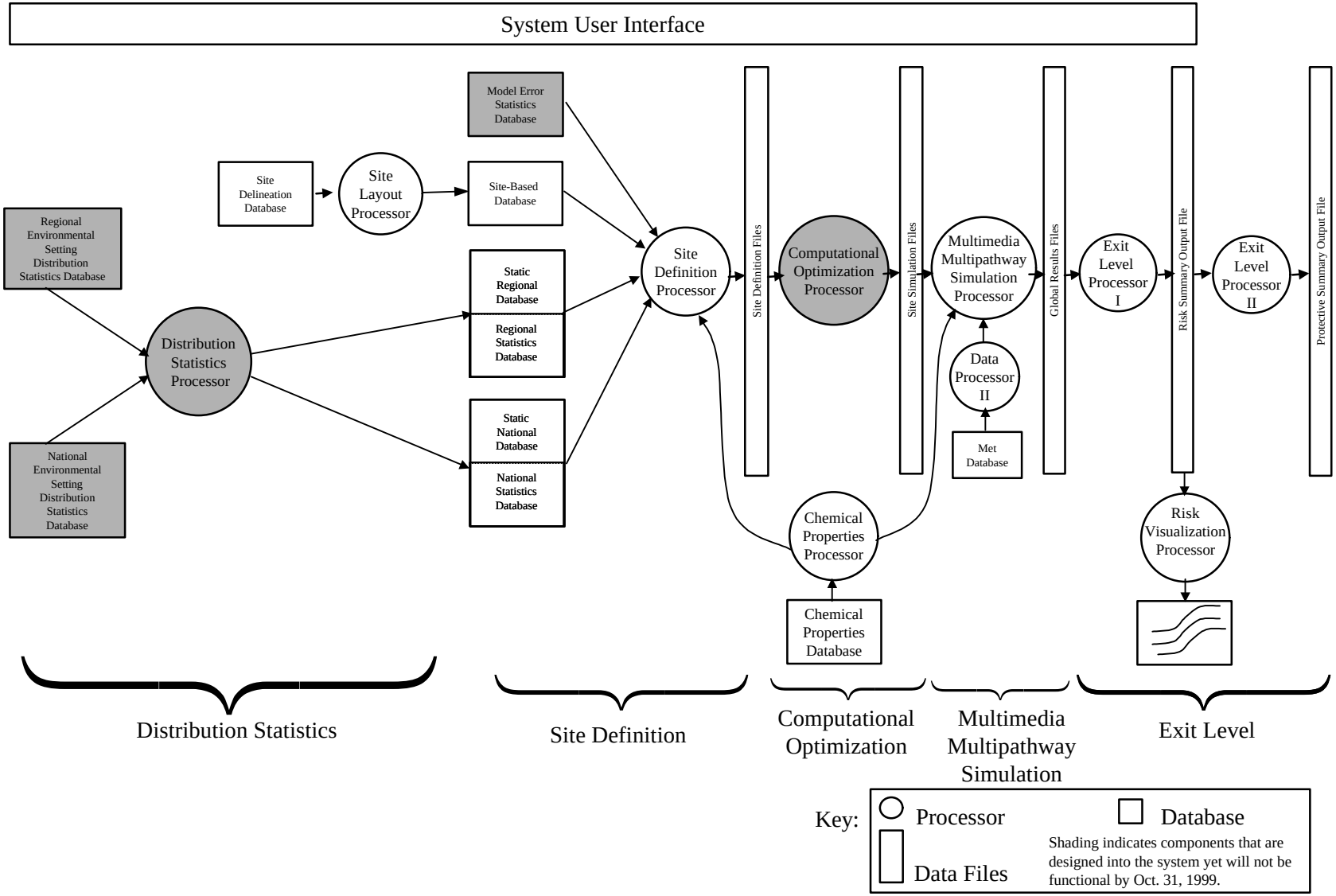

Figure 1.1 Overview of the FRAMES-HWIR Technology Software System

US EPA ARCHIVE DOCUMENT

2.0 Testing Approach

Software can be tested at the unit and system levels. Unit testing evaluates individual components in isolation from other components. System testing evaluates the performance of groups of components functioning together, data communication between the components that make up the system (also called integration testing), and the overall performance of the system. This document addresses unit testing of the main components of the parallel processing version of the HWIR Software System.

All tests are to be conducted under Windows® 98 and Windows® 2000, which represent the required operating systems for implementing the parallel processing version of the HWIR Software System.

3.0 CPU Allocator

A parallel computer system is built to perform a number of computational pieces in parallel, independent of computations on other machines. The CPU Allocator is the primary tool for load balancing, by randomly assigning jobs that are being run on the system to available machines. When a machine finds itself free (it has no tasks scheduled), it calls the CPU Allocator and is given the name of a machine that has a job that needs to be done. A job is a collection of tasks; a task is part of the computation that takes more than a couple of seconds. The CPU Allocator does not discriminate between fast machines or slow machines. The CPU Allocator must thus keep track of the jobs running on the system and which machine(s) they have been assigned to. It can also shut down or reset machines as they become free.

3.1 Requirements

The CPU allocator has several main requirements and they are:

1) To keep track of the jobs running on the multiprocessor system. It should provide: a) the name of the machine that is managing the tasks that are part of a job, b) a brief description of the job, and c) the percentage of machines in the cluster dedicated to the job.
2) Provide option buttons to restart or turn off all the machines in the cluster.
3) Act as a TCP/IP server that should accept Taskers adding and removing jobs
4) Randomly redirect Clients, which state they are free to perform some task, to a Tasker with an available job.
5) Be flexible enough to accommodate different Jobs, Tasks, or Machines without requiring software modifications or recompiling.

3.2 Design

The design of the whole HWIR-Parallel software was intended to be 1) easy to setup and 2) be based only on the TCP/IP network protocol. This reduces the cost of the system because installation is not complicated and no licenses for proprietary software are required. The resulting system is also easy to port to operating systems other than MS-Windows. .

The CPU Allocator is a TCP/IP server that is "listening" on a specified port for messages from machines that are either a) free (available to contribute to a job) or b) in need some resources to perform tasks associated with a job. The CPU Allocator understands three TCP/IP messages:

1) “IAmFree” from machine that is available
2) “AddJob” from a machine that needs computing resources
3) “RemoveJob” from a machine that no longer needs some computing resources

The "IamFree" message is from a machine making itself available. It is handled by randomly choosing from the list of jobs based on the percentile of use for each job and responding to the "IAmFree" machine with the name of the machine with the job. A Machine Name and a Job Name follow the "AddJob" message, from a machine that needs some resource. The machine name and job name are handed to machines that send the "IAmFree" message. "RemoveJob," from a machine that no longer needs some resource, is handled much the same way as "AddJob" except the job is removed from the list of jobs.

In addition to listening for messages from and sending messages to individual machines, the CPU Allocator can also communicate reset and shutdown messages to all machines, as each communicates with it. Both "Reset Clients" and "Shut Down Clients" buttons change the behavior of the CPU Allocator when it receives an "IAmFree" message. When the "Reset Clients" or "Shut Down Clients" button is depressed, the CPU Allocator tells free machines to "Restart" or "Off". (This means that a machine in the system cannot be named "Off" or "Restart" otherwise clients will turn themselves off when assigned to that job.) A list of machines that have been turned off or reset since the button was pressed allows the user to quickly find machines that might not behave as expected.

A screen that shows the current jobs in the system is part of the CPU Allocator User Interface. This allows the user to see how many jobs are in the system.

3.3 Specifications

The CPU Allocator responds to TCP/IP messages from Tasker and TaskerClient machines. The specifications for the four possible dialogs are below. In each case, the dialog is initiated by a machine other than the CPU Allocator. In a real conversation between CPU Allocator, Tasker and/or TaskerClient, the text in <> would be replaced with specific relevant information. Comments are in italics.

3.3.1 The most common dialog
TaskerClient: IAmFree <ClientMachineName>
CPUAllocator: <TaskerMachineName> *Who to contact for tasks*
TaskerClient: Ok

3.3.2 The Restart or Off dialog
TaskerClient: IAmFree <ClientMachineName>
CPUAllocator: Restart *or Off*
TaskerClient: Ok

3.3.3 The “AddJob” dialog
Tasker: AddJob <TaskerMachineName> <JobName>
CPUAllocator: Ok

3.3.4 The “RemoveJob” dialog

US EPA ARCHIVE DOCUMENT

Tasker: RemoveJob <TaskerMachineName> <JobName>
CPUAllocator: Ok

3.4 Testing Approach and Results

Requirements for the CPU Allocator are summarized in Section 3.1 of this document. These requirements were reworded as necessary and listed in Table 3.1. They were stated as concise, fundamental requirements that are testable.

Table 3.1. Fundamental Requirements for Testing the CPU Allocator

Requirement Number	Requirement
1	Keep track of the jobs running on the multiprocessor system. It should provide: a) the name of the machine that is managing the tasks that are part of a job, b) a brief description of the job, and c) the percentage of machines in the cluster dedicated to the job.
2	Provide option buttons to restart or turn off all the machines in the cluster.
3	Act as a TCP/IP server that should accept Taskers adding and removing jobs
4	Randomly redirect Clients, which state they are free to perform some task, to a Tasker with an available job.
5	Be flexible enough to accommodate different Jobs, Tasks, or Machines without requiring software modifications or recompiling.

To ensure that the CPU Allocator meets the requirements listed in Table 3.1, the following test cases were developed to verify the performance of the CPU Allocator. Table 3.2 shows the relationship between these requirements and the test cases, which are described below.

Table 3.2. Relationship Between Test Cases and Fundamental Requirements for the CPU Allocator

		Test Case Number		
		01	02	03
Requirement	1	X	X	
	2			X
	3	X	X	
	4	X	X	
	5		X	

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

3.4.1 CPU01

3.4.1.1 Description

The purpose of this test case is to test the general functionality of the CPU Allocator. The ability of the software to track the available Tasker jobs and the ability to assign available Tasker Clients to Tasker jobs will be checked. In this case, there will be only one Tasker and one Tasker Client. The Tasker job to be done is to fire off the Notepad program two times. After the job is completed the Tasker will be shut down.

3.4.1.2 Input Data

The input data for this test case consists entirely of text written in batch files (*.bat files) used to execute this test case. The contents of the batch files used are listed below followed by a description of the file contents:

cpuAllocator.bat

"c:\Program Files\JavaSoft\JRE\1.3.1_03\bin\javaw.exe" HWIRNet.AllocatorController

File format: {path to Java executable} {CPU Allocator program}

Tasker.bat

"c:\Program Files\JavaSoft\JRE\1.3.1_03\bin\javaw.exe" HWIRNet.BatchTasker WP-TAIRAR MyTasker.bat

File format: {path to Java executable} {BatchTasker program} {computer name} {Tasker job file}

MyTasker.bat

notepad
notepad
end

File format: {execute notepad} {execute notepad} {end}

TaskerClient2.bat

"c:\Program Files\JavaSoft\JRE\1.3.1_03\bin\java.exe" HWIRNet.TaskerClient WP-TAIRAR

File format: {path to Java executable} {TaskerClient program} {computer name}

* Note that the executable paths and computer name are computer specific and will need to be edited by the user for use on his/her computer.

3.4.1.3 Expected Results

US EPA ARCHIVE DOCUMENT

It is expected that the CPU Allocator will recognize the Tasker with available jobs, assign the available TaskerClient to do the jobs, and present through its user interface details describing the job. After the Tasker job is completed and the Tasker is shut down, the details of the Tasker job shown in the CPU Allocator user interface should disappear.

3.4.1.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto the user's computer. The test bed files must also be installed in order to follow this test plan and execute the tests.
2. The first thing you will need to do is to edit three of the *.bat files. As noted in the input section above (Section 3.4.1.2), you will need to change the executable paths and computer name to adjust for your computer.
3. Turn on the CPU Allocator by finding the file called 'cpuAllocator.bat' and double clicking on it. The user interface (UI) for the CPU Allocator should appear on your screen. There are two tabs entitled 'Jobs in system' and 'Machine Reset'. On the 'Jobs in system' tab, there are three header descriptions entitled 'Machine', 'Description', and 'Percentage'. When a Taskers are up and running and have jobs available, the Tasker information will be displayed by row under these headings. At this time, no Taskers are running so the area below the headers should be blank.
4. Turn on the Tasker by finding a file on your computer called 'Tasker.bat' and double clicking on it. Three things should happen: the Tasker user interface should appear on your screen, a DOS window should appear on your screen, and the Tasker information should appear in a row in the CPU Allocator UI. Under the 'Machine' heading, your computer name should appear; under 'Description', 'MyTasker.bat' should appear; and under the 'Percentage' heading, '100.0' should appear. The '100.0' percentage indicates that 100 % of the computers are dedicated to the job. This is true in this case because there is only one job and only one computer.
5. Note that in the Tasker UI there are two rows of information listed below the headings. This indicates that there are two jobs to do for this Tasker. The jobs are unassigned at the moment because the Tasker is waiting for Tasker Clients to become available to complete the jobs.
6. Turn on a Tasker Client by finding a file on your computer called 'TaskerClient2.bat' and double clicking on it. Two things should happen initially: the Tasker Client user interface should appear on your screen and a DOS window should appear on your screen. Also note that the information on the Tasker UI for the first job has changed. The 'Machine' information has changed from 'Unassigned' to your computer's name and the 'Assigned' information has changed from 'Unassigned' to the day, date, and time that the job was initiated. You will then notice that the 'Notepad' program starts up. This is the first job that the Tasker has to complete.
7. Close down 'Notepad' and you should see that the first job disappears from the Tasker UI. Like the information for the first job that was completed, the job information for the second task is now updated. The 'Machine' information has changed from 'Unassigned' to your computer's name and the 'Assigned' information has changed from 'Unassigned' to the day, date, and time that the job was initiated. You will then notice that the 'Notepad' program starts up again. This is also the second job that the Tasker has to complete. Close down 'Notepad' and you should see that the second job disappears from the Tasker UI. Because both jobs that the Tasker had to complete have been completed, the Tasker UI should now be blank under the headings.
8. Close down the Tasker UI. You should see that the information about the Tasker in the CPU Allocator UI has disappeared now too. This indicates that there are currently no Taskers with available jobs.
9. Close down the CPU Allocator UI. This is the end of this test case.

US EPA ARCHIVE DOCUMENT

3.4.1.5 Results

The CPU Allocator recognized the Tasker with available jobs, assigned the available TaskerClient to do the jobs, and presented through its user interface details describing the job. Note that the jobs were completed one at a time because each Tasker Client requires one CPU and there is only one CPU on the computer that was used for testing. After the Tasker jobs were completed and the Tasker was shut down, the details of the Tasker job shown in the CPU Allocator UI disappeared. The CPU Allocator met all of the requirements for this test case.

3.4.2 CPU02

3.4.2.1 Description

The purpose of this test case is to test that the CPU Allocator can accept, track, and manage Taskers and Tasker Clients on more than one computer. In this case, there will be one Tasker and one Tasker Client on one computer and another Tasker and Tasker Client on a separate computer. The Tasker job to be done on each machine is to fire off the Notepad program two times. After the job is completed the Taskers will be shut down.

3.4.2.2 Input Data

The input data for this test case consists entirely of text written in batch files (*.bat files) used to execute this test case. The contents of the batch files used are listed below followed by a description of the file contents:

cpuAllocator.bat

"c:\Program Files\JavaSoft\JRE\1.3.1_03\bin\javaw.exe" HWIRNet.AllocatorController

File format: {path to Java executable} {CPU Allocator program}

Tasker.bat

"c:\Program Files\JavaSoft\JRE\1.3.1_03\bin\javaw.exe" HWIRNet.BatchTasker WP-TAIRAR MyTasker.bat

File format: {path to Java executable} {BatchTasker program} {computer name} {Tasker job file}

MyTasker.bat

notepad
notepad
end

File format: {execute notepad} {execute notepad} {end}

US EPA ARCHIVE DOCUMENT

TaskerClient2.bat

"c:\Program Files\JavaSoft\JRE\1.3.1_03\bin\java.exe" HWIRNet.TaskerClient WP-TAIRAR

File format: {path to Java executable} {TaskerClient program} {computer name}

* Note that the executable paths and computer name are computer specific and will need to be edited by the user for use on his/her computer.

3.4.2.3 Expected Results

It is expected that the CPU Allocator will recognize the Tasker with available jobs on both machines, assign the available TaskerClients to do the jobs, and present through its user interface details describing the jobs. After the Tasker jobs are completed and the Taskers are shut down, the details of the Tasker jobs shown in the CPU Allocator user interface should disappear.

3.4.2.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user's computers. The test bed files must also be installed on both computers in order to follow this test plan and execute the tests.
2. The first thing you will need to do is to edit three of the *.bat files. As noted in the input section (Section 3.4.1.2), you will need to change the executable paths and computer names to adjust for both of your computers.
3. Turn on the CPU Allocator on your main computer by finding the file called 'cpuAllocator.bat' and double clicking on it. The user interface (UI) for the CPU Allocator should appear on your screen. There are two tabs entitled 'Jobs in system' and 'Machine Reset'. On the 'Jobs in system' tab, there are three header descriptions entitled 'Machine', 'Description', and 'Percentage'. When a Taskers are up and running and have jobs available, the Tasker information will be displayed by row under these headings. At this time, no Taskers are running so the area below the headers should be blank.
4. Turn on the Tasker on your main computer by finding a file on your computer called 'Tasker.bat' and double clicking on it. Three things should happen: the Tasker user interface should appear on your screen, a DOS window should appear on your screen, and the Tasker information should appear in a row in the CPU Allocator UI. Under the 'Machine' heading, your computer name should appear; under 'Description', 'MyTasker.bat' should appear; and under the 'Percentage' heading, '100.0' should appear. The '100.0' percentage indicates that 100 % of the computers are dedicated to the job. This is true in this case because there is only one job and only one computer.
5. Turn on the Tasker on the second computer by finding a file on your computer called 'Tasker.bat' and double clicking on it. Three things should happen: the Tasker user interface should appear on that computer's screen, a DOS window should appear on that computer's screen, and the Tasker information should appear in a second row in the CPU Allocator UI. Under the 'Machine' heading, the second computer name should appear; under 'Description', 'MyTasker2.bat' should appear; and under the 'Percentage' heading, '50.0' should appear. Note that the percentage for the first Tasker has changed to '50.0' also. The '50.0' percentage indicates that 50 % of the computers are dedicated to the job. This is true in this case because there are now two jobs and two computers.

US EPA ARCHIVE DOCUMENT

6. Note that in the Tasker UI on the main computer there are two rows of information listed below the headings. This indicates that there are two jobs to do for this Tasker. The jobs are unassigned at the moment because the Tasker is waiting for Tasker Clients to become available to complete the jobs. On the second computer, in the Tasker UI, there is one row of information listed because there is only one job to do.
7. Turn on a Tasker Client by finding a file on your main computer called 'TaskerClient2.bat' and double clicking on it. Two things should happen initially: the Tasker Client user interface should appear on your screen and a DOS window should appear on your screen. Also note that the information on the Tasker UI for the first job has changed. The 'Machine' information has changed from 'Unassigned' to your computer's name and the 'Assigned' information has changed from 'Unassigned' to the day, date, and time that the job was initiated. You will then notice that the 'Notepad' program starts up. This is the first job that the Tasker has to complete.
8. Turn on a Tasker Client on the second computer by finding a file called 'TaskerClient2.bat' and double clicking on it. Two things should happen initially: the Tasker Client user interface should appear on your screen and a DOS window should appear on your screen. Also note that the information on the Tasker UI for the first job has changed. The 'Machine' information has changed from 'Unassigned' to your computer's name and the 'Assigned' information has changed from 'Unassigned' to the day, date, and time that the job was initiated. You will then notice that the 'Notepad' program starts up. This is the first job that the Tasker has to complete.
9. Close down 'Notepad' on the main computer and you should see that the first job disappears from the Tasker UI. Like the information for the first job that was completed, the job information for the second task is now updated. The 'Machine' information has changed from 'Unassigned' to your computer's name and the 'Assigned' information has changed from 'Unassigned' to the day, date, and time that the job was initiated. You will then notice that the 'Notepad' program starts up again. This is also the second job that the Tasker has to complete. Close down 'Notepad' and you should see that the second job disappears from the Tasker UI. Because both jobs that the Tasker had to complete have been completed, the Tasker UI should now be blank under the headings.
10. Close down 'Notepad' on the second computer and you should see that the job disappears from the Tasker UI. There was only one job for this Tasker so the Tasker UI should now be blank under the headings.
11. Close down the Tasker UI on the main computer. You should see that the information about the Tasker in the CPU Allocator UI has disappeared. Next close down the Tasker UI on the second computer. You should see that the information about the Tasker in the CPU Allocator UI has disappeared now too. This indicates that there are currently no Taskers with available jobs.
12. Close down the CPU Allocator UI. This is the end of this test case.

3.4.2.5 Results

The CPU Allocator recognized the Tasker on both computers with available jobs, assigned the available TaskerClients to do the jobs, and presented through its user interface details describing the jobs. Note that the jobs were completed concurrently because each Tasker Client operated on a separate computer. After the Tasker jobs were completed and the Taskers were shut down, the details of the Tasker jobs shown in the CPU Allocator UI disappeared. The CPU Allocator met all of the requirements for this test case.

3.4.3 CPU03

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

3.4.3.1 Description

The CPU Allocator UI has two buttons; one for restarting clients and one for turning off clients. The purpose of this test case is to verify that these two functionalities work and that the CPU Allocator has the ability to turn off or reset all computers in the cluster.

3.4.3.2 Input Data

The input data for this test case consists entirely of text written in batch files (*.bat files) used to execute this test case. The contents of the batch files used are listed below followed by a description of the file contents:

cpuAllocator.bat

"c:\Program Files\JavaSoft\JRE\1.3.1_03\bin\javaw.exe" HWIRNet.AllocatorController

File format: {path to Java executable} {CPU Allocator program}

Tasker.bat

"c:\Program Files\JavaSoft\JRE\1.3.1_03\bin\java.exe" HWIRNet.BatchTasker
PC-Taira.milky-way.battelle.org MyTasker2.bat

File format: {path to Java executable} {BatchTasker program} {computer name} {Tasker job file}

MyTasker2.bat

notepad
end

File format: {execute notepad} {execute notepad} {end}

TaskerClient.bat

cd c:\hwir
"c:\Program Files\JavaSoft\JRE\1.3.1_03\bin\java.exe" HWIRNet.TaskerClient
PC-Taira.milky-way.battelle.org

First line format: {change directory to c:\hwir}
Second line format: {path to Java executable} {TaskerClient program} {computer name}

* Note that the executable paths and computer name are computer specific and will need to be edited by the user for use on his/her computer.

3.4.3.3 Expected Results

It is expected that when the ‘Restart Clients’ button on the CPU Allocator UI is pressed that the computers that are connected to the CPU Allocator will be restarted. It is also expected that when the

'Turn Off Clients' button on the CPU Allocator UI is pressed that the computers that are connected to the CPU Allocator will be turned off.

3.4.3.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user's computers. The test bed files must also be installed on both computers in order to follow this test plan and execute the tests.
2. The first thing you will need to do is to edit three of the *.bat files. As noted in the input section (Section 3.4.1.2), you will need to change the executable paths and computer names to adjust for both of your computers.
3. Turn on the CPU Allocator on your main computer by finding the file called 'cpuAllocator.bat' and double clicking on it. The user interface (UI) for the CPU Allocator should appear on your screen. There are two buttons entitled 'Restart Clients' and 'Turn Off Clients' at the bottom of the UI. These are the two functionalities that will be tested in this test case.
4. Turn on the Tasker on the second computer by finding a file on the computer called 'Tasker.bat' and double clicking on it. Two things should happen: the Tasker user interface should appear on that computer's screen and a DOS window should appear on that computer's screen.
5. Turn on a Tasker Client on the second computer by finding a file called 'TaskerClient2.bat' and double clicking on it. A DOS window should appear on your screen.
6. On your main computer, double click on the 'Restart Clients' button. The second computer should go through the process of closing down and rebooting back up.
7. Once the second computer is back up and running again, repeat steps 4 and 5 above. This time on your main computer, double click on 'Turn Off Clients' button. This time the second computer should go through the process of closing down and should not reboot back up.
8. Close down the CPU Allocator UI. This is the end of this test case.

3.4.3.5 Results

The CPU Allocator was able to restart the networked computer using the 'Restart Clients' button and to turn off the networked computer using the 'Turn off Clients' button. The requirements of this test case were met successfully.

4.0 Tasker Client

The Tasker Client is the workhorse of the parallel software system. It is a generalized batch file execution tool that uses TCP/IP to get the information about 1) the job it should contribute to and 2) the specific task it needs to perform. The task is communicated in a single Unicode Transformation Format (UTF) string that contains the batch file and a number of additional text files. The Tasker Client will be reusable without modification in many different projects.

The communication approach is simple: The Tasker Client makes a client socket connection to the CPU Allocator and sends the message, "IAmFree". The CPU Allocator responds with the name of a machine that has a job that needs some computing resources (the Tasker). The Tasker Client then calls the Tasker and gets the single UTF string that contains the batch and additional files. The Tasker Client copies the files to disk and executes the batch file. When the batch execution finishes, if the batch was successful, the Tasker Client sends a "Done" message to the Tasker; if something failed, it sends a "Failed" message.

US EPA ARCHIVE DOCUMENT

In addition to the main functions that the Tasker Client performs, there are two special messages from the CPU Allocator that the Tasker Client responds to. The CPU Allocator has buttons that allow the user to 1) restart the Tasker Client machine and 2) shut down the Tasker Client machine. When the Tasker Client connects to the CPU Allocator and sends an "IAmFree" message, if either the "restart" or "shutdown" button is depressed, the CPU Allocator replies with "restart" or "off". The Tasker Client then responds by restarting or shutting down its host machine.

4.1 Requirements

The Taker Client has several main requirements and they are:

1) Communicate to the CPU Allocator that it is available to do a job
2) Perform a task, as directed by the CPU Allocator, on the local machine or a networked machine that is connected to the CPU Allocator
3) Provide a status message to the CPU Allocator that communicates when a task is completed successfully and when a task was not successfully completed
4) Reboot when directed by the CPU Allocator
5) Shut down when directed by the CPU Allocator

4.2 Design

The design of the Tasker Client is simple. When it is run the Tasker Client follows these steps:

1) Call into CPU Allocator to get Tasker to work for.
2) Reset or turn off if instructed to do so by the CPU Allocator. (Execution is terminated at this point.)
3) Call into Tasker to get task to perform.
4) If task failed send to Tasker the “Failed” message.
5) Otherwise send to Tasker the “Done” message.
6) Go back to step 1

The Tasker Client communicates with the CPU Allocator and the Tasker on a TCP/IP socket on port 2080 and 2081 respectively. The specifics of the messages are in the Specification section. TCP/IP is used because it reduces the amount of Operating System (OS) dependent software that is needed to run the parallel software.

The software required to execute the batch file sent by the Tasker is expected to be available to the Tasker Client machine. For example, in HWIR each Tasker Client machine has the HWIR system software loaded so that a batch file that contains commands to execute HWIR simulations will function. Without the HWIR software, any attempt to have the Tasker Client run HWIR simulations would fail.

4.3 Specifications

There are a number of message dialogs that can be communicated between the CPU Allocator and the Tasker Client; sections 3.3.1 and 3.3.2 present these messages. Those message dialogs are not repeated here. Three additional message can be communicated between the Tasker Client and the Tasker; they are listed below:

US EPA ARCHIVE DOCUMENT

4.3.1 Getting a task to perform.

TaskerClient: GetTask <ClientMachineName>

Tasker: <BatchFile>#<Number of files>#<File1>#...<FileN>#

The files are written to \\hwir\\ssf\\hdRun1.ssf, \\hwir\\ssf\\hdRun2.ssf, etc... This is a specifically nice choice for HWIR. In the future this may be changed to a more general location but in the current version this is where the files are being written, but currently the batch file could move them to other locations and rename them for other software packages.

4.3.2 Informing the Tasker of a failure

TaskerClient: DoneTask <BatchFile>#<Number of files>#<File1>#...<FileN>#

Tasker: Ok

4.3.3 Informing the Tasker of a successful completion

TaskerClient: ErrorTask <BatchFile>#<Number of files>#<File1>#...<FileN>#

Tasker: Ok

US EPA ARCHIVE DOCUMENT

4.4 Testing Approach and Results

Table 4.1. Fundamental Requirements for Testing the Tasker Client

Requirement Number	Requirement
1	Communicate to the CPU Allocator that it is available to do a job
2	Perform a task, as directed by the CPU Allocator, on the local machine or a networked machine that is connected to the CPU Allocator
3	Provide a status message to the Tasker that communicates when a task is completed successfully and when a task was not successfully completed
4	Reboot when directed by the CPU Allocator
5	Shut down when directed by the CPU Allocator

To ensure that the Tasker Client meets the requirements listed in Table 4.1, the following test cases were developed. Table 4.2 shows the relationship between these requirements and the test cases, which are described below.

Table 4.2. Relationship Between Test Cases and Fundamental Requirements for the Tasker Client

		Test Case Number		
		01	02	03
Requirement	1	X		
	2	X		
	3	X		X
	4		X	
	5		X	

4.4.1 CTASK01

4.4.1.1 Description

The Tasker Client must communicate to the CPU Allocator that it is available to do a job, take directions from the CPU Allocator on what job to do, and provide a status to the CPU Allocator on whether or not the job was completed successfully. All of this functionality was required in the testing of the CPU Allocator and thus was verified previously in test case CPU01. Please see the write-up for test case CPU01 for the description of the testing.

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

4.4.2 CTASK02

4.4.2.1 Description

The Tasker Client should be able to receive instructions from the CPU Allocator which direct it to reboot or shutdown. This functionality was required in the testing of the CPU Allocator and thus was verified previously in test case CPU03. Please see the write-up for test case CPU03 for the description of the testing.

4.4.3 CTASK03

4.4.3.1 Description

The Tasker Client should provide a status to the Tasker indicating that the job it was directed to do was either completed successfully or if it encountered an error. This test case will verify that the correct message is sent when an error is encountered.

The header file included in this case requires that four tasks be completed. The file specifies that there is one site, one source type, one constituent, two Cw bins, and two realizations. Client taskers on two different networked computers will be used to verify that tasks are distributed to different clients.

4.4.3.2 Input Data

The input for this test case is in three files and they are the ‘hdprod2.ssf’, ‘suitasker.bat’, and the ‘run.bat’ files. The contents of the ‘hdprod2.ssf’ file are listed below with the contents of interest in bold.

HDPROD2.SSF

1,
"hdprod.ssf","data group",
62,
"Air",0,"STRING",0,"",
"c:\hwir\AirModel.exe",
"Aquatic",0,"STRING",0,"",
"c:\HWIR\afw.exe",
"Aquifer",0,"STRING",0,"",
"c:\hwir\aquifer1.exe",
"AT",0,"STRING",0,"",
"c:\hwir\AT.bat",
"CASID",0,"STRING",0,"",
"71-43-2",
"ChemCnt",0,"INTEGER",0,"",
1,
"Chems",1,"STRING",0,"",
1,"Benzene",
"COP",0,"STRING",0,"",

"c:\hwir\DummyProc.bat",
"CPDirectory",0,"STRING",0,"",
"C:\HWIR\parallel_version\Software\CPPdata",
"CW",0,"INTEGER",0,"",
0,
"CWCnt",0,"INTEGER",0,"",
2,
"CWs",1,"STRING",0,"",
2,"3","1",
"Date",0,"STRING",0,"",
"09/06/2002",
"Debug",0,"INTEGER",0,"",
0,
"DSP",0,"STRING",0,"",
"c:\hwir\DummyProc.bat",
"EcoExposure",0,"STRING",0,"",
"c:\hwir\EE.exe",
"EcoRisk",0,"STRING",0,"",
"c:\Hwir\EcoRisk5.exe",
"ELP1",0,"STRING",0,"",
"c:\hwirsrc\ELPIMSqlC\ELP1MYSQL.bat",
"ELP2",0,"STRING",0,"",
"c:\hwir\ELP2.exe",
"Farm",0,"STRING",0,"",
"c:\hwir\FarmFood.exe",
"GRFDirectory",0,"STRING",0,"",
"c:\HWIR\parallel_version\Software\GRF",
"HumanExposure",0,"STRING",0,"",
"c:\hwir\exposure.exe",
"HumanRisk",0,"STRING",0,"",
"c:\Hwir\HRord63.exe",
"Lake",0,"STRING",0,"",
"c:\Hwir\examsio.exe",
"LastCw",0,"LOGICAL",0,"",
"T",
"LAU",0,"STRING",0,"",
"c:\hwir\LAU.bat",
"LF",0,"STRING",0,"",
"c:\hwir\LF.bat",
"MemoCnt",0,"INTEGER",0,"",
0,
"MetDir",0,"STRING",0,"",
"c:\hwir\Metdata",
"MMSP",0,"STRING",0,"",
"c:\hwir\MMSP.exe",
"NationDB",0,"STRING",0,"",
"c:\HWIR\parallel_version\software\DATABASE\National4-10.mdb",
"NewChem",0,"LOGICAL",0,"",

US EPA ARCHIVE DOCUMENT

"T",
"NewRel",0,"LOGICAL",0,"",
"T",
"Permanent",0,"STRING",0,"",
"c:\hwir\Permanent",
"PSOFDirectory",0,"STRING",0,"",
"c:\elp2tb\PSOF",
"RealCnt",0,"INTEGER",0,"",
2,
"Realization",0,"INTEGER",0,"",
2,
"RegionDB",0,"STRING",0,"",
"c:\Hwir\parallel_version\software\DATABASE\SiteDB10-02-00.mdb",
"RSOFDirectory",0,"STRING",0,"",
"c:\elp2tb\RSOF",
"SDP",0,"STRING",0,"",
"c:\hwir\sdpbeta2.exe",
"Seed",0,"INTEGER",0,"",
11031,
"SI",0,"STRING",0,"",
"c:\hwir\SI.bat",
"SiteBasedDB",0,"STRING",0,"",
"c:\Hwir\parallel_version\software\DATABASE\smallsite.mdb",
"SiteCnt",0,"INTEGER",0,"",
1,
"SiteId",0,"STRING",0,"",
"0114001",
"Sites",1,"STRING",0,"",
1,"0114001",
"SiteSurveyDB",0,"STRING",0,"",
"",
"Source",0,"STRING",0,"",
"WP",
"SrcCnt",0,"INTEGER",0,"",
1,
"Srcs",1,"STRING",0,"",
1,"WP",
"SSFDirectory",0,"STRING",0,"",
"c:\HWIR\parallel_version\Software\SSF",
"StaticNationDB",0,"STRING",0,"",
"c:\HWIR\parallel_version\software\DATABASE\National4-10.mdb",
"StaticRegionDB",0,"STRING",0,"",
"c:\Hwir\parallel_version\software\DATABASE\SiteDB10-02-00.mdb",
"StopOnError",0,"INTEGER",0,"",
0,
"StopOnWarning",0,"INTEGER",0,"",
0,
"StorageLevel",0,"INTEGER",0,"",

US EPA ARCHIVE DOCUMENT

0,
"Stream",0,"STRING",0,"",
"c:\Hwir\examsio.exe",
"Terrestrial",0,"STRING",0,"",
"c:\hwir\TF.exe",
"Time",0,"STRING",0,"",
"11:36am",
"Vadose",0,"STRING",0,"",
"c:\hwir\vadose.exe",
"WaterShed",0,"STRING",0,"",
"c:\hwir\ws.bat",
"WP",0,"STRING",0,"",
"c:\hwir\WP.bat",

The contents of the 'suitasker.bat' file will need to be edited to run on the tester's computer. The contents are listed below with an explanation of what changes need to be made.

SUITASKER.BAT

"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" HWIRNet.SUITasker WP-TAIRAR C:\HWIR\parallel_version\Software\ssf hdprod.ssf Test1 c: c: WP-TAIRAR 200
pause

The first line is as follows:

"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" HWIRNet.SUITasker WP-TAIRAR C:\HWIR\parallel_version\Software\ssf hdprod.ssf Test1 c: c: WP-TAIRAR 200

The following is a description of each item in the first line:

a) "c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" - The location of java on the tester's computer. Note that this should be changed to reflect the location of java on the tester's computer.

b) HWIRNet.SUITasker - The SUI Tasker executable. This does not need to be changed.

c) WP-TAIRAR - The computer name where the SUI Tasker resides. This should be changed to reflect the name of the tester's computer.

d) C:\HWIR\parallel_version\Software\ssf - The folder location of the SSF file. This should be changed to reflect the location of the folder on the tester's computer.

e) hdprod.ssf - The name of the HWIR header file. This does not need to be changed.

f) Test1 - The name of current software study being performed. This does not need to be changed.

g) c: - The directory where hwir executables are located on server as specified in the header file. This does not need to be changed.

US EPA ARCHIVE DOCUMENT

h) c: - The directory where hwir executables are located on client machines. This does not need to be changed.

i) WP-TAIRAR - The name or IP address where mysql is running. This should be changed to reflect the name of the tester's computer.

j) 200 - The maximum number of tasks allowed in the queue. This does not need to be changed.

The second line contains the word 'pause' and does not need to be changed.

The contents of the 'run.bat' file are listed below and will be modified for this test case.

RUN.BAT

notepad
del c:\parerror.txt

4.4.3.3 Expected Results

It is expected that the SUI Tasker will read the HWIR header file, hdprod.ssf, and set up the four tasks to be run. The tasks should then be run on both computers that are available. When a task is initiated by a client tasker, the program 'Notepad.exe' should start up. To complete the task, the Notepad program should be shut down manually. The tasks run on the primary computer should end with an error due to an adjustment that is made to the 'run.bat' file on the primary computer. When the four tasks are completed, the SUI Tasker should list two as completed successfully and two as errors.

4.4.3.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user's computers. The test bed files must also be installed on both computers in order to follow this test plan and execute the tests.
2. The first thing you will need to do is locate the SUITasker.bat file on your computer and edit it. As noted previously, you will need to change the executable paths, folders, and computer name to adjust for your computer.
3. After the SUITasker.bat file has been edited and saved, you will next start up the required software components.
4. This test case utilizes two computers. The computer with the SUITasker executable on it will be referred to as the primary computer. The computer that is linked via the network and used as an additional resource to complete tasks will be referred to as the second computer. Both computers should have the entire HWIR software and test package loaded on them.
5. Check that the primary computer has the cpuallocator executable 'cpuallocator.bat', suitasker executable 'suitasker.bat', taskerclient executable 'taskerclient.bat', the header file 'hdprod2.ssf', and the run.bat file.
6. You will need to switch the names of two files such that 'hdprod2.ssf' can be used. Using Windows Explorer change the name of the current 'hdprod.ssf' to 'hdprod1.ssf'.
7. Then change the name of the file 'hdprod2.ssf' to 'hdprod.ssf'.
8. Check that the secondary computer has the 'taskerclient.bat' and the run.bat file.

US EPA ARCHIVE DOCUMENT

9. On the primary computer, open up the ‘run.bat’ file in a text editor.
10. On the second line enter the word ‘rem’ and then a space at the start of the line so that it reads ‘rem del c:\parerror.txt’. Remarking out this line will leave the file ‘parerror.txt’ on the computer. The presence of this file indicates to the SUI Tasker that there was an error.
11. On the primary computer, start up the cpuallocator by double clicking on the file ‘cpuallocator.bat’. Then start up the client tasker by double clicking on the file ‘clienttasker.bat’.
12. Start up the client tasker on the second computer by double clicking on the file ‘clienttasker.bat’.
13. Then start up the SUI Tasker by double clicking on the file ‘suitasker.bat’.
14. The ‘Tasks’ tab should immediately fill up with the list of tasks to be completed. Note that the bottom of the user interface (UI), statistics on the runs are shown. These statistics are ‘# Complete’, ‘Average Time’, ‘# Errors’, and ‘Queued Runs’. The ‘Queued Runs’ category should show the number 4, which is the total number of tasks to be run.
15. You will next note that the first two tasks in the list have been assigned. One should have been assigned to the primary computer and another to the second computer. You should also note that the Notepad program is up on both computers. Executing the Notepad program is the task that is to be completed for all tasks. The task is completed when the Notepad program is closed down by the user. Close down Notepad on both computers. Recall that the task running on the main computer should produce an error because the ‘parerror.txt’ file is not being deleted. You will note that the two tasks are removed from the ‘Tasks’ tab, the ‘# Complete’ counter goes to 1, the ‘# Errors’ counter goes to 1, and the ‘Queued Runs’ counter goes to 2.
16. The next two tasks are then assigned one to each computer again and Notepad is started on both computers. Close down Notepad and repeat this process. All tasks should now be completed.
17. When all tasks have been completed, the ‘# Complete’ counter should read 2, the ‘# Errors’ counter should read 2, and the ‘Queued Runs’ counter should read 0.
18. Click on the ‘Failed Tasks’ tab to view the tasks that failed. Also listed is the computer that the task failed on.
19. Close down the SUI Tasker, CPU Allocator, and Client Tasker on both computers.
20. This completes this test case.

4.4.3.5 Results

The SUI Tasker read the tasks to be completed from the HWIR header file, distributed tasks to the two available client taskers, and tracked their successful completions or errors. This test case was completed successfully.

5.0 SUI Tasker

In the parallel software system a Tasker is any program that generates tasks that need to be performed and registers itself with the CPU Allocator. The SUI Tasker is a specialized Tasker that reads an HWIR header file and generates tasks for the simulations specified in the header file. An HWIR header file can imply looping across many Sites, Sources, Chemicals, Waste Levels, and Monte Carlo Realizations. A description of each of these indices is available in the HWIR System Software documentation. The user using the HWIR software system System User Interface (SUI) may choose a set of simulations to be performed. Figure 5.1 shows the SUI where the user has chosen two sites , two chemicals, five source types and three Waste Levels. If this was for one Monte Carlo realization this could imply as many as 60 simulations. The SUI Tasker’s job is to generate a task set for these simulations and run the simulations on the parallel system. In the single processor version of HWIR the

US EPA ARCHIVE DOCUMENT

SUI itself invokes the simulations one at a time (serially) on the computer the SUI is running on. See the HWIR system software documentation for more details on the SUI.

A Tasker in general is a TCP/IP server that is waiting for client machines to be directed to the Tasker by the CPU Allocator. When a Tasker Client machine announces that it is free to perform some task, the Tasker gives the Tasker Client a description of the task to perform. The Tasker then expects at some time in the future for that same client to come back and state that the task was completed successfully or that it failed. The Tasker also has the requirement to register itself with the CPU Allocator and unregister when it is completed.

5.1 Requirements

The SUI Tasker has several main requirements and they are:

1) Read HWIR header files
2) Generate a task set for simulations
3) Distribute tasks to clients
4) Track tasks and reassign if tasks take too long to complete
5) Keep track of which tasks fail and which machines they fail on

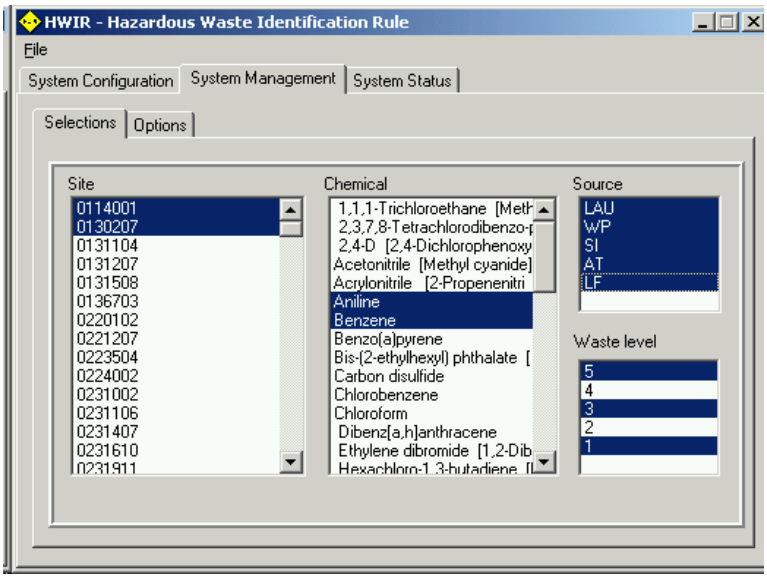

Figure 5.1 HWIR System User Interface (SUI)

5.2 Design

The SUI Tasker is a specialized Tasker, from which it inherits most of its functionality. As a sub-class of Tasker, it has all the behaviors of Tasker, plus additional functionality specific to SUI Tasker. It inherits from Tasker the ability to register with the CPU Allocator, maintain the Task list and respond to Tasker Clients. The SUI Tasker-specific code handles reading the HWIR header file and adding tasks to the task list. All this functionality is seamless to the user of the SUI Tasker.

US EPA ARCHIVE DOCUMENT

When the SUI Tasker is invoked it registers with the CPU Allocator. Then it reads the HWIR header file and loops for each Realization, Site, Source, Chemical, and Waste Level, generating a task for each one. Each task consists of two distinct data items 1) the batch file that will be run by the Tasker Client and 2) the modified header file for the particular Realization, Site, Source, Chemical, and Waste Level. Next, the task is added to the SUI Tasker user interface. Figure 5.2 shows the SUI Tasker interface that is running the selections made in Figure 1.

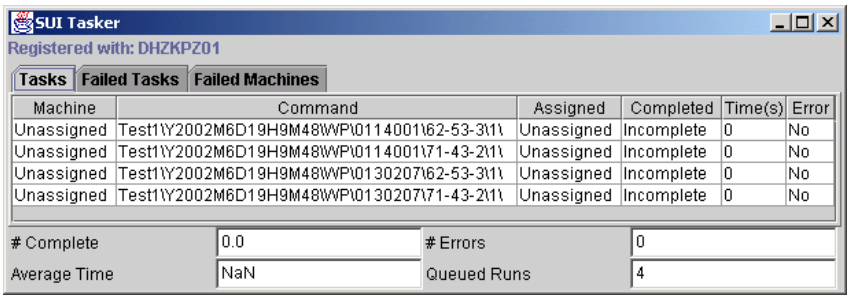

Figure 5.2 SUI Tasker User Interface

Parts of the SUI Tasker User Interface: The Machine column lists which machine in the system took the task. Command is a description of the task being performed. The Assigned column will have a time/date when the task is assigned. Completed shows when the task is completed. Time(s) displays how long the task took in seconds. The Error column displays whether an error occurred during the execution of the task. The lower part of the screen displays running averages and totals for the entire list of tasks. #Complete displays how many tasks are complete. #Errors does the same for number of tasks that failed. Average time is the average time for the tasks. Queued Runs show how many runs are waiting for a Tasker Client to become free. Currently the software maintains the queue at 200 tasks. If there are fewer than 200 tasks in the queue and additional tasks to perform, it will add tasks to the queue until there are 200. To ease the burden of discovering why some tasks failed, two tabs are provided: The Failed Tasks tab displays the same columns as above, but only for tasks that have failed. The Failed Machines tab displays a spreadsheet of the names of machines that have failed. (Machine names must be unique.). Since it is possible for a task to be taken by a Tasker Client and then never return the Complete/Error status, the Tasker itself tracks how long a task has taken. If that time is longer than 48 hours, the Tasker assumes the client is not going to come back and resets the task to unassigned. As a result, the task can be reassigned, to the next Tasker Client that becomes available. If a Tasker Client announces that it is free and the Tasker had assigned that same Tasker Client a task that should have returned a Complete/Error status, the Tasker assumes the task needs to be reset to unassigned as well. These two pieces of functionality handle the case where 1) the Tasker Client is turned off or locks up and 2) the Tasker Client is rebooted.

5.2.1 SUI Tasker Calling Arguments

The SUI Tasker is invoked with the following arguments:
CPU Allocator Name Name or IP address of CPU Allocator Server
ssf directory Path to the SSF directory
header Filename of the header file in ssf directory
name of study Name of current software study you are performing. This will be the name of the database used in the MySQL server.

US EPA ARCHIVE DOCUMENT

server hwir path Directory where hwir executables are located on server as specified in the header file. This will be changed to the value of the next argument
client hwir path Directory where hwir executables are located on client machines.
mySQL server Name or IP address where MySQL is running
queue size Maximum number of tasks allowed in the queue
timeout file filename Name of the file that explicitly lists the time to wait on tasks by chemical

It is important to understand how to use the server hwir path and client hwir path arguments, which allow for HWIR to be installed in one location on the machine running the SUI Tasker and in a different location on a client machine. Basically if a path in the header file is "d:\version1\hwir\ssf" for example the SUI Tasker will change it to "c:\hwir\ssf" if you passed "d:\version1" and "c:" as the fifth and sixth argument. If the user wants to leave these paths unchanged merely passing "d:\version1" and "d:\version1" would leave it essentially unchanged. For simplicity's sake, if the path to the HWIR directory varies across the cluster of machines, instead of attempting to pass the proper client hwir path to the SUI Tasker, the user should, for each machine in the cluster, share the drive where HWIR is installed, and map the drive to the same drive letter. For example: Machine A has hwir installed at "d:\version1\hwir" and a Machine B has hwir installed at "c:\hwir". Have Machine A share the "d:\version1\hwir" directory as "hwir". Have Machine B share the "c:\hwir" directory as "hwir". Now on both machines map the share "hwir" to "Z:". The SUI Tasker could now be called with "d:\version1\hwir" and "Z:" as the fifth and sixth arguments and it should work properly.

The mySQL server and study names are passed through the batch file to the client. These two pieces of information are expected to be used by the client to post data to a database and server specified by the SUI Tasker.

When assigning a job to a Tasker Client, the SUI Tasker passes the information that the Tasker Client requires in a batch file called RunAll.bat. The SUI Tasker puts in this batch file a number of calls to "run.bat," a batch file which is expected to be created and tested by the user of the system, to insure that it invokes a simulation of HWIR correctly. The "run.bat" batch file is passed five arguments. First, the ssf and grf paths, which are the relative paths of "ssf" (the SSF directory) and "grf" (the GRF directory). The header file is passed next. The header file name created by the Tasker Client is "hdRun#.ssf" where # is an integer from 0 to 9. For example, an "hdRun1.ssf" header file would be created in the SSF directory. The last two arguments passed to "run.bat" are the mySQL server and the study name. When the task is complete, the Tasker Client will post results to the mySQL database server, to a database called the study name.

5.2.2Reading and writing the HWIR header file: The HWIRIO.dll

The HWIRIO.dll is used to read and write parameters to the HWIR header file, which is used by the SUI Tasker. The HWIRCP.dll is also used to read the chemical properties for the selected chemicals. If, when the HWIRIO.dll is reading the header file, it reads a parameter out of bounds or incorrectly, an Error file is generated. It writes the Error file to the directory specified in the command line arguments.

5.3 Specifications

The specifications for the SUI Tasker are covered in Section 3.3 and 4.3 but are repeated here for convenience, as SUI Tasker inherits from Tasker its ability to respond to the CPU Allocator and the Tasker Client.

US EPA ARCHIVE DOCUMENT

5.3.1 The “AddJob” dialog

Tasker: AddJob <TaskerMachineName> <JobName>

CPUAllocator: Ok

5.3.2 The “RemoveJob” dialog

Tasker: RemoveJob <TaskerMachineName> <JobName>

CPUAllocator: Ok

5.3.3 Getting a task to perform.

TaskerClient: GetTask <ClientMachineName>

Tasker: <BatchFile>#<Number of files>#<File1>#...<FileN>#

The files are written to \\hwir\\ssf\\hdRun1.ssf, \\hwir\\ssf\\hdRun2.ssf, etc.. This is a specifically nice choice for HWIR. In the future this may be changed to a more general location but in the current version this is where the files are being written, but currently the batch file could move them to other locations and rename them for other software packages.

5.3.4 Informing the Tasker of a failure

TaskerClient: DoneTask <BatchFile>#<Number of files>#<File1>#...<FileN>#

Tasker: Ok

5.3.5 Informing the Tasker of a successful completion

TaskerClient: ErrorTask <BatchFile>#<Number of files>#<File1>#...<FileN>#

Tasker: Ok

The other major specification that the SUI Tasker uses is the HWIR Software System header file. This file is described in the SUI Documentation as well as the HWIR Software System Specifications.

US EPA ARCHIVE DOCUMENT

5.4 Testing Approach and Results

Table 5.1. Fundamental Requirements for Testing the SUI Tasker

Requirement Number	Requirement
1	Read HWIR header files
2	Generate a task set for simulations
3	Distribute tasks to clients
4	Track tasks and reassign if tasks take too long to complete
5	Keep track of which tasks fail and which machines they fail on

To ensure that the SDP meets the requirements listed in Table 4.1, the following test cases were developed to verify the performance of the SDP. Table 4.2 shows the relationship between these requirements and the test cases, which are described below.

Table 5.2. Relationship Between Test Cases and Fundamental Requirements for the SUI Tasker

		Test Case Number		
		01	02	03
	1	X	X	
	2	X	X	
	3	X	X	
	4	X		X
	5		X	

5.4.1 SUITasker01

5.4.1.1 Description

This test case tests the basic functionality of the SUI Tasker. The SUI Tasker should: 1) read HWIR header files, 2) generate a task set for simulations, 3) distribute tasks to clients, and 4) track tasks and reassign if tasks take too long to complete.

The header file in this case requires that twenty four tasks be completed. The file specifies that there are two sites, two source types, two constituents, two Cw bins, and two realizations. Client taskers on two different networked computers will be used to verify that tasks are distributed to different clients.

5.4.1.2 Input Data

US EPA ARCHIVE DOCUMENT

The input for this test case is in three files and they are the ‘hdprod.ssf’, ‘suitasker.bat’, and the ‘run.bat’ files. The contents of the ‘hdprod.ssf’ file are listed below with the contents of interest in bold.

HDPROD.SSF

1,
"hdprod.ssf","data group",
62,
"Air",0,"STRING",0,"",
"c:\hwir\AirModel.exe",
"Aquatic",0,"STRING",0,"",
"c:\HWIR\afw.exe",
"Aquifer",0,"STRING",0,"",
"c:\hwir\aquifer1.exe",
"AT",0,"STRING",0,"",
"c:\hwir\AT.bat",
"CASID",0,"STRING",0,"",
"7440-38-2",
"ChemCnt",0,"INTEGER",0,"",
2,
"Chems",1,"STRING",0,"",
2,"Benzene","Arsenic",
"COP",0,"STRING",0,"",
"c:\hwir\DummyProc.bat",
"CPDirectory",0,"STRING",0,"",
"C:\HWIR\parallel_version\Software\CPPdata",
"CW",0,"INTEGER",0,"",
0,
"CWCnt",0,"INTEGER",0,"",
2,
"CWs",1,"STRING",0,"",
2,"3","1",
"Date",0,"STRING",0,"",
"09/06/2002",
"Debug",0,"INTEGER",0,"",
0,
"DSP",0,"STRING",0,"",
"c:\hwir\DummyProc.bat",
"EcoExposure",0,"STRING",0,"",
"c:\hwir\EE.exe",
"EcoRisk",0,"STRING",0,"",
"c:\Hwir\EcoRisk5.exe",
"ELP1",0,"STRING",0,"",
"c:\hwirsrc\ELPIMSqlC\ELP1MYSQL.bat",
"ELP2",0,"STRING",0,"",
"c:\hwir\ELP2.exe",
"Farm",0,"STRING",0,"",
"c:\hwir\FarmFood.exe",

US EPA ARCHIVE DOCUMENT

"GRFDirectory",0,"STRING",0,"",
"c:\HWIR\parallel_version\Software\GRF",
"HumanExposure",0,"STRING",0,"",
"c:\hwir\exposure.exe",
"HumanRisk",0,"STRING",0,"",
"c:\Hwir\HRord63.exe",
"Lake",0,"STRING",0,"",
"c:\Hwir\examsio.exe",
"LastCw",0,"LOGICAL",0,"",
"T",
"LAU",0,"STRING",0,"",
"c:\hwir\LAU.bat",
"LF",0,"STRING",0,"",
"c:\hwir\LF.bat",
"MemoCnt",0,"INTEGER",0,"",
0,
"MetDir",0,"STRING",0,"",
"c:\hwir\Metdata",
"MMSP",0,"STRING",0,"",
"c:\hwir\MMSP.exe",
"NationDB",0,"STRING",0,"",
"c:\HWIR\parallel_version\software\DATABASE\National4-10.mdb",
"NewChem",0,"LOGICAL",0,"",
"T",
"NewRel",0,"LOGICAL",0,"",
"T",
"Permanent",0,"STRING",0,"",
"c:\hwir\Permanent",
"PSOFDirectory",0,"STRING",0,"",
"c:\elp2tb\PSOF",
"RealCnt",0,"INTEGER",0,"",
2,
"Realization",0,"INTEGER",0,"",
2,
"RegionDB",0,"STRING",0,"",
"c:\Hwir\parallel_version\software\DATABASE\SiteDB10-02-00.mdb",
"RSOFDirectory",0,"STRING",0,"",
"c:\elp2tb\RSOF",
"SDP",0,"STRING",0,"",
"c:\hwir\sdpbeta2.exe",
"Seed",0,"INTEGER",0,"",
11031,
"SI",0,"STRING",0,"",
"c:\hwir\SI.bat",
"SiteBasedDB",0,"STRING",0,"",
"c:\Hwir\parallel_version\software\DATABASE\smallsite.mdb",
"SiteCnt",0,"INTEGER",0,"",
2,

US EPA ARCHIVE DOCUMENT

"SiteId",0,"STRING",0,"",
"0232313",
"Sites",1,"STRING",0,"",
2,"0114001","0232313",
"SiteSurveyDB",0,"STRING",0,"",
"",
"Source",0,"STRING",0,"",
"SI",
"SrcCnt",0,"INTEGER",0,"",
2,
"Srcs",1,"STRING",0,"",
2,"WP","SI",
"SSFDirectory",0,"STRING",0,"",
"c:\HWIR\parallel_version\Software\SSF",
"StaticNationDB",0,"STRING",0,"",
"c:\HWIR\parallel_version\software\DATABASE\National4-10.mdb",
"StaticRegionDB",0,"STRING",0,"",
"c:\Hwir\parallel_version\software\DATABASE\SiteDB10-02-00.mdb",
"StopOnError",0,"INTEGER",0,"",
0,
"StopOnWarning",0,"INTEGER",0,"",
0,
"StorageLevel",0,"INTEGER",0,"",
0,
"Stream",0,"STRING",0,"",
"c:\Hwir\examsio.exe",
"Terrestrial",0,"STRING",0,"",
"c:\hwir\TF.exe",
"Time",0,"STRING",0,"",
"11:36am",
"Vadose",0,"STRING",0,"",
"c:\hwir\vadose.exe",
"WaterShed",0,"STRING",0,"",
"c:\hwir\ws.bat",
"WP",0,"STRING",0,"",
"c:\hwir\WP.bat",

The contents of the ‘suitasker.bat’ file will need to be edited to run on the tester’s computer. The contents are listed below with an explanation of what changes need to be made.

SUITASKER.BAT

"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" HWIRNet.SUITasker WP-TAIRAR
C:\HWIR\parallel_version\Software\ssf hdprod.ssf Test1 c: c: WP-TAIRAR 200
pause

The first line is as follows:

US EPA ARCHIVE DOCUMENT

"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" HWIRNet.SUITasker WP-TAIRAR C:\HWIR\parallel_version\Software\ssf hdprod.ssf Test1 c: c: WP-TAIRAR 200

The following is a description of each item in the first line:

a) "c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" - The location of java on the tester's computer. Note that this should be changed to reflect the location of java on the tester's computer.

b) HWIRNet.SUITasker - The SUI Tasker executable. This does not need to be changed.

c) WP-TAIRAR - The computer name where the SUI Tasker resides. This should be changed to reflect the name of the tester's computer.

d) C:\HWIR\parallel_version\Software\ssf - The folder location of the SSF file. This should be changed to reflect the location of the folder on the tester's computer.

e) hdprod.ssf - The name of the HWIR header file. This does not need to be changed.

f) Test1 - The name of current software study being performed. This does not need to be changed.

g) c: - The directory where hwir executables are located on server as specified in the header file. This does not need to be changed.

h) c: - The directory where hwir executables are located on client machines. This does not need to be changed.

i) WP-TAIRAR - The name or IP address where mysql is running. This should be changed to reflect the name of the tester's computer.

j) 200 - The maximum number of tasks allowed in the queue. This does not need to be changed.

k) timeout.csv - The filename that contains that lists explicitly the time to wait on tasks by chemical

The second line contains the word 'pause' and does not need to be changed.

The contents of the 'run.bat' file are listed below and will not be modified for this test case.

RUN.BAT

notepad
del c:\parerror.txt

5.4.1.3 Expected Results

It is expected that the SUI Tasker will read the HWIR header file, hdprod.ssf, and set up the twenty four tasks to be run. The tasks should then be run on both computers that are available. When a task is initiated by a client tasker, the program 'Notepad.exe' should start up. To complete the task, the

US EPA ARCHIVE DOCUMENT

Notepad program should be shut down manually. When the twenty four tasks are completed, the SUI Tasker should list all twenty four as completed successfully.

5.4.1.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user's computers. The test bed files must also be installed on both computers in order to follow this test plan and execute the tests.
2. The first thing you will need to do is locate the SUITasker.bat file on your computer and edit it. As noted previously, you will need to change the executable paths, folders, and computer name to adjust for your computer.
3. After the SUITasker.bat file has been edited and saved, you will next start up the required software components.
4. This test case utilizes two computers. The computer with the SUITasker executable on it will be referred to as the primary computer. The computer that is linked via the network and used as an additional resource to complete tasks will be referred to as the second computer. Both computers should have the entire HWIR software and test package loaded on them.
5. Check that the primary computer has the cpuallocator executable 'cpuallocator.bat', suitasker executable 'suitasker.bat', taskerclient executable 'taskerclient.bat', the header file 'hdprod.ssf', and the run.bat file.
6. Check that the secondary computer has the 'taskerclient.bat' and the run.bat file.
7. On the primary computer, start up the cpuallocator by double clicking on the file 'cpuallocator.bat'. Then start up the client tasker by double clicking on the file 'clienttasker.bat'.
8. Start up the client tasker on the second computer by double clicking on the file 'clienttasker.bat'.
9. Then start up the SUI Tasker by double clicking on the file 'suitasker.bat'.
10. The 'Tasks' tab should immediately fill up with the list of tasks to be completed. Note that the bottom of the user interface (UI), statistics on the runs are shown. These statistics are '# Complete', 'Average Time', '# Errors', and 'Queued Runs'. The 'Queued Runs' category should show the number 24, which is the total number of tasks to be run.
11. You will next note that the first two tasks in the list have been assigned. One should have been assigned to the primary computer and another to the second computer. You should also note that the Notepad program is up on both computers. Executing the Notepad program is the task that is to be completed for all tasks. The task is completed when the Notepad program is closed down by the user. Close down Notepad on both computers. You will note that the two tasks are removed from the 'Tasks' tab, the '# Complete' counter goes to 2, and the 'Queued Runs' counter goes to 22.
12. The next two tasks are then assigned one to each computer again and Notepad is started on both computers. Close down Notepad and repeat this process until all tasks have been completed.
13. When all tasks have been completed, the '# Complete' counter should read 24, the '# Errors' counter should read 0, and the 'Queued Runs' counter should read 0.
14. Close down the SUI Tasker, CPU Allocator, and Client Tasker on both computers.
15. This completes this test case.

5.4.1.5 Results

The SUI Tasker read the tasks to be completed from the HWIR header file, distributed tasks to the two available client taskers, and tracked their successful completions. This test case was completed successfully.

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

5.4.2 SUITasker02

5.4.2.1 Description

This test case tests the basic functionality of the SUI Tasker. The SUI Tasker should: 1) read HWIR header files, 2) generate a task set for simulations, 3) distribute tasks to clients, 4) track tasks and reassign if tasks take too long to complete, and 5) keep track of which tasks fail and which machines they fail on.

The header file in this case requires that four tasks be completed. The file specifies that there is one site, one source type, one constituent, two Cw bins, and two realizations. Client taskers on two different networked computers will be used to verify that tasks are distributed to different clients.

5.4.2.2 Input Data

The input for this test case is in three files and they are the ‘hdprod2.ssf’, ‘suitasker.bat’, and the ‘run.bat’ files. The contents of the ‘hdprod2.ssf’ file are listed below with the contents of interest in bold.

HDPROD2.SSF

1,
"hdprod.ssf","data group",
62,
"Air",0,"STRING",0,"",
"c:\hwir\AirModel.exe",
"Aquatic",0,"STRING",0,"",
"c:\HWIR\afw.exe",
"Aquifer",0,"STRING",0,"",
"c:\hwir\aquifer1.exe",
"AT",0,"STRING",0,"",
"c:\hwir\AT.bat",
"CASID",0,"STRING",0,"",
"71-43-2",
"ChemCnt",0,"INTEGER",0,"",
1,
"Chems",1,"STRING",0,"",
1,"Benzene",
"COP",0,"STRING",0,"",
"c:\hwir\DummyProc.bat",
"CPDirectory",0,"STRING",0,"",
"C:\HWIR\parallel_version\Software\CPPdata",
"CW",0,"INTEGER",0,"",
0,
"CWCnt",0,"INTEGER",0,"",
2,
"CWs",1,"STRING",0,"",

2,"3","1",
"Date",0,"STRING",0,"",
"09/06/2002",
"Debug",0,"INTEGER",0,"",
0,
"DSP",0,"STRING",0,"",
"c:\hwir\DummyProc.bat",
"EcoExposure",0,"STRING",0,"",
"c:\hwir\EE.exe",
"EcoRisk",0,"STRING",0,"",
"c:\Hwir\EcoRisk5.exe",
"ELP1",0,"STRING",0,"",
"c:\hwirsrc\ELPIMSqlC\ELP1MYSQL.bat",
"ELP2",0,"STRING",0,"",
"c:\hwir\ELP2.exe",
"Farm",0,"STRING",0,"",
"c:\hwir\FarmFood.exe",
"GRFDirectory",0,"STRING",0,"",
"c:\HWIR\parallel_version\Software\GRF",
"HumanExposure",0,"STRING",0,"",
"c:\hwir\exposure.exe",
"HumanRisk",0,"STRING",0,"",
"c:\Hwir\HRord63.exe",
"Lake",0,"STRING",0,"",
"c:\Hwir\examsio.exe",
"LastCw",0,"LOGICAL",0,"",
"T",
"LAU",0,"STRING",0,"",
"c:\hwir\LAU.bat",
"LF",0,"STRING",0,"",
"c:\hwir\LF.bat",
"MemoCnt",0,"INTEGER",0,"",
0,
"MetDir",0,"STRING",0,"",
"c:\hwir\Metdata",
"MMSP",0,"STRING",0,"",
"c:\hwir\MMSP.exe",
"NationDB",0,"STRING",0,"",
"c:\HWIR\parallel_version\software\DATABASE\National4-10.mdb",
"NewChem",0,"LOGICAL",0,"",
"T",
"NewRel",0,"LOGICAL",0,"",
"T",
"Permanent",0,"STRING",0,"",
"c:\hwir\Permanent",
"PSOFDirectory",0,"STRING",0,"",
"c:\elp2tb\PSOF",
"RealCnt",0,"INTEGER",0,"",

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

2,
"Realization",0,"INTEGER",0,"",
2,
"RegionDB",0,"STRING",0,"",
"c:\Hwir\parallel_version\software\DATABASE\SiteDB10-02-00.mdb",
"RSOFDirectory",0,"STRING",0,"",
"c:\elp2tb\RSOF",
"SDP",0,"STRING",0,"",
"c:\hwir\sdpbeta2.exe",
"Seed",0,"INTEGER",0,"",
11031,
"SI",0,"STRING",0,"",
"c:\hwir\SI.bat",
"SiteBasedDB",0,"STRING",0,"",
"c:\Hwir\parallel_version\software\DATABASE\smallsite.mdb",
"SiteCnt",0,"INTEGER",0,"",
1,
"SiteId",0,"STRING",0,"",
"0114001",
"Sites",1,"STRING",0,"",
1,"0114001",
"SiteSurveyDB",0,"STRING",0,"",
"",
"Source",0,"STRING",0,"",
"WP",
"SrcCnt",0,"INTEGER",0,"",
1,
"Srcs",1,"STRING",0,"",
1,"WP",
"SSFDirectory",0,"STRING",0,"",
"c:\HWIR\parallel_version\Software\SSF",
"StaticNationDB",0,"STRING",0,"",
"c:\HWIR\parallel_version\software\DATABASE\National4-10.mdb",
"StaticRegionDB",0,"STRING",0,"",
"c:\Hwir\parallel_version\software\DATABASE\SiteDB10-02-00.mdb",
"StopOnError",0,"INTEGER",0,"",
0,
"StopOnWarning",0,"INTEGER",0,"",
0,
"StorageLevel",0,"INTEGER",0,"",
0,
"Stream",0,"STRING",0,"",
"c:\Hwir\examsio.exe",
"Terrestrial",0,"STRING",0,"",
"c:\hwir\TF.exe",
"Time",0,"STRING",0,"",
"11:36am",
"Vadose",0,"STRING",0,"",

"c:\hwir\vadose.exe",
"WaterShed",0,"STRING",0,"",
"c:\hwir\ws.bat",
"WP",0,"STRING",0,"",
"c:\hwir\WP.bat",

The contents of the 'suitasker.bat' file will need to be edited to run on the tester's computer. The contents are listed below with an explanation of what changes need to be made.

SUITASKER.BAT

"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" HWIRNet.SUITasker WP-TAIRAR
C:\HWIR\parallel_version\Software\ssf hdprod.ssf Test1 c: c: WP-TAIRAR 200
pause

The first line is as follows:

"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" HWIRNet.SUITasker WP-TAIRAR
C:\HWIR\parallel_version\Software\ssf hdprod.ssf Test1 c: c: WP-TAIRAR 200

The following is a description of each item in the first line:

a) "c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" - The location of java on the tester's computer. Note that this should be changed to reflect the location of java on the tester's computer.

b) HWIRNet.SUITasker - The SUI Tasker executable. This does not need to be changed.

c) WP-TAIRAR - The computer name where the SUI Tasker resides. This should be changed to reflect the name of the tester's computer.

d) C:\HWIR\parallel_version\Software\ssf - The folder location of the SSF file. This should be changed to reflect the location of the folder on the tester's computer.

e) hdprod.ssf - The name of the HWIR header file. This does not need to be changed.

f) Test1 - The name of current software study being performed. This does not need to be changed.

g) c: - The directory where hwir executables are located on server as specified in the header file. This does not need to be changed.

h) c: - The directory where hwir executables are located on client machines. This does not need to be changed.

i) WP-TAIRAR - The name or IP address where mysql is running. This should be changed to reflect the name of the tester's computer.

j) 200 - The maximum number of tasks allowed in the queue. This does not need to be changed.

US EPA ARCHIVE DOCUMENT

The second line contains the word ‘pause’ and does not need to be changed.

The contents of the ‘run.bat’ file are listed below and will be modified for this test case.

RUN.BAT

notepad
del c:\parerror.txt

5.4.2.3 Expected Results

It is expected that the SUI Tasker will read the HWIR header file, hdprod.ssf, and set up the four tasks to be run. The tasks should then be run on both computers that are available. When a task is initiated by a client tasker, the program ‘Notepad.exe’ should start up. To complete the task, the Notepad program should be shut down manually. The tasks run on the primary computer should end with an error due to an adjustment that is made to the ‘run.bat’ file on the primary computer. When the four tasks are completed, the SUI Tasker should list two as completed successfully and two as errors.

5.4.2.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user’s computers. The test bed files must also be installed on both computers in order to follow this test plan and execute the tests.
2. The first thing you will need to do is locate the SUITasker.bat file on your computer and edit it. As noted previously, you will need to change the executable paths, folders, and computer name to adjust for your computer.
3. After the SUITasker.bat file has been edited and saved, you will next start up the required software components.
4. This test case utilizes two computers. The computer with the SUITasker executable on it will be referred to as the primary computer. The computer that is linked via the network and used as an additional resource to complete tasks will be referred to as the second computer. Both computers should have the entire HWIR software and test package loaded on them.
5. Check that the primary computer has the cpuallocator executable ‘cpuallocator.bat’, suitasker executable ‘suitasker.bat’, taskerclient executable ‘taskerclient.bat’, the header file ‘hdprod2.ssf’, and the run.bat file.
6. You will need to switch the names of two files such that ‘hdprod2.ssf’ can be used. Using Windows Explorer change the name of the current ‘hdprod.ssf’ to ‘hdprod1.ssf’.
7. Then change the name of the file ‘hdprod2.ssf’ to ‘hdprod.ssf’.
8. Check that the secondary computer has the ‘taskerclient.bat’ and the run.bat file.
9. On the primary computer, open up the ‘run.bat’ file in a text editor.
10. On the second line enter the word ‘rem’ and then a space at the start of the line so that it reads ‘rem del c:\parerror.txt’. Remarking out this line will leave the file ‘parerror.txt’ on the computer. The presence of this file indicates to the SUI Tasker that there was an error.
11. On the primary computer, start up the cpuallocator by double clicking on the file ‘cpuallocator.bat’. Then start up the client tasker by double clicking on the file ‘clienttasker.bat’.
12. Start up the client tasker on the second computer by double clicking on the file ‘clienttasker.bat’.
13. Then start up the SUI Tasker by double clicking on the file ‘suitasker.bat’.

US EPA ARCHIVE DOCUMENT

14. The ‘Tasks’ tab should immediately fill up with the list of tasks to be completed. Note that the bottom of the user interface (UI), statistics on the runs are shown. These statistics are ‘# Complete’, ‘Average Time’, ‘# Errors’, and ‘Queued Runs’. The ‘Queued Runs’ category should show the number 4, which is the total number of tasks to be run.
15. You will next note that the first two tasks in the list have been assigned. One should have been assigned to the primary computer and another to the second computer. You should also note that the Notepad program is up on both computers. Executing the Notepad program is the task that is to be completed for all tasks. The task is completed when the Notepad program is closed down by the user. Close down Notepad on both computers. Recall that the task running on the main computer should produce an error because the ‘parerror.txt’ file is not being deleted. You will note that the two tasks are removed from the ‘Tasks’ tab, the ‘# Complete’ counter goes to 1, the ‘# Errors’ counter goes to 1, and the ‘Queued Runs’ counter goes to 2.
16. The next two tasks are then assigned one to each computer again and Notepad is started on both computers. Close down Notepad and repeat this process. All tasks should now be completed.
17. When all tasks have been completed, the ‘# Complete’ counter should read 2, the ‘# Errors’ counter should read 2, and the ‘Queued Runs’ counter should read 0.
18. Click on the ‘Failed Tasks’ tab to view the tasks that failed. Also listed is the computer that the task failed on.
19. Close down the SUI Tasker, CPU Allocator, and Client Tasker on both computers.
20. This completes this test case.

5.4.2.5 Results

The SUI Tasker read the tasks to be completed from the HWIR header file, distributed tasks to the two available client taskers, and tracked their successful completions or errors. This test case was completed successfully.

5.4.3 SUITasker03

5.4.3.1 Description

The purpose of this test case is to verify that the SUI Tasker will reassign tasks that take longer than the time specified in the ‘timeout.csv’ file. The timeout for benzene will be reduced to 20 seconds, which is longer than the task will take to finish. The SUI Tasker should recognize this and reassign the task to another client tasker. The test case set up in test case SUITasker03 will be re-run with slight modifications.

The header file in this case requires that four tasks be completed. The file specifies that there is one site, one source type, one constituent, two Cw bins, and two realizations. Client taskers on two different networked computers will be used to verify that tasks are distributed to different clients. Note that whether or not the task itself errors out is not important to this test case. This test case is just verifying that tasks will be redistributed if they take longer to run than is specified in the ‘timeout.csv’ file.

5.4.3.2 Input Data

US EPA ARCHIVE DOCUMENT

The input for this test case is in three files and they are the ‘hdprod2.ssf’, ‘suitasker.bat’, ‘run.bat’, and the ‘timeout.csv’ files. The contents of the ‘hdprod2.ssf’ file are listed below with the contents of interest in bold.

HDPROD2.SSF

1,
"hdprod.ssf","data group",
62,
"Air",0,"STRING",0,"",
"c:\hwir\AirModel.exe",
"Aquatic",0,"STRING",0,"",
"c:\HWIR\afw.exe",
"Aquifer",0,"STRING",0,"",
"c:\hwir\aquifer1.exe",
"AT",0,"STRING",0,"",
"c:\hwir\AT.bat",
"CASID",0,"STRING",0,"",
"71-43-2",
"ChemCnt",0,"INTEGER",0,"",
1,
"Chems",1,"STRING",0,"",
1,"Benzene",
"COP",0,"STRING",0,"",
"c:\hwir\DummyProc.bat",
"CPDirectory",0,"STRING",0,"",
"C:\HWIR\parallel_version\Software\CPPdata",
"CW",0,"INTEGER",0,"",
0,
"CWCnt",0,"INTEGER",0,"",
2,
"CWs",1,"STRING",0,"",
2,"3","1",
"Date",0,"STRING",0,"",
"09/06/2002",
"Debug",0,"INTEGER",0,"",
0,
"DSP",0,"STRING",0,"",
"c:\hwir\DummyProc.bat",
"EcoExposure",0,"STRING",0,"",
"c:\hwir\EE.exe",
"EcoRisk",0,"STRING",0,"",
"c:\Hwir\EcoRisk5.exe",
"ELP1",0,"STRING",0,"",
"c:\hwirsrc\ELPIMSqlC\ELP1MYSQL.bat",
"ELP2",0,"STRING",0,"",
"c:\hwir\ELP2.exe",
"Farm",0,"STRING",0,"",

US EPA ARCHIVE DOCUMENT

"c:\hwir\FarmFood.exe",
"GRFDirectory",0,"STRING",0,"",
"c:\HWIR\parallel_version\Software\GRF",
"HumanExposure",0,"STRING",0,"",
"c:\hwir\exposure.exe",
"HumanRisk",0,"STRING",0,"",
"c:\Hwir\HRord63.exe",
"Lake",0,"STRING",0,"",
"c:\Hwir\examsio.exe",
"LastCw",0,"LOGICAL",0,"",
"T",
"LAU",0,"STRING",0,"",
"c:\hwir\LAU.bat",
"LF",0,"STRING",0,"",
"c:\hwir\LF.bat",
"MemoCnt",0,"INTEGER",0,"",
0,
"MetDir",0,"STRING",0,"",
"c:\hwir\Metdata",
"MMSP",0,"STRING",0,"",
"c:\hwir\MMSP.exe",
"NationDB",0,"STRING",0,"",
"c:\HWIR\parallel_version\software\DATABASE\National4-10.mdb",
"NewChem",0,"LOGICAL",0,"",
"T",
"NewRel",0,"LOGICAL",0,"",
"T",
"Permanent",0,"STRING",0,"",
"c:\hwir\Permanent",
"PSOFDirectory",0,"STRING",0,"",
"c:\elp2tb\PSOF",
"RealCnt",0,"INTEGER",0,"",
2,
"Realization",0,"INTEGER",0,"",
2,
"RegionDB",0,"STRING",0,"",
"c:\Hwir\parallel_version\software\DATABASE\SiteDB10-02-00.mdb",
"RSOFDirectory",0,"STRING",0,"",
"c:\elp2tb\RSOF",
"SDP",0,"STRING",0,"",
"c:\hwir\sdpbeta2.exe",
"Seed",0,"INTEGER",0,"",
11031,
"SI",0,"STRING",0,"",
"c:\hwir\SI.bat",
"SiteBasedDB",0,"STRING",0,"",
"c:\Hwir\parallel_version\software\DATABASE\smallsite.mdb",
"SiteCnt",0,"INTEGER",0,"",

US EPA ARCHIVE DOCUMENT

1,
"SiteId",0,"STRING",0,"",
"0114001",
"Sites",1,"STRING",0,"",
1,"0114001",
"SiteSurveyDB",0,"STRING",0,"",
"",
"Source",0,"STRING",0,"",
"WP",
"SrcCnt",0,"INTEGER",0,"",
1,
"Srcs",1,"STRING",0,"",
1,"WP",
"SSFDirectory",0,"STRING",0,"",
"c:\HWIR\parallel_version\Software\SSF",
"StaticNationDB",0,"STRING",0,"",
"c:\HWIR\parallel_version\software\DATABASE\National4-10.mdb",
"StaticRegionDB",0,"STRING",0,"",
"c:\Hwir\parallel_version\software\DATABASE\SiteDB10-02-00.mdb",
"StopOnError",0,"INTEGER",0,"",
0,
"StopOnWarning",0,"INTEGER",0,"",
0,
"StorageLevel",0,"INTEGER",0,"",
0,
"Stream",0,"STRING",0,"",
"c:\Hwir\examsio.exe",
"Terrestrial",0,"STRING",0,"",
"c:\hwir\TF.exe",
"Time",0,"STRING",0,"",
"11:36am",
"Vadose",0,"STRING",0,"",
"c:\hwir\vadose.exe",
"WaterShed",0,"STRING",0,"",
"c:\hwir\ws.bat",
"WP",0,"STRING",0,"",
"c:\hwir\WP.bat",

The contents of the 'suitasker.bat' file will need to be edited to run on the tester's computer. The contents are listed below with an explanation of what changes need to be made.

SUITASKER.BAT

"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" HWIRNet.SUITasker WP-TAIRAR
C:\HWIR\parallel_version\Software\ssf hdprod.ssf Test1 c: c: WP-TAIRAR 200
pause

US EPA ARCHIVE DOCUMENT

The first line is as follows:

"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" HWIRNet.SUITasker WP-TAIRAR C:\HWIR\parallel_version\Software\ssf hdprod.ssf Test1 c: c: WP-TAIRAR 200

The following is a description of each item in the first line:

a) "c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" - The location of java on the tester's computer. Note that this should be changed to reflect the location of java on the tester's computer.

b) HWIRNet.SUITasker - The SUI Tasker executable. This does not need to be changed.

c) WP-TAIRAR - The computer name where the SUI Tasker resides. This should be changed to reflect the name of the tester's computer.

d) C:\HWIR\parallel_version\Software\ssf - The folder location of the SSF file. This should be changed to reflect the location of the folder on the tester's computer.

e) hdprod.ssf - The name of the HWIR header file. This does not need to be changed.

f) Test1 - The name of current software study being performed. This does not need to be changed.

g) c: - The directory where hwir executables are located on server as specified in the header file. This does not need to be changed.

h) c: - The directory where hwir executables are located on client machines. This does not need to be changed.

i) WP-TAIRAR - The name or IP address where mysql is running. This should be changed to reflect the name of the tester's computer.

j) 200 - The maximum number of tasks allowed in the queue. This does not need to be changed.

The second line contains the word 'pause' and does not need to be changed.

The contents of the 'run.bat' file are listed below and will be modified for this test case.

RUN.BAT

```
notepad
del c:\parerror.txt
```

The contents of the 'timeout.csv' file are listed below and will be modified for this test case.

TIMEOUT.CSV

US EPA ARCHIVE DOCUMENT

2
"default",172800
"71-43-2",2000

5.4.3.3 Expected Results

It is expected that the SUI Tasker will read the HWIR header file, hdprod.ssf, and set up the four tasks to be run. The tasks should then be run on both computers that are available. When a task is initiated by a client tasker, the program 'Notepad.exe' should start up. To complete the task, the Notepad program should be shut down manually. The first task run on the primary computer will not end in the time specified through the 'timeout.csv' file because Notepad.exe will not be shut down. The tasks on the second computer will be shut down. After the other three tasks have completed, the task that is running on the primary computer will then be reassigned to run on the second. When the four tasks are completed, the SUI Tasker should list four tasks as completed successfully.

5.4.3.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user's computers. The test bed files must also be installed on both computers in order to follow this test plan and execute the tests.
2. The first thing you will need to do is locate the SUITasker.bat file on your computer and edit it. As noted previously, you will need to change the executable paths, folders, and computer name to adjust for your computer.
3. After the SUITasker.bat file has been edited and saved, you will next start up the required software components.
4. This test case utilizes two computers. The computer with the SUITasker executable on it will be referred to as the primary computer. The computer that is linked via the network and used as an additional resource to complete tasks will be referred to as the second computer. Both computers should have the entire HWIR software and test package loaded on them.
5. Check that the primary computer has the cpuallocator executable 'cpuallocator.bat', suitasker executable 'suitasker.bat', taskerclient executable 'taskerclient.bat', the header file 'hdprod2.ssf', and the run.bat file.
6. You will need to switch the names of two files such that 'hdprod2.ssf' can be used. Using Windows Explorer change the name of the current 'hdprod.ssf' to 'hdprod1.ssf'.
7. Then change the name of the file 'hdprod2.ssf' to 'hdprod.ssf'.
8. Check that the secondary computer has the 'taskerclient.bat' and the run.bat file.
9. On the primary computer, open up the 'run.bat' file in a text editor.
10. The second line should read 'del c:\parerror.txt'. If the 'rem' is at the beginning of the line, delete it. This line will delete the file 'parerror.txt' from the computer. The presence of this file indicates to the SUI Tasker that there was an error. Close down the file.
11. On the secondary computer, open up the 'run.bat' file in a text editor.
12. The second line should read 'del c:\parerror.txt'. If the 'rem' is at the beginning of the line, delete it. This line will delete the file 'parerror.txt' from the computer. The presence of this file indicates to the SUI Tasker that there was an error. Close down the file.
13. On the primary computer, open up the 'timeout.csv' file in a text editor.
14. The second line should read ,"71-43-2",20'. If the number at the end of the line is not '20' then change it to '20'. This instructs the SUI Tasker to reassign a task if it takes longer than 20 seconds to complete. Close down the file.

US EPA ARCHIVE DOCUMENT

15. On the primary computer, start up the cpuallocator by double clicking on the file 'cpuallocator.bat'. Then start up the client tasker by double clicking on the file 'clienttasker.bat'.
16. Start up the client tasker on the second computer by double clicking on the file 'clienttasker.bat'.
17. Then start up the SUI Tasker by double clicking on the file 'suitasker.bat'.
18. The 'Tasks' tab should immediately fill up with the list of tasks to be completed. Note that the bottom of the user interface (UI), statistics on the runs are shown. These statistics are '# Complete', 'Average Time', '# Errors', and 'Queued Runs'. The 'Queued Runs' category should show the number 4, which is the total number of tasks to be run.
19. You will next note that the first two tasks in the list have been assigned. One should have been assigned to the primary computer and another to the second computer. You should also note that the Notepad program is up on both computers. Executing the Notepad program is the task that is to be completed for all tasks. The task is completed when the Notepad program is closed down by the user.
20. Close down Notepad on the secondary computer. You will note that the one task is removed from the 'Tasks' tab, the '# Complete' counter goes to 1 and the 'Queued Runs' counter goes to 3. The task assigned to the first computer should still be in the queue and on the list. After 20 seconds have passed, the task will change from being assigned to the primary computer to being unassigned. The change will occur when an available tasker client notifies the SUI Tasker that it is available.
21. The next task on the list is then assigned to secondary computer because the primary computer has not completed its first task yet. Notepad should be started on the secondary computer.
22. Close down Notepad on the secondary computer.
23. Again, the next task on the list is then assigned to secondary computer because the primary computer has not completed its first task yet. Notepad should be started on the secondary computer.
24. Close down Notepad on the secondary computer. Now three of the four tasks have been completed on the secondary computer. Because it has been more than 20 seconds, The SUI Tasker should reassign the first task from the primary computer to the secondary computer.
25. The fourth task should switch to the secondary computer and Notepad should start up on the secondary computer. Close down Notepad. All tasks should now be completed.
26. When all tasks have been completed, the '# Complete' counter should read 4, the '# Errors' counter should read 0, and the 'Queued Runs' counter should read 0.
27. Close down the SUI Tasker, CPU Allocator, and Client Tasker on both computers.
28. Close down Notepad on the primary computer.
29. This completes this test case.

5.4.3.5 Results

The SUI Tasker read the tasks to be completed from the HWIR header file, distributed tasks to the two available client taskers, and tracked their successful completions or errors. When the task assigned to the primary computer did not finish within the 20 seconds specified, the SUI Tasker reassigned the task to the secondary computer. This test case was completed successfully.

6.0 Process Error Program

The Process Error Program (PEP) is used to keep track of which components of an HWIR simulation have succeeded or failed, so that the user can determine whether the cause of the failure is a

US EPA ARCHIVE DOCUMENT

modeling problem or a machine-specific problem. PEP provides the user with the ability to capture error and warning messages and store them in the same location as the Site Summary Tool. It uses the fact that when any component of HWIR system software fails, an error or warning file is written in the grf directory. The PEP simply copies the Warning. or Error file from the grf directory to the MySQL database passed to it in its command line arguments. It has no user interface.

The PEP can be run from the command line or from a batch file with a command in the following form: " *if exist grf\warning. java ProcessMessages c:\hwir\ssf hdprod.ssf warning. DataServer Study1"*. This statement will cause the ProcessMessages program to copy the warning file in the grf directory to the Study1 MySql database on the machine with the name DataServer. If the "warnings" or "errors" table does not exist on the specified machine then the table is created. For the field specification for these tables, see section 6.3. If the database itself is missing it will be created as well.

The "errors" and "warnings" tables will contain all the relevant information needed to reproduce the error. The relevant data items are Component, Site, Source, CASID, Cw, Realization, Seed, MachineName, and the text of the message file. All but MachineName are HWIR indices controlled by the SUI or SUI Tasker. MachineName is the name of the machine the failure or message occurred on. More information on the HWIR indices is available in the HWIR System Software documentation. By examination of the contents of the "errors" and "warnings" tables, the user can determine whether it is a modeling problem or a machine specific problem that caused the failure.

6.1 Requirements

The PEP has several main requirements and they are:

1) Transfer the contents of the error file to a database
2) Transfer the contents of the warning file to a database
3) Create a database for storing error and warning messages in the correct format
4) Provide an option for limiting the amount of text from the warning file that is copied into the database to 80 characters
5) Allow the user to specify the processor or module that created the error and warning file
6) Determine the processor or module that created the error and warning file if it is not specified explicitly by the user
7) Only operate if the MySQL server is running

6.2 Design

HWIR requires that modules use the HWIRIO.DLL so that the grf\error and grf\warning file will be produced. PEP is a Java program that takes that message file and summarizes the indices into a MySQL database. PEP takes the following arguments.

ProcessMessages <ssfdir> <header> <messagefile> <mysqlserver> <studyName>
{ <Component> | /F |/f }
Where: <ssfdir> Directory of the SSF files for HWIR
<header> Header file to read
<messagefile> Message File to read and process
<MySQLServer> Server name that is running mysql
<study> Name of the current study

US EPA ARCHIVE DOCUMENT

{<Component>} optional Component name
{ /F or /f } keep full warning messages

The ssf directory and header file are best understood by reading the HWIR System software documentation. The "message file" is the file you wish to process, either grf\error or grf\warning. The PEP uses this filename to determine which table in the MySQL database that will be updated. "MySQLServer" is the name or IP address of a machine that is running the MySQL. "Study" is the name of the database that should be updated with the messages. The name study is given to this because conceptually users of the software will want to accumulation results for each study. "Component" is used to specify which component is failed/ran if it cannot be determined by the content of the grf\error or grf\warning file. For example:

ProcessMessages c:\hwir\ssf hdrun1.ssf grf\error 0100Data study1 SDP

would read the grf\error and update the study1 database on server 0100Data. The component would be the SDP. The absolute path of the grf and ssf directories will be read from the hdrun1.sff file. More information on the relationship between the hd*.ssf files and the ssf and grf directories can be found in the HWIR system software documentation.

6.3 Specifications

The PEP relies on the HWIRIO.DLLs error reporting mechanism and the structure of a MySQL table that will be created if it does not exist. The field definition for the errors and warnings table is as follows:

Table 6.1. Field Definitions For Errors and Warnings Table

FieldName	Type	Description
Component	Char(20)	Component associated with message
Site	Char(10)	HWIR Site name
Source	Char(3)	WP, LF, AT, SI, LAU HWIR Source type
CASID	Char(20)	Chemical Abstract System Identifier
Cw	int	Index of Concentration in the waste 0-4
Realization	int	Monte Carlo simulation count
Seed	int	Seed for Pseudo Random Number Generator

US EPA ARCHIVE DOCUMENT

Line1, Line2, FullText	text (unconstrained string of character)	The first two lines of the message file as well as the whole text
MachineName	Char(40)	Name of the machine that spawned the message

Where the primary key is Component, Site, Source, CASID, CW, Realization, Seed, and MachineName.

6.4 Testing Approach and Results

Table 6.2. Fundamental Requirements for Testing the PEP

Requirement Number	Requirement
1	Transfer the contents of the error file to a database
2	Transfer the contents of the warning file to a database
3	Create a database for storing error and warning messages in the correct format
4	Provide an option for writing the entire text from the warning file into the database. The default is to only write the first 80 characters.
5	Allow the user to specify the processor or module that created the error and warning file
6	Read the 39th and 40th characters in the ERROR or WARNING files to determine the processor or module that created the error and warning file if it is not specified explicitly by the user
7	Only operate if the MySQL server is running

To ensure that the PEP meets the requirements listed in Table 6.2, the following test cases were developed to verify the performance of the PEP. Table 6. shows the relationship between these requirements and the test cases, which are described below.

US EPA ARCHIVE DOCUMENT

Table 6.3. Relationship Between Test Cases and Fundamental Requirements for the PEP

		Test Case Number			
		01	02	03	04
Requirement	1	X			
	2		X		
	3			X	
	4		X		
	5	X	X		
	6	X	X		
	7				X

6.4.1 PEP01

6.4.3.1 Description

This test case tests the basic functionality of the PEP. The PEP should: 1) copy messages from the ERROR file in the GRF folder to an existing database, 2) allow the user to specify the processor that created the error message, and 3) determine the processor that created the error message if none is specified by the user.

6.4.3.2 Input Data

The input for this test case is in two files and they are the 'ProcessError.bat' and the 'ERROR' files.

The contents of the ERROR file are listed below:

" Failed to call closegroups writing sw1.grf".

The contents of the ProcessError.bat file are listed below:

"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" -cp MySQL.jar;. HWIRNet.ProcessMessages c:\hwir\parallel_version\Software\ssf hdprod.ssf c:\hwir\parallel_version\Software\grf\ERROR. WP-TAIRAR Study1 ELP1
"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" -cp MySQL.jar;. HWIRNet.ProcessMessages c:\hwir\parallel_version\Software\ssf hdprod.ssf c:\hwir\parallel_version\Software\grf\ERROR. WP-TAIRAR Study1
Pause

US EPA ARCHIVE DOCUMENT

The file contains three instructions. A description of all will be provided here. Note that the file and folder locations must be altered to run on the computer being used for testing.

1) The first complete instruction set is:

"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" -cp MySQL.jar;. HWIRNet.ProcessMessages c:\hwir\parallel_version\Software\ssf hdprod.ssf c:\hwir\parallel_version\Software\grf\ERROR. WP-TAIRAR Study1 ELP1

a. "c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" -cp MySQL.jar;. HWIRNet.ProcessMessages

This part of the command describes the location of the java program, SQL server, and fires off the ProcessMessages program

b. c:\hwir\parallel_version\Software\ssf

This is the location of the SSF folder

c. hdprod.ssf

This is the name of the header file in the SSF folder

d. c:\hwir\parallel_version\Software\grf\ERROR.

This is the location of the ERROR file and the name of the file

e. WP-TAIRAR

This is the name of the computer

f. Study1

This is the name of the SQL database

g. ELPI

This is the name of the processor that created the ERROR file

2) The second complete instruction set is:

"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" -cp MySQL.jar;. HWIRNet.ProcessMessages c:\hwir\parallel_version\Software\ssf hdprod.ssf c:\hwir\parallel_version\Software\grf\ERROR. WP-TAIRAR Study1

US EPA ARCHIVE DOCUMENT

The instructions are identical to the first instruction set except that the processor/model name has been left off.

3) The third instruction set is:

"Pause"

This command instructs the computer to leave the DOS window up until the user hits a key. All instructions performed by the PEP are echoed to the screen and this enables the user to read them.

6.4.3.3 Expected Results

It is expected that the first two instruction lines will be carried out and that two entries will be made in the errors table in the SQL database. The first should indicate that the error originated in the ELP1 and the entire message text should be listed. The second should indicate that the error originated in the SW model and the entire message text should be listed. Also the DOS window should remain open until the user hits a key and then it should close down.

6.4.3.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user's computers. The test bed files must also be installed on both computers in order to follow this test plan and execute the tests. In addition to this, the MySQL and MySQL front software should be loaded. Both are freeware and can be downloaded from the internet. Install both into a folder entitled 'MySQL' on your C drive.
2. The first thing you will need to do is locate the ProcessError.bat file on your computer and edit it. As noted previously, you will need to change the executable paths, folders, and computer name to adjust for your computer.
3. After the ProcessError.bat file has been edited, double click on it to fire off the PEP. A DOS window will appear. Hit a key to close it down.
4. Bring up MySQL-Front. This is the graphical user interface for MySQL. When you installed it, it should have created a shortcut on your desktop.
5. A screen should appear entitled 'Connection to MySQL-Host...'. Click on the 'Connect!' button. The MySQL information listing the available databases should appear.
6. In the tree-view structure on the left of the new window, double click on 'Study1', which is the database where the errors table should be located.
7. Click on 'errors' and then double click on 'errors' again in the table on the right.
8. Click on the 'data' tab. You should see two entries. One should list the ELP1 processor and the entire message in the ERROR file. Check that ELP1 is listed in under the 'Component' field and then scroll over to the 'Full' field and click on the cell. The contents of the message will be listed in the window at the bottom of the screen. Check that the entire message from the ERROR file is listed. The second should list the SW model and the entire text message in the ERROR file. Check that SW is listed in under the 'Component' field and then scroll over to the 'Full' field and click on the cell. The contents of the message will be listed in the window at the bottom of the screen.
9. This completes this test case. Close down the MySQL-Front by using the 'File' menu and selecting 'Exit'.

US EPA ARCHIVE DOCUMENT

6.4.3.5 Results

The PEP carried out the first two instruction lines and two entries were made in the errors table in the SQL database. One indicated that the error originated in the ELP1 and the entire message was listed. The second indicated that the error originated in the SW model and the entire message text was listed. Also the DOS window should remained open until the user hit a key and then it closed down.

6.4.2 PEP02

6.4.2.1 Description

This test case tests the basic functionality of the PEP. The PEP should: 1) copy messages from the WARNING file in the GRF folder to an existing database, 2) provide an option for limiting the amount of text from the warning file that will be copied into the database, 3) allow the user to specify the processor that created the warning message, and 4) determine the processor that created the warning message if none is specified by the user.

6.4.2.2 Input Data

The input for this test case is in two files and they are the 'ProcessError.bat' and the 'ERROR' files.

The contents of the WARNING file are listed below:

```
"  Failed to call closegroups writing sw1.grf   ".
"**** 30 Characters Long ****"
"More Than 80 Characters"
```

The contents of the ProcessWarning.bat file are listed below:

```
"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" -cp MySQL.jar;. HWIRNet.ProcessMessages
c:\hwir\parallel_version\Software\ssf hdprod.ssf c:\hwir\parallel_version\Software\grf\warning.
WP-TAIRAR Study1 SDP /F
"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" -cp MySQL.jar;. HWIRNet.ProcessMessages
c:\hwir\parallel_version\Software\ssf hdprod.ssf c:\hwir\parallel_version\Software\grf\Warning.
WP-TAIRAR Study1
Pause
```

The file contains three instructions. A description of all will be provided here. Note that the file and folder locations must be altered to run on the computer being used for testing.

1) The first complete instruction set is:

```
"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" -cp MySQL.jar;. HWIRNet.ProcessMessages
c:\hwir\parallel_version\Software\ssf hdprod.ssf c:\hwir\parallel_version\Software\grf\warning.
WP-TAIRAR Study1 SDP /F
```


US EPA ARCHIVE DOCUMENT

a. "c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" -cp MySQL.jar;. HWIRNet.ProcessMessages

This part of the command describes the location of the java program, SQL server, and fires off the ProcessMessages program

b. c:\hwir\parallel_version\Software\ssf

This is the location of the SSF folder

c. hdprod.ssf

This is the name of the header file in the SSF folder

d. c:\hwir\parallel_version\Software\grf\warning.

This is the location of the WARNING file and the name of the file

e. WP-TAIRAR

This is the name of the computer

f. Study1

This is the name of the SQL database

g. SDP

This is the name of the processor that created the WARNING file

h. /F

This is the command that specifies that the entire text of the WARNING message should be copied into the WARNING table in the Study1 database.

2) The second complete instruction set is:

"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" -cp MySQL.jar;. HWIRNet.ProcessMessages c:\hwir\parallel_version\Software\ssf hdprod.ssf c:\hwir\parallel_version\Software\grf\Warning. WP-TAIRAR Study1

The instructions are identical to the first instruction set except that the processor name and the '/F' command have been left off. Leaving the '/F' instruction off specifies that only the first 80 characters of the WARNING message should be copied into the WARNING table in the Study1 database.

3) The third instruction set is:

"Pause"

US EPA ARCHIVE DOCUMENT

This command instructs the computer to leave the DOS window up until the user hits a key. All instructions performed by the PEP are echoed to the screen and this enables the user to read them.

6.4.2.3 Expected Results

It is expected that the first two instruction lines will be carried out and that two entries will be made in the warnings table in the SQL database. One should indicate that the warning originated in the SW model and only the first 80 characters of the message text should be listed. The second should indicate that the warning originated in the SDP processor and the entire message text should be listed. Also the DOS window should remain open until the user hits a key and then it should close down.

6.4.2.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user's computers. The test bed files must also be installed on both computers in order to follow this test plan and execute the tests. In addition to this, the MySQL and MySQL front software should be loaded. Both are freeware and can be downloaded from the internet. Install both into a folder entitled 'MySQL' on your C drive.
2. The first thing you will need to do is locate the ProcessWarning.bat file on your computer and edit it. As noted previously, you will need to change the executable paths, folders, and computer name to adjust for your computer.
3. After the ProcessWarning.bat file has been edited, double click on it to fire off the PEP. A DOS window will appear. Hit a key to close it down.
4. Bring up MySQL-Front. This is the graphical user interface for MySQL. When you installed it, it should have created a shortcut on your desktop.
5. A screen should appear entitled 'Connection to MySQL-Host...'. Click on the 'Connect!' button. The MySQL information listing the available databases should appear.
6. In the tree-view structure on the left of the new window, double click on 'Study1', which is the database where the warnings table should be located.
7. Click on 'warnings' and then double click on 'warnings' again in the table on the right.
8. Click on the 'data' tab. You should see two entries. One should list the SW model and the first 80 characters of the message in the WARNING file. Check that SW is listed in under the 'Component' field and then scroll over to the 'Full' field and click on the cell. The contents of the message will be listed in the window at the bottom of the screen. Check that only the first 80 characters (the first two lines in the file) of the message are listed. The second should list the SDP processor and the entire text message in the WARNING file. Check that SDP is listed in under the 'Component' field and then scroll over to the 'Full' field and click on the cell. The contents of the message will be listed in the window at the bottom of the screen. Check that the entire message is listed.
9. This completes this test case. Close down the MySQL-Front by using the 'File' menu and selecting 'Exit'.

6.4.2.5 Results

The PEP carried out the first two instruction lines and two entries were made in the warnings table in the SQL database. One indicated that the warning originated in the SW model and only the first 80 characters of the message text were listed. The second indicated that the error originated in the SDP

US EPA ARCHIVE DOCUMENT

processor and the entire message text was listed. Also the DOS window remained open until the user hit a key and then it closed down.

6.4.3 PEP03

6.4.3.1 Description

The PEP should create a database in the MySQL format if the database name specified in the command line does not exist. The database and tables within it should be in the correct format for storing error and warning messages. In this test case, the database name in the ProcessError.bat file will be changed to a file that doesn't exist. The PEP should create the new database and populate it with the error messages.

6.4.3.2 Input Data

The input data for this test case are identical to the input data for test case PEP01 except for two changes in the ProcessError.bat file. For the entry 'f', which is the database name, instead of 'Study1', type in 'Study2'. The 'Study2' database does not exist.

6.4.3.3 Expected Results

It is expected that the PEP will create a database called 'Study2' and will populate it exactly as it populated the 'Study1' database.

6.4.3.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user's computers. The test bed files must also be installed on both computers in order to follow this test plan and execute the tests. In addition to this, the MySQL and MySQL front software should be loaded. Both are freeware and can be downloaded from the internet. Install both into a folder entitled 'MySQL' on your C drive.
2. The first thing you will need to do is locate the ProcessError.bat file on your computer and edit it. As noted previously, you will need to change the executable paths, folders, and computer name to adjust for your computer. Also change 'Study1' to 'Study2' in both places in the file.
3. After the ProcessError.bat file has been edited, double click on it to fire off the PEP. A DOS window will appear. Hit a key to close it down.
4. Bring up MySQL-Front. This is the graphical user interface for MySQL. When you installed it, it should have created a shortcut on your desktop.
5. A screen should appear entitled 'Connection to MySQL-Host...'. Click on the 'Connect!' button. The MySQL information listing the available databases should appear.
6. In the tree-view structure on the left of the new window, double click on 'Study2', which is the database where the warnings table should be located.
7. Click on 'errors' and then double click on 'errors' again in the table on the right.
8. Click on the 'data' tab. You should see two entries. One should list the ELP1 processor and the entire message in the ERROR file. Check that ELP1 is listed in under the 'Component' field and then scroll over to the 'Full' field and click on the cell. The contents of the message will be listed in the window at the bottom of the screen. Check that the entire message from the ERROR

US EPA ARCHIVE DOCUMENT

file is listed. The second should list the SW model and the entire text message in the ERROR file. Check that SW is listed in under the 'Component' field and then scroll over to the 'Full' field and click on the cell. The contents of the message will be listed in the window at the bottom of the screen.

9. This completes this test case. Close down the MySQL-Front by using the 'File' menu and selecting 'Exit'.

6.4.3.5 Results

The PEP created a new database entitled 'Study2' and populated it as expected.

6.4.4 PEP04

6.4.4.1 Description

The PEP should only operate if the MySQL server is running. In the previous test cases, the MySQL server was turned on. In this case it will be turned off to see if the PEP will work.

6.4.4.2 Input Data

The input for this test case is identical to the input listed in test case PEP01 above.

6.4.4.3 Expected Results

It is expected that java will produce an error message when the PEP is invoked indicating that the MySQL server is not on.

6.4.4.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user's computers. The test bed files must also be installed on both computers in order to follow this test plan and execute the tests. In addition to this, the MySQL and MySQL front software should be loaded. Both are freeware and can be downloaded from the internet. Install both into a folder entitled 'MySQL' on your C drive.
2. Bring up a DOS prompt and type in 'net stop mysql'. This should turn off the mysql server. Messages should appear that the MySQL service is stopping and has stopped.
3. Locate the ProcessError.bat file on your computer and edit it. As noted previously, you will need to change the executable paths, folders, and computer name to adjust for your computer.
4. After the ProcessError.bat file has been edited, double click on it to fire off the PEP. A DOS window will appear. There will be a lot of text but near the bottom of the message it should state 'Cannot connect to MySQL server...'. Hit any key to close down the DOS window.
5. This completes this test case. Close down the MySQL-Front by using the 'File' menu and selecting 'Exit'.

6.4.4.5 Results

The PEP did not execute when the MySQL server was not operating.

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

7.0 Update Client Program

The Update Client tool facilitates the copying of files to a large number (hundreds) of machines that make up a cluster. It relies on Microsoft's extended naming convention for shared resources. For example if the C:\hwir drive is shared as HWIR on a machine named OtherMachine then "copy AFile.txt \\OtherMachine\HWIR\" would copy the "AFile.Txt" file on the machine executing the copy command to the "OtherMachine"'s c:\hwir directory. It is easy to see how this can benefit a cluster of Windows machines. The Update Client tool uses a table of information about the client machines, which is located in a .csv file, to define a number of environment variables that are to be used in a batch file. The .csv file can have any name that the user wants to use, however, for the purposes of this testing, the file was called 'machines.csv'. These variables allow the user to change the name, shared drive letter, shared path, and IP address as a batch file executes for each machine. The user interface consists of an area to edit a batch

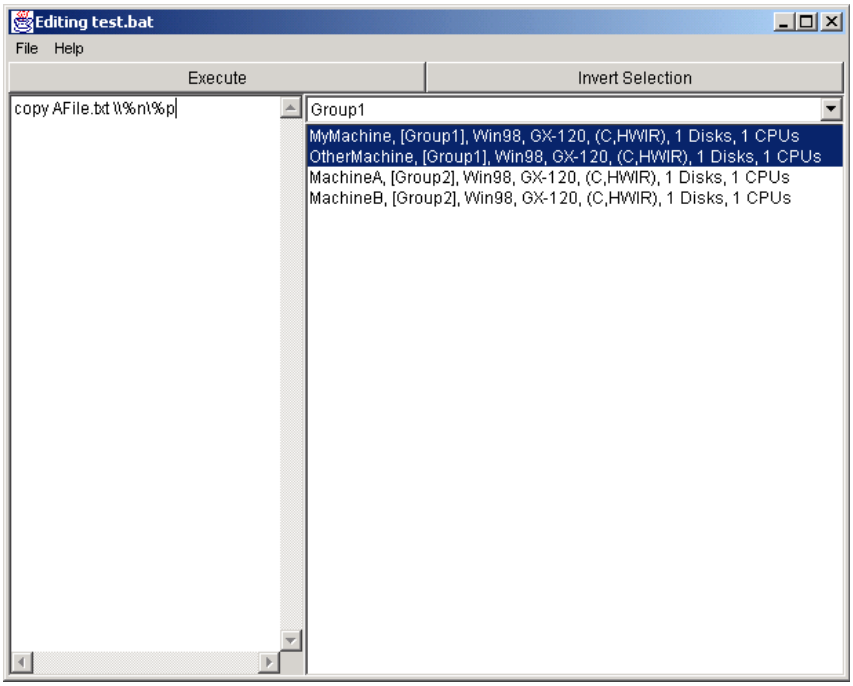

Figure 7.1 Update Client User Interface

file and an ability to choose a subset of machines where the batch files are executed. The sets are determined by fields in the Machine table, which is located in the machines.csv file. So if the batch file says "copy AFile.txt \\%n\%p" and the "OtherMachine" is chosen in the machine list, it is equivalent to "copy AFile.txt \\OtherMachine\HWIR". The user interface looks like Figure 1. The Execute button will execute the batch file for each machine selected. The Invert Selection button will deselect any machines currently selected and select the machines not currently selected. The choice of Group will set the selected list to a set of machines. The Group that each computer is in is specified in the machines.csv file. Section 7.3 will have more detail on how to create groups as well as the format of the machines table file.

US EPA ARCHIVE DOCUMENT

7.1 Requirements

The Update Client has several main requirements and they are:

1) Read in a batch file containing instructions
2) Allow the user to open, edit, and save changes to a batch file
3) Provide a button for executing the update client tool
4) Provide a button for inverting the selection of machines
5) Provide a list of available machines to select from
6) Provide an option for selecting a group of machines to select
7) Provide the mechanism for copying files from a central computer to multiple computers
8) Provide the mechanism for copying files from multiple computers to a central computer

7.2 Design

The Client Update tool gets its input from two files. The first is a standard batch file that contains a number of special environment variables. The second file is a .csv file that lists the available machines and each machines specific values that can may need to be altered depending on the task specified in the batch file. Specifically the KVM_ID, Drive, Path, and IP are the items that can be changed. The variables %n, %d, %p, and %i are used to substitute the Name, Drive, Path, and IP. So "echo %n uses %d:\%p and is at %i" in a batch file that would echo "MyMachine uses c\HWIR and is at 130.20.0.1" for the "MyMachine" values in the csv file table. The table contains more information about each computer but that information is used to help the user in the selection of machines. A complete list of the computer information fields can be found in Section 7.3.

7.3 Specifications

Two specifications are used by the Update Client tool. The first is a standard batch or script file. This is a text file that contains a number of commands to be executed as if they were typed by the user. Operating system documentation should cover this adequately except for the additional %n, %d, %p, %i variables discussed previously.

The second specifications is the Machines table. The Machines table is expected to be a comma separated values file that contains the following columns.

Column Header	Column Description
KVM_ID	Keyboard, Video, Mouse Switch ID Value will replace %n
CPUSpeed	Speed of CPU
CPU Type	Description of CPU
RAM	Amount of Random Access Memory
Group_ID	Group of machines this machine belongs to
OS	Operating system
Model	Model of the machine
Drive	Drive letter. Value will replace %d
Path	Path. Value will replace %p
NumHardDisk	Number of hard disks
NumCPUs	Number of hard disks
IP	Internet Protocol address. Value will replace %I

The columns can be in any order but must have the Column Header text exactly as written on the first line separated by commas. The following is an example of what a table might look like:

KVM_ID,CPUSpeed,CPU
Type,RAM,Group_ID,OS,Model,Drive,Path,NumHardDisk,NumCPUs,IP
MyMachine,500MHz,Pentium,128MB,Group1,Win98,GX-120,C,HWIR,1,1,130.20.0.1
OtherMachine,500MHz,Pentium,128MB,Group1,Win98,GX-120,C,HWIR,1,1,130.20.0.2
MachineA,500MHz,Pentium,128MB,Group2,Win98,GX-120,C,HWIR,1,1,130.20.0.3
MachineB,500MHz,Pentium,128MB,Group2,Win98,GX-120,C,HWIR,1,1,130.20.0.4

This kind of file can easily be produced by creating a spreadsheet with the column names and then saving as it as a comma separated variables (.csv) file.

US EPA ARCHIVE DOCUMENT

7.4 Testing Approach and Results

Table 7.1. Fundamental Requirements for Testing the Update Client Tool

Requirement Number	Requirement
1	Read in a batch file containing instructions
2	Allow the user to open, edit, and save changes to a batch file
3	Read in a .csv file containing machine information
4	Provide a button for executing the update client tool
5	Provide a button for inverting the selection of machines
6	Provide a list of available machines to select from
7	Provide an option for selecting a group of machines to select
8	Provide the mechanism for copying files from a central computer to multiple networked computers
9	Provide the mechanism for copying files from multiple networked computers to a central computer
10	Provide the option to save the results of the executed commands to a file that can be specified by the user

To ensure that the Update Client Tool meets the requirements listed in Table 7.1, the following test cases were developed to verify its performance. Table 7.2 shows the relationship between these requirements and the test cases, which are described below.

Table 7.2. Relationship Between Test Cases and Fundamental Requirements for the Update Client Tool

		Test Case Number	
		01	02
Requirement	1	X	
	2		X
	3	X	
	4	X	
	5		X
	6	X	
	7		X
	8	X	
	9		X
	10		X

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

7.4.1 UCT01

7.4.1.1 Description

This test case will verify that some of the basic functionality of the Update Client tool is working properly. Specifically, the tool's ability to read a batch file containing instructions, read a .csv file containing machine information, provide a button for executing, providing a list of available machines, and providing a mechanism for copying files from a central machine to other networked computers will be checked.

7.4.1.2 Input Data

The input for this test case is located in three different files that have already been pre-populated. The files are the 'UpdateClient.bat', 'test.bat', 'machines.csv', and 'UdateClient2.txt'. The 'UpdateClient.bat' file contains the location of the java program on the user's computer, the command to fire off the UpdateClient program, and the names of the instructions batch file and the machines comma separated file. Note that the location of the java program should be changed such that it is accurate for the user's computer. The 'test.bat' file contains the instructions to be performed by the UpdateClient tool. The 'machines.csv' file contains the computer specific information. Note that the user will need to add a machine or machines to this list that can be accessed on the network that the user is running the Update Client tool on. The 'UpdateClient2.txt' file contains a short text string. It is simply used to verify that a file is being transferred as instructed. The files are listed below:

UpdateClient.bat

```
"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" UpdateClient.Propagate UCtest.bat machines.csv
```

UCtest.bat

```
c:\winnt\system32\cmd.exe /c echo %n %d %p %i
c:\winnt\system32\cmd.exe /c rem copy \\%n\%d\%p\UpdateFile.txt c:\hwir
c:\winnt\system32\cmd.exe /c copy c:\hwir\UpdateFile2.txt \\%n\%d\%p
```

machines.csv

```
KVM_ID,CPUSpeed,CPU Type,RAM,Group_ID,OS,Model,Drive,Path,NumHardDisk,NumCPUs,IP
MyMachine,500MHz,Pentium,128MB,Group1,Win98,GX-120,C,HWIR,1,1,130.20.0.1
OtherMachine,500MHz,Pentium,128MB,Group1,Win98,GX-120,C,HWIR,1,1,130.20.0.2
MachineA,500MHz,Pentium,128MB,Group2,Win98,GX-120,C,HWIR,1,1,130.20.0.3
MachineB,500MHz,Pentium,128MB,Group2,Win98,GX-120,C,HWIR,1,1,130.20.0.4
204.200.142.225,300MHz,Pentium,128MB,Group3,Win98,D300,C,HWIR,1,1,204.200.142.225
WP-TairaR,1.8 GHz,Pentium,261MB,Group3,Win2000,D8100,C$,HWIR,1,1,204.200.142.95
```

UpdateFile2.txt

```
Dummy File for testing
```

7.4.1.3 Expected Results

It is expected that the Update Client tool will read a batch file containing instructions, read a .csv file containing machine information, provide a button for executing, providing a list of available machines, and providing a mechanism for copying files from a central machine to other networked computers.

7.4.1.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user's computers. The test bed files must also be installed on both computers in order to follow this test plan and execute the tests.
2. You will need to edit two of the files that are part of the test bed. Open up the file 'UpdateClient.bat' using a text editor. It will look like this:

   "c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" UpdateClient.Propagate UCtest.bat machines.csv
3. The first entry in the file, which is in quotes, is the location of the java executable on your computer. Change this as necessary so that it describes the location of the java program on your computer. Save the file and exit the editor.
4. You will next need to edit the 'machines.csv' file. Bring up M.S. Excel. Use the 'File' menu to open the file 'machines.csv' as a comma separated file.
5. Add in the information for two computers. One should be the main computer that you are performing the testing on. The other should be a networked computer that you can access and place files on. The Specifications section above contains brief descriptions of the required fields. Save the file as a .csv file and exit M.S. Excel.
6. Place the file 'UpdateFile2.txt' in the c:\hwir drive on the computer that has the Update Client tool on it.
7. Find the file 'UpdateClient.bat' and double click on it. The user interface should appear. The contents of the 'UCtest.bat' file should be listed in the left window and the contents of the 'machines.csv' file should be listed in the right window.
8. The list on the right is the list of the available computers. Note that only the computers that you have added will actually be available to you. In the list, select the computer that you entered that is connected by the network to the main computer that the Update Client tool is being tested on. After clicking on the computer, it should be highlighted in blue.
9. Now that a computer is selected, the Update Client tool is ready to execute the instructions in the batch file on the left (i.e., UCtest.bat). Click on the 'Execute' button, which is above the window on the left.
10. A user interface will appear that asks you to verify that you would like to execute the tool. Click on the 'ok' button.
11. A screen entitled 'Results of execution' will appear, which lists the results of the batch file execution. The contents should look similar to the following with the exception that the computer specific information will be different:

    c:\winnt\system32\cmd.exe /c echo 204.200.142.225 C HWIR 204.200.142.225
    204.200.142.225 C HWIR 204.200.142.225
    Complete
    c:\winnt\system32\cmd.exe /c rem copy \\204.200.142.225\C\HWIR\UpdateFile.txt c:\hwir

    Complete
    c:\winnt\system32\cmd.exe /c copy c:\hwir\UpdateFile2.txt \\204.200.142.225\C\HWIR
    1 file(s) copied.

US EPA ARCHIVE DOCUMENT

CompleteBatch Completed

12. Click on the ‘Stop and Don’t Save’ button to close down the window.
13. Check the c:\hwir folder on your networked computer to verify that the file ‘UpdateFile2.txt’ is there. This will confirm that the Update Client tool executed successfully.
14. To exit the Update Client tool use the ‘File’ menu on the user interface to select ‘Quit’.
15. This concludes this test case.

7.4.1.5 Results

The Update Client tool read read in the batch file containing instructions, read a .csv file containing machine information, provided a button for executing, provided a list of available machines, and copied a file from the central machine to other networked computers.

7.4.2 UCT02

7.4.2.1 Description

This test case will verify that some of the basic functionality of the Update Client tool is working properly. Specifically, the tool’s ability to open, edit, and save changes to a batch file containing instructions, provide a button for inverting the selection of machines to use, provide a method for selecting a group of machines to use, provide a mechanism for copying files from multiple computers to a central computer, and provide the option to save the results of the executed commands to a file that can be specified by the user will be checked.

7.4.2.2 Input Data

The input for this test case is located in three different files that have already been pre-populated. The files are the ‘UpdateClient.bat’, ‘UCtest.bat’, ‘Ucdummy.bat’, ‘machines.csv’, and ‘UdateClient.txt’. The ‘UpdateClient.bat’ file contains the location of the java program on the user’s computer, the command to fire off the UpdateClient program, and the names of the instructions batch file and the machines comma separated file. Note that the location of the java program should be changed such that it is accurate for the user’s computer. The ‘UCtest.bat’ file contains the instructions to be performed by the UpdateClient tool. This file has been pre-populated but will be edited during this case. The ‘Ucdummy.bat’ file contains a line of text that will not be executed. The ‘machines.csv’ file contains the computer specific information. Note that the user will need to add a machine or machines to this list that can be accessed on the network that the user is running the Update Client tool on. The ‘UpdateClient.txt’ file contains a short text string. It is simply used to verify that a file is being transferred as instructed. The files are listed below:

UpdateClient.bat

"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" UpdateClient.Propagate UCtest.bat machines.csv

UCtest.bat

c:\winnt\system32\cmd.exe /c echo %n %d %p %i

US EPA ARCHIVE DOCUMENT

c:\winnt\system32\cmd.exe /c rem copy \\%n\%d\%p\UpdateFile.txt c:\hwir
c:\winnt\system32\cmd.exe /c copy c:\hwir\UpdateFile2.txt \\%n\%d\%p

UCdummy.bat

dummy batch file for test case UCT02

machines.csv

KVM_ID,CPUSpeed,CPU Type,RAM,Group_ID,OS,Model,Drive,Path,NumHardDisk,NumCPUs,IP
MyMachine,500MHz,Pentium,128MB,Group1,Win98,GX-120,C,HWIR,1,1,130.20.0.1
OtherMachine,500MHz,Pentium,128MB,Group1,Win98,GX-120,C,HWIR,1,1,130.20.0.2
MachineA,500MHz,Pentium,128MB,Group2,Win98,GX-120,C,HWIR,1,1,130.20.0.3
MachineB,500MHz,Pentium,128MB,Group2,Win98,GX-120,C,HWIR,1,1,130.20.0.4
204.200.142.225,300MHz,Pentium,128MB,Group3,Win98,D300,C,HWIR,1,1,204.200.142.225
WP-TairaR,1.8 GHz,Pentium,261MB,Group3,Win2000,D8100,C$,HWIR,1,1,204.200.142.95

UpdateFile.txt

Dummy File for testing

7.4.2.3 Expected Results

It is expected that the Update Client tool will open, edit, and save changes to a batch file containing instructions, provide a button for inverting the selection of machines to use, provide a method for selecting a group of machines to use, provide a mechanism for copying files from a networked computer to a central computer, and provide the option to save the results of the executed commands to a file that can be specified by the user.

7.4.2.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user's computers. The test bed files must also be installed on both computers in order to follow this test plan and execute the tests.
2. You will need to edit two of the files that are part of the test bed. Open up the file 'UpdateClient.bat' using a text editor. It will look like this:
   "c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" UpdateClient.Propagate test.bat machines.csv
3. The first entry in the file, which is in quotes, is the location of the java executable on your computer. Change this as necessary so that it describes the location of the java program on your computer. Save the file and exit the editor.
4. You will next need to edit the 'machines.csv' file. Bring up M.S. Excel. Use the 'File' menu to open the file 'machines.csv' as a comma separated file.
5. Add in the information for two computers. One should be the main computer that you are performing the testing on. The other should be a networked computer that you can access and place files on. The Specifications section above contains brief descriptions of the required fields. Save the file as a .csv file and exit M.S. Excel.

US EPA ARCHIVE DOCUMENT

6. Place the file 'UpdateFile.txt' in the c:\hwir drive on the networked computer that can be accessed by your main computer (i.e., the computer that has the Update Client tool on it.
7. Find the file 'UpdateClient.bat' and double click on it. The user interface should appear. The contents of the 'UCtest.bat' file should be listed in the left window and the contents of the 'machines.csv' file should be listed in the right window.
8. Check that other .bat files can be opened by using the 'File' menu and selecting 'Load'. In the window that appears, select the file 'Ucdummy.bat'. The contents of the file should appear in the window below and should look like this:

   dummy batch file for test case UCT02
9. You will note that when you use the 'Load' feature, a new instance of the Updated Client tool is launched. Close down the original instance of the tool by going to the 'File' menu and selecting 'Quit'.
10. Next re-open the original .bat file which was 'Uctest.bat' by using the 'File menu, selecting 'Load', and picking the file 'Uctest.bat'. The contents of the file should appear in the window to the left. Close down the second instance of the tool by going to the 'File' menu and selecting 'Quit'.
11. Next edit the contents of the 'Uctest.bat' file by deleting the word 'rem' from the second row. Also type in the word 'rem' on the third row right before the word 'copy'.
12. Use the 'File' menu and select 'Save'. You have now turned off the command to copy a file from the central computer to a networked computer and have turned on the command to copy a file from the networked computer onto the central computer.
13. The list on the right side of the screen is the list of the available computers. Use the drop down menu below the 'Invert Selection' button to select 'Group2'. Then use it to select 'Group1'. Note that the computers in the various groups are highlighted in blue.
14. Next hit the 'Invert Selection' button. The result should be that all of the computers except those in 'Group1' should be highlighted in blue.
15. You will next select the networked computer that you will use in this test case. Note that only the computers that you have added will actually be available to you. In the list, select the computer that you entered that is connected by the network to the main computer that the Update Client tool is being tested on. After clicking on the computer, it should be highlighted in blue.
16. Make sure that the file 'UpdateFile.txt' is located on the networked computer in the C:\hwir folder and is not located on the C:\hwir folder on the central computer. The objective of this part of the test is to verify that files can be copied from networked computers to the main computer.
17. Now that a computer is selected, the Update Client tool is ready to execute the instructions in the batch file on the left (i.e., UCtest.bat). Click on the 'Execute' button, which is above the window on the left.
18. A user interface will appear that asks you to verify that you would like to execute the tool. Click on the 'ok' button.
19. A screen entitled 'Results of execution' will appear, which lists the results of the batch file execution. The contents should look similar to the following with the exception that the computer specific information will be different:

    c:\winnt\system32\cmd.exe /c echo 204.200.142.225 C HWIR 204.200.142.225
    204.200.142.225 C HWIR 204.200.142.225
    Complete
    c:\winnt\system32\cmd.exe /c copy \\204.200.142.225\C\HWIR\UpdateFile.txt c:\hwir
    1 file(s) copied.

    Complete

US EPA ARCHIVE DOCUMENT

c:\winnt\system32\cmd.exe /c rem copy c:\hwir\UpdateFile2.txt
\\204.200.142.225\C\HWIR
Complete
Batch Completed

20. Before closing down this window, check to see if the file 'UpdateFile.txt' was copied over correctly. Check the C:\hwir folder on your central computer to verify that the file 'UpdateFile.txt' is there. This will confirm that the Update Client tool executed successfully.
21. Next go back to the 'Results of Execution' window. Click on the 'Save' button. A new window will appear and you will be prompted to name the file where the results of the run will be saved to. Type in 'UCT02testResult.txt' and save it to the directory that contains the .bat files.
22. Check the folder where the .bat files are located to confirm that the results file was created. Open up the file to verify that the results from the 'Results of Execution' window are in the file.
23. To exit the Update Client tool use the 'File' menu on the user interface to select 'Quit'.
24. This concludes this test case.

7.4.2.5 Results

The Update Client tool allowed the user to open, edit, and save changes to a batch file containing instructions, provided a button for inverting the selection of machines to use, provided a method for selecting a group of machines to use, provided a mechanism for copying files from a networked computer to a central computer, and provided the option to save the results of the executed commands to a file that was specified by the user.

8.0 Site Summary Tool

The Site Summary Tool is a generalized tool that reads input and output files created by the HWIRIO.dll and summarizes them in tables. The HWIRIO.DLL is the standard mechanism for saving all data produced and consumed in HWIR. Specific details about the HWIRIO.DLL will not be provided here but can be found in the HWIR System Software Documentation. The user in general writes a script that tells the HWIRIO.DLL what values to read at the end of an MMSP simulation in HWIR. If the values read represent a set of numbers, the user has several options for summarizing those values or writing out every value. The summaries include parameterizing (i.e., reporting the mean and standard deviation) as a normal or log-normal distribution as well as specifying certain percentiles to be reported. There is no user interface involved with the Site Summary Tool because it is expected to execute without user intervention during a set of HWIR simulations. All instructions are given through a comma separated text file.

8.1 Requirements

The Site Summary Tool has several main requirements and they are:

1) Use the HWIRIO.DLL to read input and output files
2) Read commands from a text file
3) Write results to a specified mysql table
4) Provide details that describe the results written to the mysql table
5) Process numeric data from input or output files and present the following summary information: number of entries, max, min, mean, and standard deviation.

US EPA ARCHIVE DOCUMENT

6) Provide summary information of numeric data in lognormal space
7) Process numeric data from input or output files and write out percentiles specified by the user
8) Create a table to write results to if the table specified by the user does not exist

8.2 Design

In order to run, the Site Summary Tool requires the HWIR ssf and grf directories, header file, the script to execute, and the database name, which are the command line arguments. Details of the script are given in Section 8.3. The form of the database name controls how the database is updated. Several forms are recognized: .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.

So the command line for the Site Summary to is a follows:

Summary <ssfdir><grfdir> <header> <testscript> \\<mysqlserver>\<studyName>.mysql <delay> <retry>
{ <Component> | /F |/f }
Where: <ssfdir> Directory of the SSF files for HWIR
<grfdir> Directory of the GRF files for HWIR
<header> Header file to read
<testscript>A script which includes the extraction commands.
<MySQLServer> Server name that is running mysql
<studyName> Name of the current study
<delay> Number of seconds to retry
<retry> Number of times to retry the posting of data

If a ".mdb" or ".mysql" database is used, two tables are created to store the summary: The "<testscript>_description" table and the "<testscript>_results" table. The "description" table contains a lengthy description of each column in the "results" table and has the test script filename pre appended. The "results" table contains the values requested for each simulation that the Site Summary tool is executed on. In contrast, if a ".csv" database is used, it produces only one table with the lengthy descriptions as the columns in the file. The ".mdb" and ".csv" functionality are no longer supported or tested.

8.3 Specifications

The script devised for summarizing HWIR inputs and outputs is intended to be both simple and extendable. It is a comma-separated text file that contains commands to the HWIRIO.dll. The commands are "variable","nx", "ny", "nz", "parameterize", "percentiles", and "end". Currently any variable in any dataset can be read as a single value or as part of a set of values. This value or set of values is called and extraction. A single value extraction can be assigned to the special count of "nx", "ny", and "nz".

The"end" is the easiest to describe it simply ends the script.

The "variable" command is followed by an extraction. An extraction consists of the following information:

US EPA ARCHIVE DOCUMENT

Group	A string that is the HWIR group to be read
Variable	A string that is the variable within the group to be read
Units	A string that is the units for the variable. All HWIR variables have units.
Type	A string “Real”,”Integer”,”Logical”, or ”String”
Index1	A number or the special variable “x”,”y”, or “z” that is the first index.
Index6	A number or the special variable “x”, ”y”, or “z” that is the last index.

So “variable,hd.ssf,CASID,,String,1" would tell the summary tool to add to the results the CASID string.

The “nx”, ”ny” and “nz” commands are identical. The represent an ability to define a number to be used as an index in a later extraction. These two command are followed either by a number or an extraction itself. For example “nx,1" will set the “nx” count to 1. But the “nx,extract,CP.ssf,NumChem,,1" would set “nx” to the value found in the NumChem variable in the cp.ssf file. The extraction for “nx”,”ny” and “nz” can include “nx”,”ny” and “nz” themselves.

If an extraction attempts to read a value that is not present a “NULL” value will be stored in the database. So for example is you attempted to extract the 5th element of a 4 element array a “NULL” will be stored in the database. “NULL” is not a semaphore value such as -999.9 or 0.0 but is the absence of value in a database. The “NULL” value mechanism is supported by most databases and database queries can be written that will then ignore these missing values. The nx,ny, and nz indexing system allows the user to read a consistent shape results across all simulations without failure. So if the number of surface water media changes from simulation to simulation the table will simply have “NULL” put in where the surface water media results should have been. A database can then ignore these missing values.

The group names in the script are not exactly the group names used during extraction. For the most part the group name in the script has the site and setting ID injected after the name and before the extension. So for site la123123 the script group name entry of “hd.ssf” would really read “hdla123123.ssf”. This is consistent with the behavior of HWIR system software. The behavior is modified for the “sw.ssf” and “sw.grf” (surface water) groups. The surface water groups have the first index given attached to the group name and the site and setting ID are not added so an extraction of “sw.ssf,SomeVar,String,1" would have the 1 removed from the index list and added to the group name. So in this case the actual group read would be “sw1.ssf.” This is again consistent the other HWIR system software.

“Parameterize” tells the Site Summary Tool to read a set of values then calculate the mean, standard deviation, number of values, minimum, and maximum for the set of values. The script can either produce results for a “Normal” or “LogNormal” distribution. A couple of script entries like:

```
nx,extract,aq.grf,WCFNY,yr,Float,1
parameterize,lognormal,aq.grf,WCF,mg/l,Float,1,x
```

would produce a n, mean, sd, min, max column for this variable. If the WCFNY and WCF variables represented concentration then the output would be the parameters for the lognormal distribution of water concentrations across time.

A script that contains “Percentiles” does much the same thing as “Parameterize” but before the extraction a number of percentiles are given. Using the WCF example again:

```
nx,extract,aq.grf,WCFNY,yr,Float,1
```


US EPA ARCHIVE DOCUMENT

percentiles,4,25,50,75,95,aq.grf,WCF,mg/l,Float,1,x

would produce a 25, 50, 75, and 95 percentile value for the water concentration across time. The ‘4' that preceded the 25 states how many percentages the user wants. So a script that reports CASID, log-normal distribution and percentiles for Water Concentrations would be

```
nx,extract,cp.ssf,NumChem,,String,1
variable,cp.ssf,ChemCASID,,String,x
nx,extract,aq.grf,WCFNY,yr,Float,1
parameterize,lognormal,aq.grf,WCF,mg/l,Float,1,x
percentiles,4,25,50,75,95,aq.grf,WCF,mg/l,Float,1,x
end
```

Would produce results that contain ChemCASID, WCF(1,nx)_n, WCF(1,nx)_mean, WCF(1,nx)_sd, WCF(1,nx)_min, WCF(1,nx)_max,WCF(1,nx)_25, WCF(1,nx)_50, WCF(1,nx)_75, and WCF(1,nx)_95 for each simulation it was invoked on.

The description table gives the details of a column in the results table separated into columns. The fields of the description table are constant and are:

Column Name	Type	Description
alias	char(32)	Short name for result
summary	char (32)	Long name for result
datagroup	char (32)	HWIR datagroup read
variable	char (32)	Variable name
units	char (32)	Units for variable
idx1	int	Integer indices of values (-1 represents nx, -2 represents ny)
idx2	int	
idx3	int	
idx4	int	
idx5	int	
idx6	int	

The fields of the result table depend on the “type” of the variable being stored. If it is a “String” then the column is defined as char(80) all other types are mapped to MySQL or MS Access appropriate types. One column, the MachineName column, is always added to the results table. This allows the Site Summary Tool to describe which machine produced the row of results. The value of MachineName is read using Java’s InetAddress. getLocalHost(). getHostName() function. This feature was added to aid in the tracking down of error or faulty results in the system.

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

8.4 Testing Approach and Results

Table 8.1. Fundamental Requirements for Testing the Site Summary Tool

Requirement Number	Requirement
1	Use the HWIRIO.DLL to read input and output files
2	Read commands from a text file
3	Write results to a specified table in a specified mysql database
4	Provide details that describe the results written to the mysql table
5	Process numeric data from input or output files and present the following summary information: number of entries, max, min, mean, and standard deviation.
6	Have an option to provide summary information of numeric data in lognormal space
7	Process numeric data from input or output files and write out percentiles specified by the user
8	Create a table to report results to if the table specified by the user does not exist

To ensure that the Site Summary Tool (SST) meets the requirements listed in Table 8.1, the following test cases were developed to verify its performance. Table 8.2 shows the relationship between these requirements and the test cases, which are described below.

Table 8.2. Relationship Between Test Cases and Fundamental Requirements for the Site Summary Tool

		Test Case Number	
		01	02
Requirements	1	X	
	2	X	
	3	X	
	4	X	
	5	X	
	6	X	
	7	X	
	8		X

US EPA ARCHIVE DOCUMENT

8.4.1 SST01

8.4.1.1 Description

The SST uses the HWIR IODLL to extract data from standard HWIR input and output files. These files are the Site Simulation File (SSF) and the Global Results File (GRF). The SST then writes this data to a specified table in a mysql database. The SST is also capable of performing summary statistics on numeric data that it retrieves and reporting it in the table in either normal or log-normal space. The SST is also capable of processing numeric data and returning percentile information that is requested by the user. It receives its instructions through reading a text file. The purpose of this test case is to verify that the SST is meeting these requirements. Three parameter values will be extracted from a header file and two parameter values will be extracted from a chemical properties file. In addition, summary statistics will be performed on the numeric data retrieved.

8.4.1.2 Input Data

The input for this test case is located in four files which are the 'Summary.bat', 'Summarytest1.txt', 'hdprod.ssf', and 'cpsr.ssf'. The contents of the files are listed below. Note that changes to the 'Summary.bat' file will be necessary to ensure that executable and folder locations are accurate for the computer being used by the tester.

Summary.bat

```
"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" -cp MySQL.jar;. gov.epa.hwir.util.Summary
c:\hwir\parallel_version\Software\ssf c:\hwir\parallel_version\software\grf hdprod.ssf Summarytest1.txt
\\WP-TAIRAR\study1.mysql 5 1000
pause
```

As noted above, this file will need to be edited by the tester to ensure that executable and folder locations are accurate. The commands in the file will be broken down and it will be noted where changes should be made.

a) "c:\Program Files\javasoft\JRE\1.3.1_03\bin\java"

This section describes the location of the java executable on the tester's computer and will need to be edited by the tester.

b) -cp MySQL.jar;. gov.epa.hwir.util.Summary

This section does not need to be edited.

c) c:\hwir\parallel_version\Software\ssf

This section describes the location of the SSF files on the tester's computer and will need to be edited by the tester.

d) c:\hwir\parallel_version\software\grf

US EPA ARCHIVE DOCUMENT

This section describes the location of the GRF files on the tester's computer and will need to be edited by the tester.

e) hdprod.ssf

This is the name of the header file to be used and does not need to be edited.

f) Summarytest1.txt

This is the name of the instructions file and does not need to be edited.

g) \\WP-TAIRAR\study1.mysql

This is the name of the computer and table where results should be written to. The computer name will need to be edited by the tester.

h) 5

The number of times to retry running the tool.

i) 1000

The time to wait between retries in milliseconds.

Summarytest1.txt

```
variable,hd.ssf,CASID,,String,
nx,extract,hd.ssf,ChemCnt,,Integer,
variable,hd.ssf,Chems,,String,X,
nx,extract,cpsr.ssf,numChem,,Integer,
ny,8,
variable,cpsr.ssf,ChemKd,L/kg,Float,X,Y,
parameterize,lognormal,cpsr.ssf,ChemKd,L/kg,Float,X,Y,
parameterize,normal,cpsr.ssf,ChemKd,L/kg,Float,X,Y,
percentiles,4,25,50,75,99,cpsr.ssf,ChemKd,L/kg,Float,X,Y,
end,
```

hdprod3.ssf

```
1,
"hdprod.ssf","data group",
62,
"Air",0,"STRING",0,"",
"c:\hwir\AirModel.exe",
"Aquatic",0,"STRING",0,"",
"c:\HWIR\afw.exe",
"Aquifer",0,"STRING",0,"",
"c:\hwir\aquifer1.exe",
"AT",0,"STRING",0,"",
```


US EPA ARCHIVE DOCUMENT

"c:\hwir\AT.bat",
"CASID",0,"STRING",0,"",
"71-55-6",
"ChemCnt",0,"INTEGER",0,"",
2,
"Chems",1,"STRING",0,"",
2,"Aniline","Benzene",
"COP",0,"STRING",0,"",
"c:\hwir\DummyProc.bat",
"CPDirectory",0,"STRING",0,"",
"c:\hwir\CPPData",
"CW",0,"INTEGER",0,"",
-1,
"CWCnt",0,"INTEGER",0,"",
3,
"CWs",1,"STRING",0,"",
3,"5","3","1",
"Date",0,"STRING",0,"",
"06/19/2002",
"Debug",0,"INTEGER",0,"",
0,
"DSP",0,"STRING",0,"",
"c:\hwir\DummyProc.bat",
"EcoExposure",0,"STRING",0,"",
"c:\hwir\EE.exe",
"EcoRisk",0,"STRING",0,"",
"c:\Hwir\EcoRisk5.exe",
"ELP1",0,"STRING",0,"",
"c:\hwir\elp1.exe",
"ELP2",0,"STRING",0,"",
"c:\hwir\ELP2.exe",
"Farm",0,"STRING",0,"",
"c:\hwir\FarmFood.exe",
"GRFDirectory",0,"STRING",0,"",
"c:\hwir\GRF",
"HumanExposure",0,"STRING",0,"",
"c:\hwir\exposure.exe",
"HumanRisk",0,"STRING",0,"",
"c:\Hwir\HRord63.exe",
"Lake",0,"STRING",0,"",
"c:\Hwir\examsio.exe",
"LastCw",0,"LOGICAL",0,"",
"T",
"LAU",0,"STRING",0,"",
"c:\hwir\LAU.bat",
"LF",0,"STRING",0,"",
"c:\hwir\LF.bat",
"MemoCnt",0,"INTEGER",0,"",

US EPA ARCHIVE DOCUMENT

0,
"MetDir",0,"STRING",0,"",
"c:\hwir\Metdata",
"MMSP",0,"STRING",0,"",
"c:\hwir\MMSP.exe",
"NationDB",0,"STRING",0,"",
"c:\HWIR\DATABASE\National4-10.mdb",
"NewChem",0,"LOGICAL",0,"",
"F",
"NewRel",0,"LOGICAL",0,"",
"F",
"Permanent",0,"STRING",0,"",
"c:\hwir\Permanent",
"PSOFDirectory",0,"STRING",0,"",
"c:\hwir\PSOF",
"RealCnt",0,"INTEGER",0,"",
1,
"Realization",0,"INTEGER",0,"",
0,
"RegionDB",0,"STRING",0,"",
"c:\Hwir\DataBase\SiteDB10-02-00.mdb",
"RSOFDirectory",0,"STRING",0,"",
"c:\hwir\RSOF",
"SDP",0,"STRING",0,"",
"c:\hwir\sdpbeta2.exe",
"Seed",0,"INTEGER",0,"",
11031,
"SI",0,"STRING",0,"",
"c:\hwir\SI.bat",
"SiteBasedDB",0,"STRING",0,"",
"c:\Hwir\DataBase\SiteDB10-02-00.mdb",
"SiteCnt",0,"INTEGER",0,"",
2,
"SiteId",0,"STRING",0,"",
"",
"Sites",1,"STRING",0,"",
2,"0114001","0130207",
"SiteSurveyDB",0,"STRING",0,"",
"",
"Source",0,"STRING",0,"",
"",
"SrcCnt",0,"INTEGER",0,"",
5,
"Srcs",1,"STRING",0,"",
5,"LAU","WP","SI","AT","LF",
"SSFDirectory",0,"STRING",0,"",
"c:\hwir\SSF",
"StaticNationDB",0,"STRING",0,"",

US EPA ARCHIVE DOCUMENT

"c:\HWIR\DATABASE\National4-10.mdb",
"StaticRegionDB",0,"STRING",0,"",
"c:\Hwir\DataBase\SiteDB10-02-00.mdb",
"StopOnError",0,"INTEGER",0,"",
0,
"StopOnWarning",0,"INTEGER",0,"",
0,
"StorageLevel",0,"INTEGER",0,"",
0,
"Stream",0,"STRING",0,"",
"c:\Hwir\examsio.exe",
"Terrestrial",0,"STRING",0,"",
"c:\hwir\TF.exe",
"Time",0,"STRING",0,"",
"09:45am",
"Vadose",0,"STRING",0,"",
"c:\hwir\vadose.exe",
"WaterShed",0,"STRING",0,"",
"c:\hwir\ws.bat",
"WP",0,"STRING",0,"",
"c:\hwir\WP.bat",

(Note: The three parameters extracted from this file are in bold and italics.)

cpsr.ssf

1,
"cpsr.ssf","data group",
6,
"ChemCASID",1,"STRING",0,"",
1,"71-43-2",
"ChemKd",2,"FLOAT",0,"L/kg",
1,
8,2,3,4,5,6,7,8,9,
"ChemKDoc",1,"FLOAT",0,"mL/g",
1,0,
"NumChem",0,"INTEGER",0,"",
1,
"NumNegIon",1,"INTEGER",0,"",
1,0,
"NumPosIon",1,"INTEGER",0,"",
1,0,

(Note: The two parameters extracted from this file are in bold and italics. Also note that this is a truncated version of a cp.ssf file that was used for testing.)

8.4.1.3 Expected Results

US EPA ARCHIVE DOCUMENT

It is expected that the SST will use the HWIR IODLL to extract data from the header file and the chemical properties file. The SST should then write this data to a table called 'Study1' in a mysql database. The SST should also perform summary statistics on numeric data retrieved and report it in the table in either normal or log-normal space. The SST should also process the numeric data and return percentile information that is requested in the 'Summarytest1.txt' file. It should follow all its instructions through reading the 'Summarytest1.txt' file.

Specifically the SST should extract the following information from the files:

1. Variable CASID from the hdprod.ssf: 71-55-6
2. Number of chemicals from the hdprod.ssf: 2
3. Chemical names from the hdprod.ssf: Aniline and Benzene
4. Number of chemicals from the cp.ssf: 1
5. The Kd values from cp.ssf: 2,3,4,5,6,7,8,9
6. The lognormal summary statistics for the Kd values:
    a) number of parameters: 8
    b) mean of parameters: 4.95
    c) standard deviation of parameters: 1.675
    d) minimum value of parameters: 2
    e) maximum value of parameters: 9
7. The values above were calculated by hand in a spreadsheet. The spreadsheet is entitled 'SST_handcalc.xls' and it is included in the test package.
8. The normal summary statistics for the Kd values:
    a) number of parameters: 8
    b) mean of parameters: 5.5
    c) standard deviation of parameters: 2.449
    d) minimum value of parameters: 2
    e) maximum value of parameters: 9
9. The values above were calculated by hand in a spreadsheet. The spreadsheet is entitled 'SST_handcalc.xls' and it is included in the test package.
10. The values for the 25^{th}, 50^{th}, 75^{th}, 99^{th} percentiles: 4,6,8,9
11. The values above were calculated by hand.

8.4.1.4 Conducting the Test

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user's computers. The test bed files must also be installed on the tester's computer in order to follow this test plan and execute the tests. It is also required that the program 'MySGL-Front' be installed in order to follow these instructions.
2. The first thing you will need to do is to edit the 'Summary.bat' file. As noted in the input section (Section 8.4.1.2), you will need to change the executable path, folder locations, and computer name to adjust for your computer.
3. Double click on the 'Summary.bat' file to execute it. A DOS window will appear and the text we written out to the screen describes the work that is being done. When it is finished you will be prompted to hit any key to continue. Hit any key. The DOS window should close down.
4. Now use 'MySQL-Front' to view the results of the summary run. Double click on the shortcut to bring up the result database.

US EPA ARCHIVE DOCUMENT

5. You will be prompted to enter the connection to the MySQL Host. Use the default information and hit the ‘connect’ button to continue.
6. The name of the database where the results will reside is called ‘Study1'. Click on ‘Study1' in the tree-view menu on the left of the screen.
7. Under ‘Study1' click on ‘results’. Several new tabs should appear to the right. One of them is labeled ‘data’. This is the table that holds the results for the summary run.
8. The results are in a row and should correlate directly to the expected results listed in Section 8.4.1.3.
9. Close down the MySQL-Front program. This is the end of this test case.

8.4.1.5 Results

The SST used the HWIR IODLL to extract data from the header file and the chemical properties file. The SST wrote this data to a table called ‘results’ in the MySQL database called ‘Study1'. The SST performed summary statistics on numeric data retrieved and reported it in the table in both normal and log-normal space. The SST processed the numeric data and returned percentile information that was requested in the ‘Summarytest1.txt’ file. It followed all its instructions through reading the ‘Summarytest1.txt’ file.

8.4.2 SST02

8.4.2.1 Description

The purpose of this test case is to verify that the Site Summary Tool will create a results table in the specified database if one does not exist prior to the run. The exact same case that was run in test case SST01 will be run except that the results table will be deleted from the database prior to the run.

8.4.2.2 Input Data

The input for this test case is located in four files which are the ‘Summary.bat’, ‘Summarytest1.txt’, ‘hdprod.ssf’, and ‘cpsr.ssf’. The contents of the files are listed below. Note that changes to the ‘Summary.bat’ file will be necessary to ensure that executable and folder locations are accurate for the computer being used by the tester.

Summary.bat

```
"c:\Program Files\javasoft\JRE\1.3.1_03\bin\java" -cp MySQL.jar;. gov.epa.hwir.util.Summary
c:\hwir\parallel_version\Software\ssf c:\hwir\parallel_version\software\grf hdprod.ssf Summarytest1.txt
\\WP-TAIRAR\study1.mysql 5 1000
pause
```

As noted above, this file will need to be edited by the tester to ensure that executable and folder locations are accurate. The commands in the file will be broken down and it will be noted where changes should be made.

a) "c:\Program Files\javasoft\JRE\1.3.1_03\bin\java"

US EPA ARCHIVE DOCUMENT

This section describes the location of the java executable on the tester's computer and will need to be edited by the tester.

b) -cp MySQL.jar;. gov.epa.hwir.util.Summary

This section does not need to be edited.

c) c:\hwir\parallel_version\Software\ssf

This section describes the location of the SSF files on the tester's computer and will need to be edited by the tester.

d) c:\hwir\parallel_version\software\grf

This section describes the location of the GRF files on the tester's computer and will need to be edited by the tester.

e) hdprod.ssf

This is the name of the header file to be used and does not need to be edited.

f) Summarytest1.txt

This is the name of the instructions file and does not need to be edited.

g) \\WP-TAIRAR\study1.mysql

This is the name of the computer and table where results should be written to. The computer name will need to be edited by the tester.

h) 5

The number of times to retry running the tool.

i) 1000

The time to wait between retries in milliseconds.

Summarytest1.txt

```
variable,hd.ssf,CASID,,String,
nx,extract,hd.ssf,ChemCnt,,Integer,
variable,hd.ssf,Chems,,String,X,
nx,extract,cpsr.ssf,numChem,,Integer,
ny,8,
variable,cpsr.ssf,ChemKd,L/kg,Float,X,Y,
parameterize,lognormal,cpsr.ssf,ChemKd,L/kg,Float,X,Y,
parameterize,normal,cpsr.ssf,ChemKd,L/kg,Float,X,Y,
percentiles,4,25,50,75,99,cpsr.ssf,ChemKd,L/kg,Float,X,Y,
```


US EPA ARCHIVE DOCUMENT

end,

hdprod3.ssf

1,
"hdprod.ssf","data group",
62,
"Air",0,"STRING",0,"",
"c:\hwir\AirModel.exe",
"Aquatic",0,"STRING",0,"",
"c:\HWIR\afw.exe",
"Aquifer",0,"STRING",0,"",
"c:\hwir\aquifer1.exe",
"AT",0,"STRING",0,"",
"c:\hwir\AT.bat",
"CASID",0,"STRING",0,"",
"71-55-6",
"ChemCnt",0,"INTEGER",0,"",
2,
"Chems",1,"STRING",0,"",
2,"Aniline","Benzene",
"COP",0,"STRING",0,"",
"c:\hwir\DummyProc.bat",
"CPDirectory",0,"STRING",0,"",
"c:\hwir\CPPData",
"CW",0,"INTEGER",0,"",
-1,
"CWCnt",0,"INTEGER",0,"",
3,
"CWs",1,"STRING",0,"",
3,"5","3","1",
"Date",0,"STRING",0,"",
"06/19/2002",
"Debug",0,"INTEGER",0,"",
0,
"DSP",0,"STRING",0,"",
"c:\hwir\DummyProc.bat",
"EcoExposure",0,"STRING",0,"",
"c:\hwir\EE.exe",
"EcoRisk",0,"STRING",0,"",
"c:\Hwir\EcoRisk5.exe",
"ELP1",0,"STRING",0,"",
"c:\hwir\elp1.exe",
"ELP2",0,"STRING",0,"",
"c:\hwir\ELP2.exe",
"Farm",0,"STRING",0,"",
"c:\hwir\FarmFood.exe",
"GRFDirectory",0,"STRING",0,"",

US EPA ARCHIVE DOCUMENT

"c:\hwir\GRF",
"HumanExposure",0,"STRING",0,"",
"c:\hwir\exposure.exe",
"HumanRisk",0,"STRING",0,"",
"c:\Hwir\HRord63.exe",
"Lake",0,"STRING",0,"",
"c:\Hwir\examsio.exe",
"LastCw",0,"LOGICAL",0,"",
"T",
"LAU",0,"STRING",0,"",
"c:\hwir\LAU.bat",
"LF",0,"STRING",0,"",
"c:\hwir\LF.bat",
"MemoCnt",0,"INTEGER",0,"",
0,
"MetDir",0,"STRING",0,"",
"c:\hwir\Metdata",
"MMSP",0,"STRING",0,"",
"c:\hwir\MMSP.exe",
"NationDB",0,"STRING",0,"",
"c:\HWIR\DATABASE\National4-10.mdb",
"NewChem",0,"LOGICAL",0,"",
"F",
"NewRel",0,"LOGICAL",0,"",
"F",
"Permanent",0,"STRING",0,"",
"c:\hwir\Permanent",
"PSOFDirectory",0,"STRING",0,"",
"c:\hwir\PSOF",
"RealCnt",0,"INTEGER",0,"",
1,
"Realization",0,"INTEGER",0,"",
0,
"RegionDB",0,"STRING",0,"",
"c:\Hwir\DataBase\SiteDB10-02-00.mdb",
"RSOFDirectory",0,"STRING",0,"",
"c:\hwir\RSOF",
"SDP",0,"STRING",0,"",
"c:\hwir\sdpbeta2.exe",
"Seed",0,"INTEGER",0,"",
11031,
"SI",0,"STRING",0,"",
"c:\hwir\SI.bat",
"SiteBasedDB",0,"STRING",0,"",
"c:\Hwir\DataBase\SiteDB10-02-00.mdb",
"SiteCnt",0,"INTEGER",0,"",
2,
"SiteId",0,"STRING",0,"",

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

"",
"Sites",1,"STRING",0,"",
2,"0114001","0130207",
"SiteSurveyDB",0,"STRING",0,"",
"",
"Source",0,"STRING",0,"",
"",
"SrcCnt",0,"INTEGER",0,"",
5,
"Srcs",1,"STRING",0,"",
5,"LAU","WP","SI","AT","LF",
"SSFDirectory",0,"STRING",0,"",
"c:\hwir\SSF",
"StaticNationDB",0,"STRING",0,"",
"c:\HWIR\DATABASE\National4-10.mdb",
"StaticRegionDB",0,"STRING",0,"",
"c:\Hwir\DataBase\SiteDB10-02-00.mdb",
"StopOnError",0,"INTEGER",0,"",
0,
"StopOnWarning",0,"INTEGER",0,"",
0,
"StorageLevel",0,"INTEGER",0,"",
0,
"Stream",0,"STRING",0,"",
"c:\Hwir\examsio.exe",
"Terrestrial",0,"STRING",0,"",
"c:\hwir\TF.exe",
"Time",0,"STRING",0,"",
"09:45am",
"Vadose",0,"STRING",0,"",
"c:\hwir\vadose.exe",
"WaterShed",0,"STRING",0,"",
"c:\hwir\ws.bat",
"WP",0,"STRING",0,"",
"c:\hwir\WP.bat",

(Note: The three parameters extracted from this file are in bold and italics.)

cpsr.ssf

1,
"cpsr.ssf","data group",
6,
"ChemCASID",1,"STRING",0,"",
1,"71-43-2",
"ChemKd",2,"FLOAT",0,"L/kg",
1,
8,2,3,4,5,6,7,8,9,

"ChemKDoc",1,"FLOAT",0,"mL/g",
1,0,
"NumChem",0,"INTEGER",0,"",
1,
"NumNegIon",1,"INTEGER",0,"",
1,0,
"NumPosIon",1,"INTEGER",0,"",
1,0,

(Note: The two parameters extracted from this file are in bold and italics. Also note that this is a truncated version of a cp.ssf file that was used for testing.)

8.4.2.3 Expected Results

It is expected that the SST will use the HWIR IODLL to extract data from the header file and the chemical properties file. The SST should then write this data to a table called 'Study1' in a mysql database. The SST should also perform summary statistics on numeric data retrieved and report it in the table in either normal or log-normal space. The SST should also process the numeric data and return percentile information that is requested in the 'Summarytest1.txt' file. It should follow all its instructions through reading the 'Summarytest1.txt' file.

Specifically the SST should extract the following information from the files:

1. Variable CASID from the hdprod.ssf: 71-55-6
2. Number of chemicals from the hdprod.ssf: 2
3. Chemical names from the hdprod.ssf: Aniline and Benzene
4. Number of chemicals from the cp.ssf: 1
5. The Kd values from cp.ssf: 2,3,4,5,6,7,8,9
6. The lognormal summary statistics for the Kd values:
    a) number of parameters: 8
    b) mean of parameters: 4.95
    c) standard deviation of parameters: 1.675
    d) minimum value of parameters: 2
    e) maximum value of parameters: 9
7. The values above were calculated by hand in a spreadsheet. The spreadsheet is entitled 'SST_handcalc.xls' and it is included in the test package.
8. The normal summary statistics for the Kd values:
    a) number of parameters: 8
    b) mean of parameters: 5.5
    c) standard deviation of parameters: 2.449
    d) minimum value of parameters: 2
    e) maximum value of parameters: 9
9. The values above were calculated by hand in a spreadsheet. The spreadsheet is entitled 'SST_handcalc.xls' and it is included in the test package.
10. The values for the 25^{th}, 50^{th}, 75^{th}, 99^{th} percentiles: 4,6,8,9
11. The values above were calculated by hand.

8.4.2.4 Conducting the Test

US EPA ARCHIVE DOCUMENT

1. To proceed with this test, it is assumed that all of the required HWIR software and the Java runtime environment are loaded onto both of the user's computers. The test bed files must also be installed on the tester's computer in order to follow this test plan and execute the tests. It is also required that the program 'MySQL-Front' be installed in order to follow these instructions.
2. You will first need to edit the 'Study1' database. Double click on the 'MySQL-Front' shortcut.
3. You will be prompted to enter the connection to the MySQL Host. Use the default information and hit the 'connect' button to continue.
4. The name of the database where the results will reside is called 'Study1'. Click on 'Study1' in the tree-view menu on the left of the screen.
5. Under 'Study1' right click on 'results'. Select 'Drop Table'. When prompted again, select 'Yes'. The 'results' table should now be deleted.
6. Use the 'File' menu to shut down the database.
7. Next you will need to edit the 'Summary.bat' file. As noted in the input section (Section 8.4.2.2), you will need to change the executable path, folder locations, and computer name to adjust for your computer.
8. Double click on the 'Summary.bat' file to execute it. A DOS window will appear and the text we written out to the screen describes the work that is being done. When it is finished you will be prompted to hit any key to continue. Hit any key. The DOS window should close down.
9. Now use 'MySQL-Front' to view the results of the summary run. Double click on the shortcut to bring up the result database.
10. You will be prompted to enter the connection to the MySQL Host. Use the default information and hit the 'connect' button to continue.
11. The name of the database where the results will reside is called 'Study1'. Click on 'Study1' in the tree-view menu on the left of the screen.
12. Under 'Study1' click on 'results'. Several new tabs should appear to the right. One of them is labeled 'data'. This is the table that holds the results for the summary run. Note that this is the table that was deleted in step 5 above.
13. The results are in a row and should correlate directly to the expected results listed in Section 8.4.1.3.
14. Close down the MySQL-Front program. This is the end of this test case.

8.4.2.5 Results

The Site Summary Tool created a results table in the 'Study1' MySQL database when one did not exist prior to the run. The results were written to the table that was created.

US EPA ARCHIVE DOCUMENT

US EPA ARCHIVE DOCUMENT

9.0 Quality Assurance Program

The processors were developed under a quality assurance program documented in Gelston et al. (1998). In that program, quality is defined as the ability of the software to meet client needs. Meeting client needs starts with a shared understanding of how the software must perform and continues throughout the software lifecycle of design, development, testing, and implementation through attention to details.

Figure 9.1 outlines the software development process used for the processors, highlighting the quality check points. (Note: the processors activities flow down the left side of the figure because it is software developed for the first time, as opposed to a modification to existing software.) The process shown is designed for compatibility with similar processes used by other government agencies. For example, this quality process compares favorably with that in the EPA Directive 2182*, System Design and Development Guidance* (EPA 1997). It also compares favorably with the Office of Civilian Radioactive Waste Management's *Quality Assurance Requirements and Description, Supplement I, Software* (OCRWM 1995). Activities roughly equivalent across these processes are shown in Table 9.1.

Development of the processors included the implementation of a quality assurance checklist (see Figure 9.2). Understanding of this checklist by all team members resulted in the shared understanding of component requirements and design necessary to ensure quality. Completion of this checklist verified that all documentation was completed for transfer of the software to client use.

US EPA ARCHIVE DOCUMENT

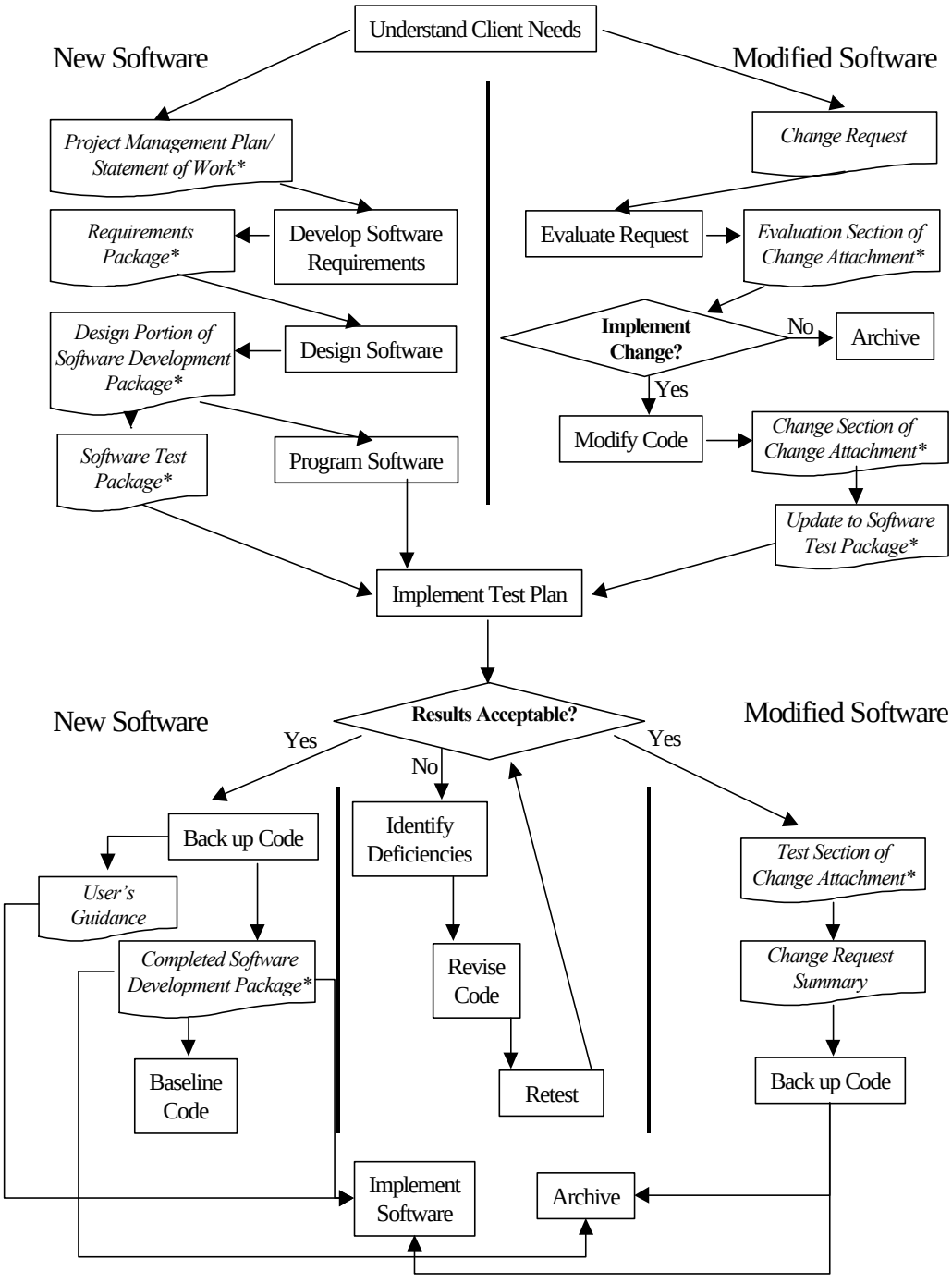

Figure 9.1 Ensuring Quality in the Environmental Software Development Process (* Indicates quality review stage; box with wavy bottom line and italics font indicates document versus activity)

Table 9.1 Relationship of PNNL Environmental Software Development Process to Quality Assurance Requirements (OCRWM 1995; EPA 1997)

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

(a) Note that OCRWM requirement I.2.8, Control of the Use of Software, is the responsibility of the OCRWM-related client.

(b) Elements 1 through 3 are generally completed by clients in the EPA before contract initiation with the project team.

(c) Verification includes informal code testing by software engineers to ensure that code functions as required.

(d) Validation includes testing by those other than the software engineers who developed the code to provide an independent confirmation that software functions as required.

(e) Note that some changes requested by clients may not be made in the software unless funding has been allocated for such modifications.

US EPA ARCHIVE DOCUMENT

A. General Requirements Analysis
--Documented in
_____Statement of Work (stored in project file; see Gene Whelan, Gariann Gelston, or current Integration Leader)
--Contains information on (all of the following)
_____problem description
_____deliverables
_____project team
_____capabilities to be used
_____restrictions
_____difficulties envisioned
_____compatibilities with existing software/hardware
_____scope of the project

B. Specific Requirements Analysis
--Documented in
_____requirements section of documentation (PNNL-11914, Volume 6, Section 2.0)
--Contains information on (all of the following)
_____purpose of the software
_____structure of the software
_____hardware and software requirements
_____input and output requirements
_____scientific basis
_____assumptions
_____limitations
_____post-October 31 requirements

C. Design Documentation
--Documented in
_____design portion of documentation (PNNL-11914, Volume 6, Section 3.0)
_____team task plans/Project Management Plan (stored in project file; see Gene Whelan, Gariann Gelston, or current Integration Leader)
--Contains information on (all of the following)
_____code type and description
_____development team members
_____specifications
_____logic diagrams
_____"help" descriptions
_____methods to ensure consistency in components
_____mathematical formulations
_____need for pre/post-processors
_____post-October 31 design elements

D. Development Documentation
--Documented in
_____Specifications Document (PNNL-11914, Volume 8)
_____Quality Assurance Archive (see Gariann Gelston or current Integration Leader)
--Contains information on (all of the following)
_____baseline hard copy of the source code
_____diskette copy
_____name of computer language(s) used

E. Testing Documentation
--Documented in
_____test plan that meets quality assurance requirements (PNNL11914, Volume 6, Section 4.0)
--Contains information on (all of the following)
_____description of software
_____testing scope
_____relationship between test cases and requirements
_____test activity description
_____hardware and software needed to implement plan
_____test case specifications
_____expected results

Figure 9.2 Quality Assurance Implementation Checklist for the Module Execution Manager

US EPA ARCHIVE DOCUMENT

F. User's Guidance

--Documented in

_____hard copy printout of user's guidance for system (PNNL-11914, Volume 11)

--Contains information on (all of the following)

_____description of software

_____description of use of user interface

_____mathematical formulations

_____example problems

_____explanation of modules included

G. General Quality Assurance Documentation

--Documented in

_____Quality Assurance Program Document (PNNL-11880)

_____Quality Assurance Software-Specific Checklist (PNNL-11914, Volume 6, Section 5.0)

--Contains information on (all of the following)

_____purpose of quality assurance program

_____client-specified activities

_____activities required to ensure quality in software

H. Quality Assurance Archive

--Documented in

_____hard copy files (see Gariann Gelston or current Integration Leader)

_____back up disk files in multiple storage locations (see Gariann Gelston or current Integration Leader)

--Contains information on (all of the following)

_____all quality assurance documentation

_____client correspondence regarding software

_____modifications made to baselined software

_____disk copy back ups

_____reproducibility of code (check code for comments)

Completed by _____ Date _____

Approved by
System/Module Manager _____ Date _____

Figure 9.2 Quality Implementation Checklist (contd)

US EPA ARCHIVE DOCUMENT

10.0 References

Documentation for the FRAMES-3MRA Technology Software System

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

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

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

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

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

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

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

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

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

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

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

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

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

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

US EPA ARCHIVE DOCUMENT

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

Quality Assurance Program Document

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

Additional References

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

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

US EPA ARCHIVE DOCUMENT

Distribution List

No. of Copies

OFFSITE

2 DOE/Office of Scientific and Technical Information

C. B. Nelson
U.S. Environmental Protection Agency
J.S. Building, Rm 3102S
Washington, D.C.

2 G. F. Laniak
U.S. Environmental Protection Agency
Environmental Restoration Lab
College Station Rd.
Athens, GA 30613

No. of Copies

ONSITE

33 Pacific Northwest National Laboratory

J. W. Buck	K6-80
K. J. Castleton	K6-80
G. M. Gelston (20)	K6-80
B. L. Hoopes	K6-80
R. E. Lundgren	K9-69
J. P. McDonald	K6-96
M. A. Pelton	K6-80
R. Y. Taira	K9-33
G. Whelan	K9-36
Technical Report Files (5)	

US EPA ARCHIVE DOCUMENT