Multimedia Integrated Modeling System User Guide

Prepared for:

William Benjey, Thomas Pierce, and Dan Loughlin
United States Environmental Protection Agency
National Exposure Research Laboratory
Office of Research and Development
Research Triangle Park, North Carolina

Prepared by:

Science Applications International Corporation
615 Oberlin Road, Suite 100
Raleigh, North Carolina 27605

Contract GS-35F-4461G
Task Order GST0403BF0247

December 2004

Table Of Contents

MIMS User Guide Overview

This User Guide introduces you to the Multimedia Integrated Modeling System (MIMS) Framework developed by the U.S. Environmental Protection Agency (EPA). The information within this document matches that found within the Help pages of the framework.

The User Guide is organized into sections:

- **Introduction** - discusses how the MIMS framework is organized and introduces the terms
- **QuickStart Tutorial** - step-by-step instructions develop sample applications within the MIMS framework
- **Terms and Concepts** - presents the language of MIMS and describes the functions of the MIMS members
- **How To Use MIMS** - acts as the guide to the individual MIMS screens, their menu options, and buttons
- **Parameter Types and Values** - details the use of various parameters within MIMS, how to choose the appropriate types, and how to assign them values
- **Appendix** - provides support information for the MIMS framework

Introduction

What is a Framework?

A modeling framework application provides an interface so that you can design and augment projects that involve many programs and files. It gives you a common interface to access similar files, libraries, and executable programs. A framework connects these elements together and allows you to ascertain the progress of multiple operations. By using the framework approach, investigators are encouraged to establish standard formats for sharing information.

The Multimedia Integrated Modeling System (MIMS) is a framework application that supports composing, configuring, and executing complex simulations involving multiple models and their associated tools (e.g., iterators and analysis tools). The MIMS framework directs the execution of models based on your designations of data dependencies.

The MIMS framework can enable compatible environmental models to communicate with one another, sharing input and output. As part of the MIMS design, you may pull models from different media (atmosphere, land, surface water, macrobiota, groundwater, etc.) into a single software application that enables you to study an entire ecosystem. By acting as a single framework, MIMS promotes the use of common data sets and fluxes to describe interactions at natural interfaces (air-surface water, surface water-groundwater, groundwater-macrobiota, etc.).

A MIMS Project serves as your library of building blocks (referred to as “members” in MIMS): data sets, executable programs, other tools, and operational sequences. The project’s members can be shared among both projects and users.

CAUTION: Executing a MIMS scenario invokes a variety of software that has full access to your computer, including the ability to modify and delete files and invoke additional programs. Only accept and use MIMS members (modules, projects, parameters, etc.) from a source you trust. Be sure to review all modules, formulas for parameters, formulas in control file templates, and module preprocessors from outside sources before executing a scenario.

Why Should I Choose the Framework Approach?

EPA's MIMS framework gives you a flexible way to integrate different tools (models, analysis tools, gridding tools, iterators, data sets, etc.) by building scenarios. A MIMS Scenario dictates how the tools communicate and how the tool parameters should be set. The ability to coordinate multiple models and tools is the central feature of the MIMS framework. Through this coordination, you will have the ability to:

- Visually represent the data flow involved in complex modeling efforts
- Copy the specifications for complex model runs and then modify the runs
- Control multiple programs through a graphical user interface and eliminate the need to edit scripts and control files
- Perform complex model runs in a Microsoft Windows environment and share efforts with users on other platforms
- Interface some models that use different time steps or geospatial grids (future application)
- Integrate different models more easily
- Ensure that the correct input values are exchanged between models when repeating a simulation
- Automate repetitive tasks, such as Monte Carlo uncertainty studies
- Integrate data analysis tools to execute during and at the conclusion of a simulation
- Allow new users to follow the course of previous investigators

What Are Some Sample MIMS Projects?

The MIMS framework is a useful tool for multimedia and many other modeling studies. A few examples of current applications are listed below:

1. To predict ambient ozone concentrations, air quality modelers first need to know pollutant emission rates. The Sparse Matrix Operating Kernel Emissions (SMOKE) system is designed to estimate emissions. SMOKE represents a series of MIMS modules whose outputs are used as input for the Community-Scale Model of Air Quality (CMAQ) modules. The CMAQ modules then predict the ambient ozone concentrations.
2. To determine the optimal design for urban drainage systems, investigators are coupling the Stormwater Management Model (SWMM) with nonlinear optimization tools under the MIMS framework. The SWMM module is evaluated by the optimization tools, and optimization tools feed back the most cost-effective proposal into the next run of SWMM. The MIMS ties that link one module's output to another's input are called Connections.
3. Dry deposition is one process that transfers pollutants from the atmosphere to land and water bodies. The dry deposition velocity is calculated from measurements of ambient ozone and acids on filters by the Clean Air Status and Trends Network (CASTNet). The application of the MIMS framework to the CASTNet analyses will make the CASTNet model testing easier in the future and will allow investigators to process data from multiple sites more quickly. Under the MIMS framework, a Scenario would describe the deposition to a particular area. A MIMS Domain Object represents a grouping of processes, modules, and parameters that all share a common theme or area of interest within a simulation. a media (e.g., atmosphere, soil, surface water, and plants) or media interface (e.g., atmosphere-to-surface water) where a process or behavior occurs, and the MIMS modules are associated with a particular domain object.

Publications describing the applications and design of the MIMS framework can be found at http://www.epa.gov/asmdnerl/mims/publications/index.html.

How Do I Install MIMS?

The MIMS framework relies on the operation of other publicly available software programs (Java Runtime Environment and R), so these programs must also be installed to operate the Analysis Engine. The MIMS Framework is designed to operate in the Java Runtime Environment (version 1.4 or higher). Below you will find instructions on:

Checking your current version of the Java Runtime Environment in Windows

Installing MIMS for Windows Systems

Installing MIMS for UNIX Systems

Installing R Statistical Software to use the Analysis Engine, Plot Page Object, Scatter Plot, and Time Series Modules

Checking your current version of the Java Runtime Environment in Windows:

Using the Windows **Start** menu:

1. Select the **Search/For Files or Folder** menu option.
2. Enter "java.exe" as the search criterion on your local drives, then click the **Search Now** button.

If you find the file, use a DOS prompt to put yourself in the file's directory. If you have more than one copy, check the one in the Windows or WINNT directory first.

If, for example, java.exe is in the folder c:\Windows\system\, you would type

```
> cd windows\system
```

at a command prompt (i.e., a DOS window). Note that in DOS, file names must be reduced to 8 characters or fewer.

Then type:

```
> java -version
```

at the command prompt.

Installing MIMS for Windows Systems:

1. Install the Java 1.4 Java Runtime Environment (for non-developers) or Java Development Kit, available for free from Sun at http://java.sun.com/j2se/. Note that some problems have been observed with Java 1.4.1 on Windows operating systems, and Java 1.4.2 is recommended.
2. Download the MIMS framework installation file from http://www.epa.gov/asmdnerl/mims/software/index.html.
3. Execute the downloaded zip file (mims_rel.exe) or run PKUNZIP on (mims_rel.zip) to extract the MIMS files.
4. Edit the batch file bin\runMIMS.bat to set the MIMSHOME variable to the directory where you installed MIMS (default is c:\mims).
5. Put Java 1.4 (or higher) on your path (using system properties on NT; this will not take effect until after you reboot) or edit runMIMS.bat to explicitly refer to the Java and R executables. The Java installations from Sun contain a bin directory, which contains the Java executable.
6. On Windows 98/Me, set the memory properties of the runMIMS.bat file so the initial environment has at least 2560 bytes of memory. This property does not need to be set with other versions of Windows.
7. Double-click runMIMS.bat or its shortcut (Windows 98/Me) to start the MIMS framework.
8. An example scenario is included, and on-line Help is available.
9. See the QuickStart Tutorial in the on-line Help for an example using MIMS.

Installing MIMS for UNIX Systems and MacOS:

1. Have your system administrator install the Java 1.4 Java Runtime Environment (for non-developers) or Java Development Kit, available for free from Sun at http://java.sun.com/j2se/.
2. Download the MIMS installation file from http://www.epa.gov/asmdnerl/mims/software/index.html.
3. Expand and untar the downloaded compressed tar file.
4. Edit the shell script file bin/runMIMS.sh to set the MIMSHOME variable to the directory where you installed MIMS (default is ${USER}/mims). For MacOS installations, the path should be edited to "/Users/${USER}/mims" to take advantage of the usual installation path. Also make sure that the path to the Java executable (variable "JAVA") is correct in the runMIMS.sh script.
5. Put Java on your path or set the JAVA_HOME environment variable.
6. Execute "runMIMS.sh" to start the MIMS framework.
7. An example scenario is included, and on-line Help is available.

Installing R Statistical Software:

The R software provides options for plot types, layouts, and output formats not found in the MIMS framework alone. The framework object "Plot Page Object" and the modules

"Scatter Plot Module" and "Time Series Module" will only work if R has been installed and is in the runMIMS.bat path.

1. Click "Download CRAN" at http://www.r-project.org/ and choose the appropriate link and operating system. Then download the base version of the R statistical software. The R versions 1.9.1 and 2.0.1 have been successfully used with MIMS.
2. Read the Readme file for instructions on installing the package. Then install the base R package according to those instructions. Note the directory where the package is installed.
3. Open the runMIMS.bat file with a text editor (e.g., Microsoft WordPad). In the line following the statement "`add location of R executable to PATH (used for generating plots)`," be sure that the directory path matches the one you specified in Step 2. Save and close this file.

For on-line Help within MIMS, select the **Help/User Guide** menu option.

What Are the First Concepts To Learn about the MIMS Framework Structure?

The MIMS framework contains several hierarchical levels with which you must become familiar before constructing a full project. After designing a project once, you will probably be familiar enough with the framework structure to navigate through the MIMS windows.

Following the chapter on the QuickStart tutorial, the User Guide presents a chapter on Terms and Concepts within the MIMS framework. That section provides more details on the concepts described below and highlights additional terms.

A MIMS Project serves as your library of building blocks (referred to as “members” in MIMS): data sets, executable programs, other tools, and operational sequences. Within a MIMS project, you can define the members, their file paths, and their associated parameters. All of this information is then available for you to create different simulations.

A MIMS Scenario defines the operational sequence and settings for a given simulation. The MIMS Scenario window lets you follow the progress of a simulation as each member's operation is executed.

In the examples presented below, scenarios are usually composed of MIMS Domain Objects. A domain object is a grouping of processes, modules, and parameters that describe one area of a simulation. The most common domain objects represent a particular media (e.g., atmosphere, soil, surface water, plants, and exposed populations) or a natural interface (e.g., atmosphere-to-surface water) when the domain object describes fluxes between media. A scenario generally contains one or more domain objects.

One or more actions may occur within each domain object. The actions are defined as MIMS Processes. In most examples, the MIMS processes involve an action by the computer (running code, reading/writing data, displaying information, etc.). A Process often requires input information and generates output.

Many MIMS processes are implemented by executing code, a MIMS Module. A MIMS module is essentially the device that would be operated to accomplish a particular task. An internal module may be implemented with Java code that is called by MIMS, and an external module represents a stand-alone program that may be invoked by MIMS. A single application of a module to implement a MIMS process within a particular scenario is termed a Module Instance.

Scenarios, Domain Objects, Processes, and other MIMS members may have MIMS Parameters associated with them. Parameters refer to the information that a MIMS process needs to complete the simulation.

If the output from one process serves as the input to another, then you must define a Connection within the scenario. Connections allow you to track how one process feeds into another. In MIMS, you do not define the sequence of operations explicitly. Instead, MIMS determines the order of events based on the connections.

In the Introduction, you are introduced to several events to illustrate how modules, processes, and domain objects fit together. The table below may be used as a reference to recall the nomenclature.

Event	Module	Process	Domain Object
CMAQ predicts ozone concentrations in the atmosphere.	CMAQ	Predicts ozone concentrations	Atmosphere
The rake creates piles in the yard.	Rake	Create piles	Yard
The bags transport leaves in the yard.	Bags	Transport leaves	Yard
The cream treats blisters as a first aid measure.	Cream	Treats blisters	First aid (framework object)
The Dep_Module simulates dry deposition in the atmosphere.	Dep_Module	Simulates dry deposition	Atmosphere
The Time Series Plotter graphs the time series for the analysis.	Time Series Plotter	Graphs the time series	Analysis

Publications describing the design and applications of the MIMS framework can be found at http://www.epa.gov/asmdnerl/mims/publications/index.html.

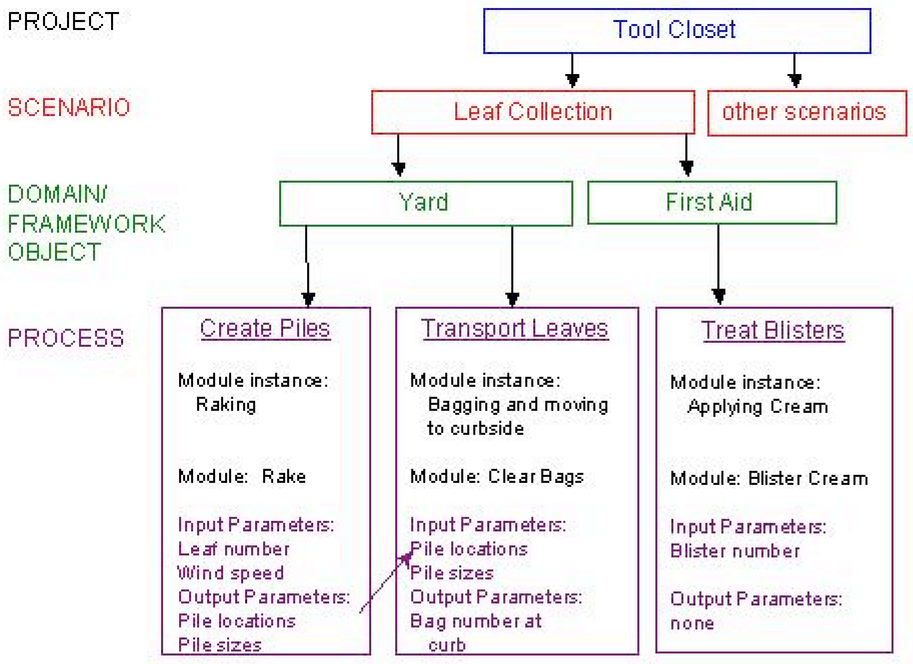
Is There an Analogy To Help Me Understand the MIMS Framework Structure?

A simple analogy for the framework would be to consider a MIMS Project as a tool closet. An organized tool closet may have different shelves for yard work tools, painting applicators, plumbing supplies, etc.

Most shelves may be regarded as MIMS Domain Objects, having only one place (media) where their tools would be useful. One would use a rake from the yard work shelf outdoors but never consider using a rake in the kitchen. Domain objects are typically associated with a single media. Different tools may be available to accomplish the same task (e.g., both a rake and a leaf blower), and any could be used in the yard.

The rake found on the yard work shelf would be analogous to a MIMS module. The rake is a device that can be used to accomplish a particular task. In MIMS, an individual task is described as a process.

The Create Piles process is associated with the rake as a tool. Use of a leaf blower would also fall under the Create Piles process, but the tool (module) would change. In MIMS, the rake creating piles in the yard one time would be defined as a module instance. The module instance may be influenced by different parameters (e.g., the number of leaves, the wind speed, and the desired location for the piles). In MIMS, Parameters are associated with every module instance. In MIMS, you would define the input parameters (number of leaves and wind speed) or specify a file to read. The output parameters (leaf pile locations and sizes) would be the results from the module instance.


PROJECT
Tool Closet
SCENARIO
Leaf Collection
other scenarios
DOMAIN/ FRAMEWORK OBJECT
Yard
First Aid
PROCESS
Create Piles
Module instance: Raking
Module: Rake
Input Parameters: Leaf number Wind speed
Output Parameters: Pile locations Pile sizes
Transport Leaves
Module instance: Bagging and moving to curbside
Module: Clear Bags
Input Parameters: Pile locations Pile sizes
Output Parameters: Bag number at curb
Treat Blisters
Module instance: Applying Cream
Module: Blister Cream
Input Parameters: Blister number
Output Parameters: none

After raking, the leaves must be bagged and transported to the curb. This particular sequence of events gives the homeowner the desired result and cannot be performed in any other order. Within MIMS, module instances are performed in a particular order as defined by the MIMS Scenario. The scenario approach lays out all of the tasks that must be performed and their sequence. For tasks more complicated than raking, the homeowner may choose to keep a copy of a sequential checklist in the tool closet, much the same way that MIMS users store scenarios within a MIMS project.

The output from the Create Piles process was the pile of leaves, and this pile represented the input to the Transport Leaves process. Within MIMS, this relationship between the output of one module and the input for another is represented by a Connection. The sequence of the connections determines the order that calculations for a scenario should be performed.

Finally, some shelves in the tool closet contain items that may be used in different places but still have a focused purpose. For example, a first aid supply shelf in the tool closet may contain the blister cream that needs to be applied after the raking is completed. The first aid shelf would represent a MIMS Framework Object, and the blister cream container would be another MIMS module. Application of the blister cream would constitute a MIMS process. MIMS framework objects could include modules that are scheduling programs, Monte Carlo tools, visualization tools, and analytical programs. Framework objects generally do not represent processes within the simulation but instead support the simulation or its analysis.

How Are Parameters Associated?

In the Introduction's analogy, the Leaf Pile Locations parameter is an output parameter from the Raking module instance and an input to the Bagging module instance. Since the leaf piles are also associated with the Yard and the Leaf Collection, the Leaf Pile Locations parameter could have been considered a parameter of the Yard domain object and the Leaf Collection scenario.

The MIMS framework allows parameters to be passed between members within a scenario through four functions:

Icon	Function Name	Function
⊞	Add to Scenario	Copy a parameter from a module or object within a scenario into the scenario parameters table
Right click	Add Parameter Source	Allow the parameter value from one object to be used in another object
-■■-	Connect to Parameter	Allow the parameter value from one object or module to be used in another module or in the scenario
⊛	Copy from Module	Copy a parameter from a module within an object into the object's parameters table

When building parameter connections, it is helpful to understand how the variables are passed through a scenario after the connections are made. The following values are passed when you execute the scenario (for the case where the module instance is associated with that object's processes):

- If an *input* parameter value is specified in the scenario parameters table and is connected to an object or to a module instance, the object or the module instance parameter assumes that value.

- Similarly, if an *input* parameter value is specified in the object parameters table and is connected to a module instance, the module instance parameter assumes that value.
- If an *output* variable is specified at the conclusion of a module instance and is connected to an object or scenario, the object or scenario parameter assumes that value.
- If an *output* variable is specified by an object and is connected to the scenario, the scenario parameter assumes that value.

The Source column of the parameter tables lets you know the source (module, object, and scenario names) of their parameter values. Many module instances will also need to draw parameters from objects other than their own. The **Connect to Parameter** (⊶) icon allows you to feed input parameters to a module directly from the output parameters of other objects and from the scenario.

At the end of the QuickStart tutorial, you will encounter an example where the Dep_Monte scenario is nested as a module within the Scen_Monte Carlo scenario. In this case, the Scen_Monte Carlo scenario exchanges the parameters listed in the Dep_Monte scenario parameters table as the input and output. The outer scenario cannot read any of the other parameters within Dep_Monte.

What is the Goal of the Example Shown in the QuickStart Tutorial?

To familiarize you with some of the MIMS concepts and capabilities, a simple demonstration simulation has been included that uses a dry deposition model. This demonstration is likely not as extensive as a real application of the framework would be but is sufficient to illustrate the most important concepts. Only a Windows version of the model has been provided with this distribution. Forward (**>>**) and back (**<<**) buttons at the top of each page of the tutorial enable easy navigation in on-line Help.

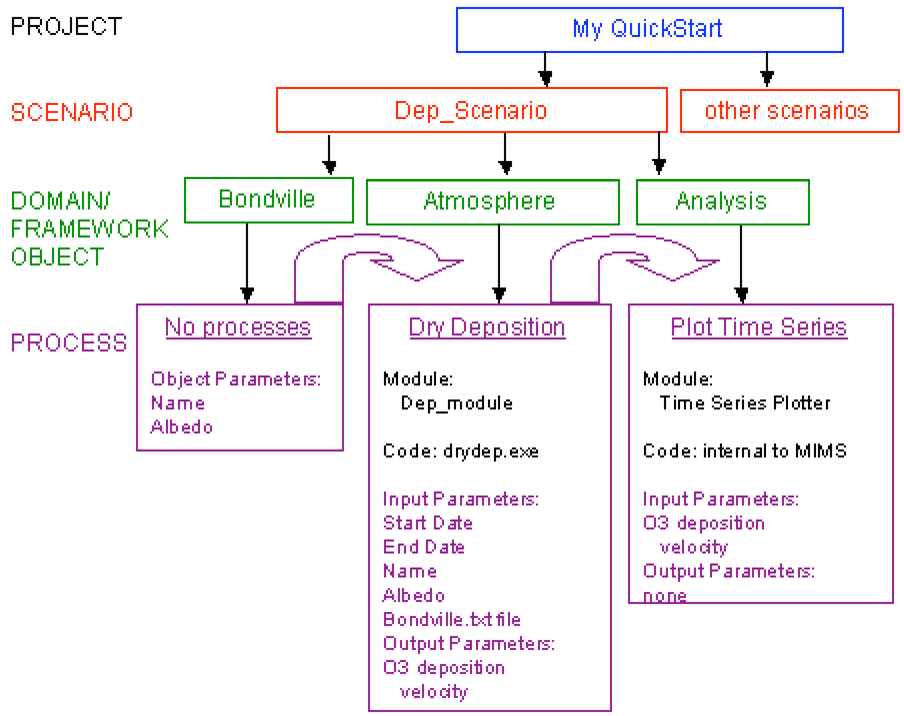
The dry deposition model computes the deposition velocity for selected trace gases. The deposition velocity is proportional to the flux (mass per unit area per time increment) of a chemical to the land surface. The deposition velocity depends on the state of the atmosphere, the plant canopy, the soil, and chemical properties.

The model that is used is a development version that incorporates the Multilayer Model (Meyers, T.P. et al. J. Geophys. Res., 103(D7), 22,645-22,661, 1998) and the Multilayer Bio-Chemical Dry Deposition Model (Wu Y., B. Brashers, P. L. Finkelstein, and J. E. Pleim, A Multilayer Bio-Chemical Dry Deposition Model, submitted to J.Geophys. Res.). The model and its data sets are in the QuickStart directory within the MIMS directory you installed. The model reads and writes data in a tab-delimited format, as seen in the file bondville.txt in the data subdirectory. The model also reads a text file that describes what computations should be performed. An example of this file is bond.input. It contains start and end dates, file names, control flags, and parameter values.

In this framework application, you will assemble the members of a MIMS project called “My QuickStart.” Two MIMS processes will be created to accomplish the tasks:

1. Dry Deposition: this process calculates the ozone deposition rates using the external dry deposition module
2. Plot Time Series: this process will display the calculated deposition rates versus time by using the internal module called Time Series Plotter

The diagram below illustrates the connections and nesting of the project members. Straight arrows connect to the members associated with a particular project, scenario, or object. The U-turn arrows show that parameters are passed between domain objects and into the next module instance.



The Bondville domain object holds some input parameters but does not execute any processes. The Dry Deposition process is located in the Atmosphere Domain Object and has the type Atmosphere. The Plot Time Series process appears in the Analysis Framework Object and has the type Analysis.

You are interested in the data contained by the Bondville Domain Object serving as the input to the Dry Deposition process, so a connection will be established in that direction. After the deposition fluxes are calculated, the Time Series Plotter module will display the results. Therefore, a second connection will be required that carries output from the Dry Deposition process to the Time Series Plotter. This entire sequence will be controlled from the Dep_Scenario scenario window.

After you create a time series plot, the QuickStart tutorial goes onto describe a way to repeat the scenario 30 times as part of a Monte Carlo simulation. The Monte Carlo simulation considers the uncertainty of plant cover values and shows how the maximum ozone deposition velocity changes with different values.

A sample MIMS project called Demo QuickStart has been provided to show what the completed example should look like. If you open the "deposition" scenario in that project, set the scenario parameter Default Directory to point to the QuickStart directory on your machine, and reset the scenario (all of those steps are described in the following chapter), you should be able to execute the scenario in the Demo QuickStart project.

How Do I Complete the QuickStart Tutorial?

How Do I Complete the QuickStart Tutorial?

The programs for the QuickStart Tutorial are currently only available for Microsoft Windows users. However, other users can also learn the methods for creating their own MIMS projects by following this guide. If working in a UNIX environment, remember to change "\" to "/" when specifying file paths.

The MIMS framework opens a new window for every MIMS Project, Scenario, Domain Object/Framework Object, Module, and Module Instance. In this tutorial, you will become accustomed to creating, organizing, and dismissing these windows.

Since MIMS records the project information whenever you close a window, it is not necessary to save your progress as you proceed. However, to prevent loss of your work from a power or system failure, you should choose the **File/Save** menu item on the Scenario window after any significant effort.

In this tutorial, you will begin a new project. First you will add the deposition model to the project as a new module and then assign that module to a new domain object. You will create a second domain object describing the site data and then connect the two together with a scenario. You will execute the scenario and then add another domain object to display your results.

To begin MIMS, click on the MIMS icon to run the runMIMS.bat file. The *MIMS - Project Selection Window* appears.

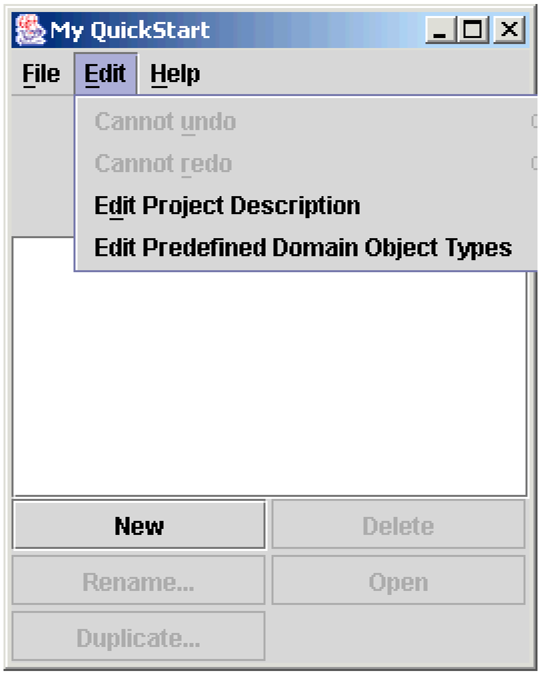
Create the My QuickStart Project (QuickStart)

You are going to create a new MIMS project and name it "My QuickStart". This project will contain all of the members associated with the tutorial.

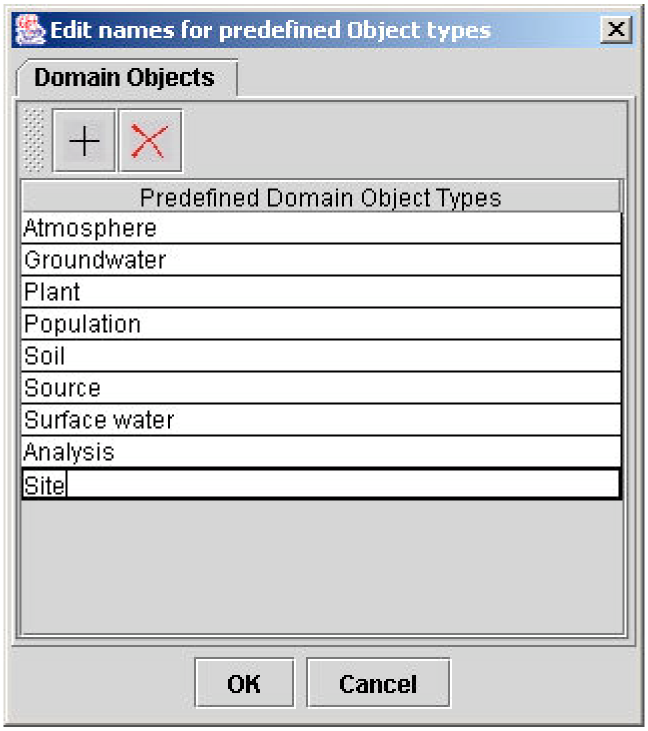
1. On the *MIMS - Project Selection* window, select the **File/New Project** menu item (i.e., click on the **File** menu item and then select **New Project** from the drop-down menu).

2. When prompted for the Project name, type “My QuickStart” on the *Create Project* dialog box and then click **OK**. The *My Quickstart* MIMS project window appears and shows the contents of your project, currently empty. The *Category* list near the top of the window shows the member categories in the project and is currently empty for this new project.

3. Select the **Edit/Edit Predefined Domain Object Types** menu item. The *Edit names for predefined Object types* dialog box appears with several rows listing common media that are represented in MIMS (e.g., Atmosphere, Groundwater, Plant, and Population). The domain object type is used to determine which modules and parameters may be connected with a particular domain object.



4. Click on the **Add New Parameter** (+) icon to add a new row. Double-click the mouse on the blank row and then type "Analysis" in the blank. Repeat the process to create an Object Type named "Site." Click the **OK** button to dismiss this dialog box.



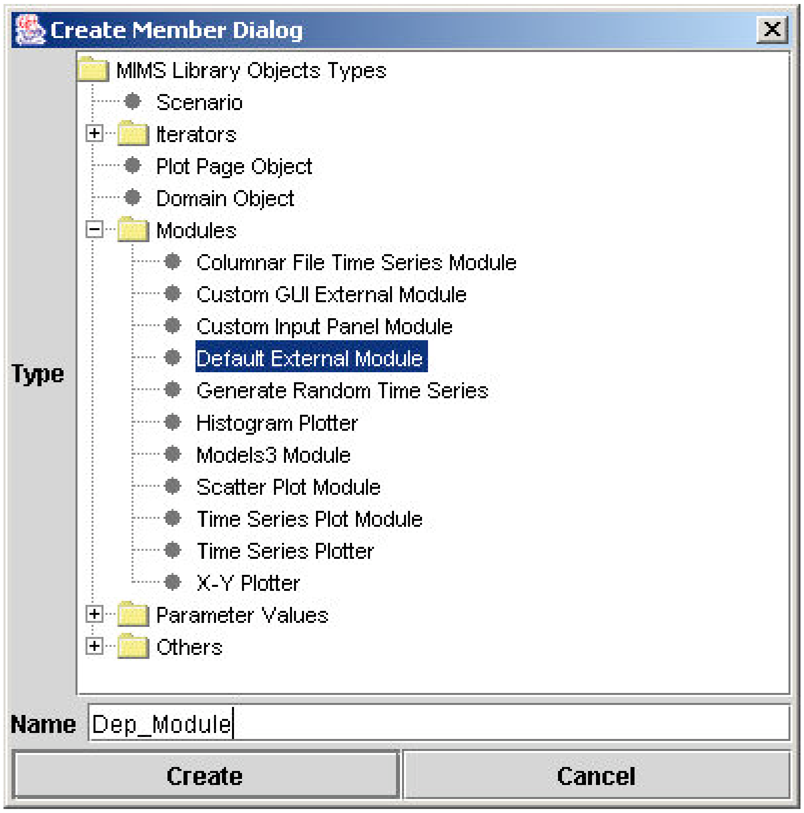
Predefined Domain Object Types
Atmosphere
Groundwater
Plant
Population
Soil
Source
Surface water
Analysis
Site

Create the Dep_Module MIMS Module (QuickStart)

You will first create the deposition module. Because MIMS allows you to copy information from a module into a domain object, this sequence will save some effort in later steps.

1. From the *My QuickStart* window, click on the **New** button to invoke the *Create Member Dialog*.

2. From the Modules folder, select Default External Module, indicating that you plan to use a program external to the MIMS framework. Type “Dep_Module” in the *Name* field as the name for the module, then click the **Create** button.



Type: Default External Module

Name: Dep_Module

3. The *New Members Created* dialog box will then appear. Click **Yes** to open the *Module [Dep_Module]* window where you may enter the module commands, classification, description, parameters, etc.

4. In the *Execution Command* field, type “bin\drydep.exe –b config.txt” to describe the partial directory path (bin\), program name (drydep.exe), and command line

arguments (-b config.txt). You will define the full default directory path when you create the scenario.

5. Beside *Control Files*, click the **Add** button. The MIMS control file acts as a template for the model's control file. When prompted by the *Create Control File* dialog box, name the control file config.txt, then click **OK**.

6. Click the *Control Files* **Edit** button on the *Module [Dep_Module]* window. This button opens the *Control File Editor* dialog box.

7. Instead of typing the entire control file for the drydep.exe program in the *Control File Editor* dialog box, this tutorial provides you with the file bond.input in the QuickStart directory. Open bond.input using Microsoft WordPad and copy all of the text in that file. (Highlight the text and then press Ctrl-C to copy the selected text.)

   Paste the text (using Ctrl-V) into the text box on the *Control File Editor* dialog box. The text in the bond.input file was constructed to perform a specific simulation, but you will modify the text in the following steps to read several parameters from MIMS members instead. To read the parameters from within the framework, you need to use the MIMS parameter substitution syntax (described further under Calling Parameters).

8. On the *Control File Editor* dialog box, replace the text "08/17/94" with "${getDate("Start Date","MM/dd/yy")}". Note: The command is case-sensitive. The getDate method will return the value of a date parameter as a formatted string. Next replace "10/01/94" with "${getDate("End Date","MM/dd/yy")}" to instruct MIMS how to insert the end date into the control file.

9. Next, scroll down in the *Control File Editor* to the File Locations and Names Section of the config.txt file. In the first line, replace "bond" with "${str("Name")}" beside the title Station Name. In the second line, replace "data\bondville.txt" to read "${str("State")}". Also replace "data\bond_v25.txt" in the third line with "data\output.txt".

   The first 13 lines should now read:

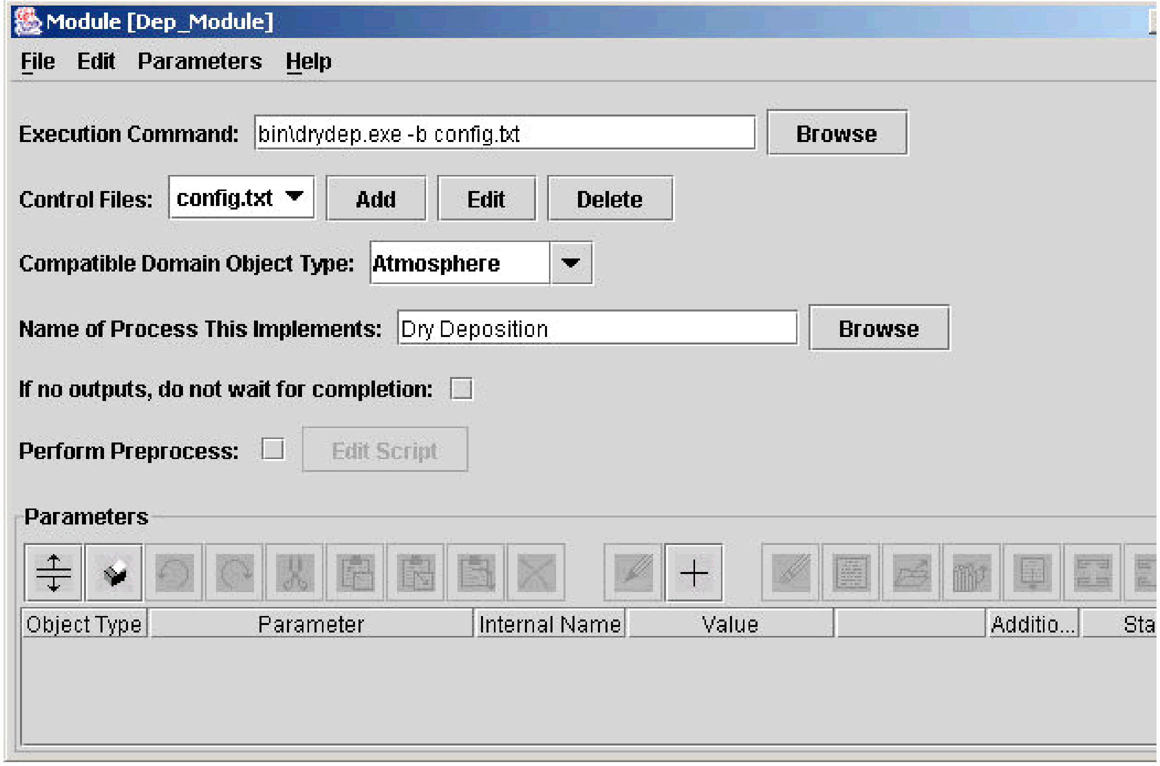
```
========= Input file for program DRYDEP.  Spaces don't matter. =====
25        Input file version number
\         Character used in filename parsing, "/" for unix, "\" for Windows
${getDate("Start Date", "MM/dd/yy")}  Start Date
${getDate("End Date", "MM/dd/yy")}  End Date
-15341 DATETIME Offset: 18993 for Statistica & Quattro, -15341 for jday (94)
BAB       Initials of operator
Comments (entire next line):
Berry model, output only good O3 data
=== File locations and names section =======================
${str("Name")}            Station Name
${str("State")}     Input Data File, blank to use Sonic/Campbell data
data\output.txt      Output Data File
```

10. Next scroll down to the last line of the User's switches section. Replace the text "0.10" with "${str("Albedo")}" to instruct the control file to accept the Albedo parameter. Albedo is the fraction of light that is reflected by a body or surface.

    Click **OK** to close the *Control File Editor* dialog box.

11. The compatible domain object type determines which domain objects will be able to use this module. In the *Compatible Domain Object Type* field of the *Module [Dep_Module]* window, enter "Atmosphere".

12. In the *Name of Process This Implements* field, type "Dry Deposition".



Module [Dep_Module]

Execution Command: bin\drydep.exe -b config.txt

Control Files: config.txt

Compatible Domain Object Type: Atmosphere

Name of Process This Implements: Dry Deposition

If no outputs, do not wait for completion: □

Perform Preprocess: □

Finally the Dep_Module module needs some parameters.

Adding Parameters to the Dep_Module Module (QuickStart)

Follow the steps below to add six parameters to the Dep_Module module.

1. To add a new parameter, click on the **Add New Parameter** (+) icon or select the **Edit/Add New Parameter** menu item on the *Module [Dep_Module]* window to invoke the *New Parameter Type* dialog box.

2. The *New Parameter Type* dialog box provides a list of parameter choices. Pick a parameter type listed in the table below, then click **OK** to invoke the *Enter Name* dialog box. Enter the parameter name listed in the second column of the table below. Click **OK**.

   The *Parameter Properties* dialog box is presented next. Based on the table below, set the following parameter properties in this dialog box: Domain Object Type, Is Local/Input/Output, and Required. Leave the *InternalName*, *Domain Object Name,* and *Formula* fields as they appear and the *Is ReadOnly* check box unchecked for all parameters. Click **OK**. Return to Step 1 and enter the next parameter listed in the table until all the parameters in the table have been entered.

   The Start Date and End Date parameters should be automatically assigned to have formats of MM/dd/yyyy HH:mm:ss (as the formatting appears in the Parameters table).

Parameter Type	Parameter Name	Domain Object Type	Is Local/ Is Input/ Is Output	Not Required/ Required/ By Formula
Date parameter	Start Date	<leave blank>	Is Input	Required
Date parameter	End Date	<leave blank>	Is Input	Required
String parameter	Name	Site	Is Input	Required
Floating point parameter	Albedo	Site	Is Input	Required
File parameter	State	Atmosphere	Is Input	Required
Floating point time series (generic)	O3 Deposition Velocity	Atmosphere	Is Output	Required

Note: If you make a mistake while creating the parameters, select the **Edit/Undo Add Parameter Value** menu item. Changes cannot be made to the above attributes after the parameter is created; you need to first delete the parameter. Highlight it, click the **Delete Parameter** (✕) icon, then begin again.

3. After the last variable has been entered, highlight the Name variable row in the table, then select the **Parameters/Edit Parameter Type** menu item to invoke the *Edit Name* dialog box. Enter "40" in the *Maximum Length* field. Click **OK**.

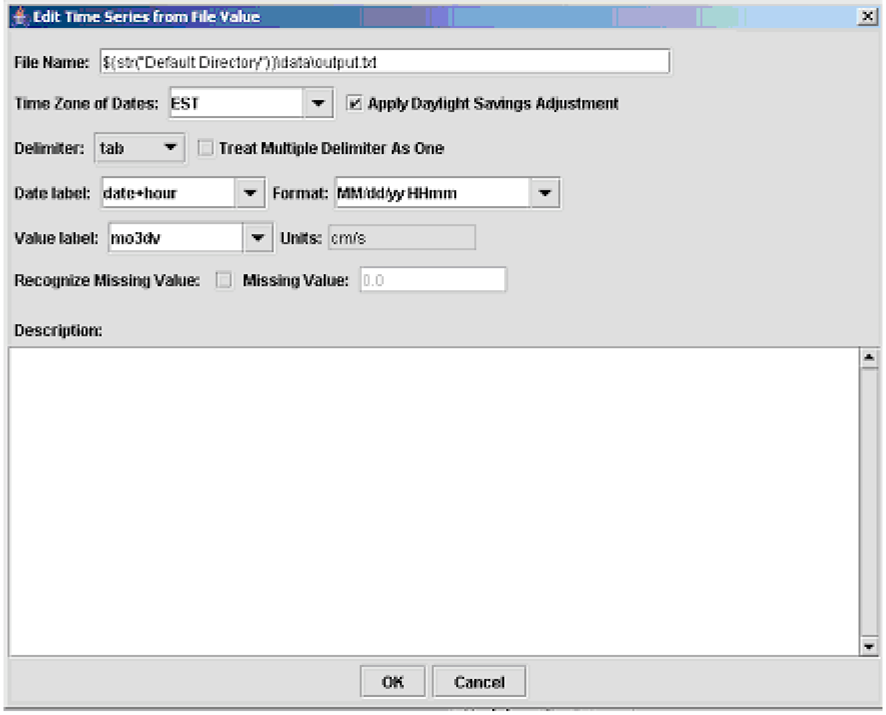
   Highlight the Albedo variable row in the table, then select the **Parameters/Edit Parameters Type** menu item to invoke the *Edit Albedo* dialog box. Enter "na" in the *Units* field since albedo is dimensionless. Click **OK**.

Highlight the O3 Deposition Velocity variable row in the table, then select the **Parameters/Edit Parameters Type** menu item to invoke the *Edit O3 Deposition Velocity* dialog box. Enter "cm/s" in the *Units* field. Click **OK**.

4. In the O3 Deposition Velocity variable, the Additional Type Info column contains a button called **Set**. Click **Set** to invoke the *Select Type* dialog box with a list of type of time series; choose **Time Series from File Value**, click **OK**.

5. Then the *Edit Time Series from File Values* dialog box appears that allows you to edit the time series parameter. In the *File Name* field, type "${str("Default Directory")}\data\output.txt" to specify the name and location of the output file. In the *Time Zone of Dates* field, choose "EST" (Eastern Standard Time) as the time zone from the drop down list and check the *Apply Daylight Savings Time Adjustment* check box.

   The remaining information describes the file format. Leave the *Delimiter* field as "tab" and uncheck the *Treat Multiple Delimiter As One* check box. Enter "date+hour" in the *Date label* field and type "MM/dd/yy HHmm" in the *Format* field (this format is not in the pulldown menu). Enter "mo3dv" in the *Value label* field to indicate the column label for the O3 deposition velocity. Leave the *Recognize Missing Value* check box unchecked. Click **OK** to close the *Edit Time Series from File Values* dialog box.



Edit Time Series from File Value

File Name: ${str("Default Directory")}\data\output.txt

Time Zone of Dates: EST ☑ Apply Daylight Savings Adjustment

Delimiter: tab ☐ Treat Multiple Delimiter As One

Date label: date+hour Format: MM/dd/yy HHmm

Value label: mo3dv Units: cm/s

Recognize Missing Value: ☐ Missing Value: 0.0

Description:

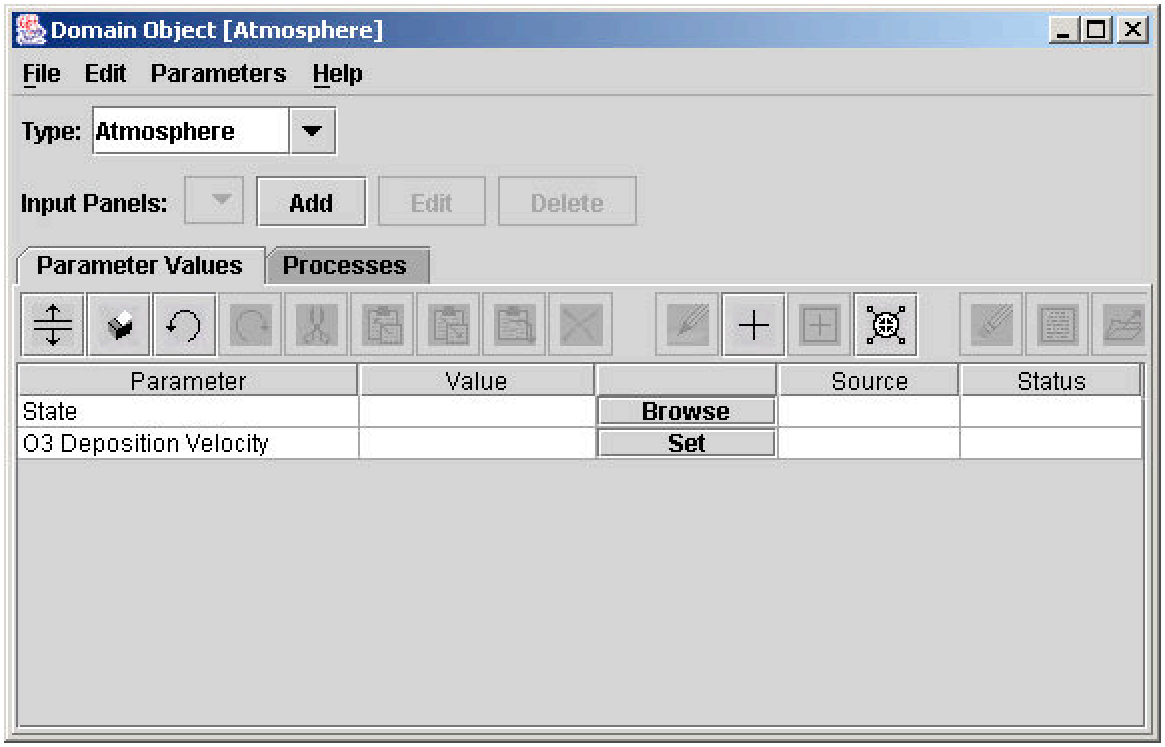
6. Close the *Module [Dep_Module]* window by choosing the **File/Close Module** menu item.

Create the Atmosphere MIMS Domain Object (QuickStart)

First you are going to create the Atmosphere domain object. This domain object will house the Dry Deposition process in the scenario. During the creation, two parameters (State and O3 Deposition Velocity) will be copied from the Dep_Module module into the Atmosphere domain object.

1. In the *My QuickStart* window, click **New** to invoke the *Create Member Dialog.* Select Domain Object from the tree, enter "Atmosphere" in the *Name* field, then click **Create**. The *New Members Created* dialog box is presented prompting you to open a new member, "Atmosphere". Click **Yes** to open the *Domain Object [Atmosphere]* window.

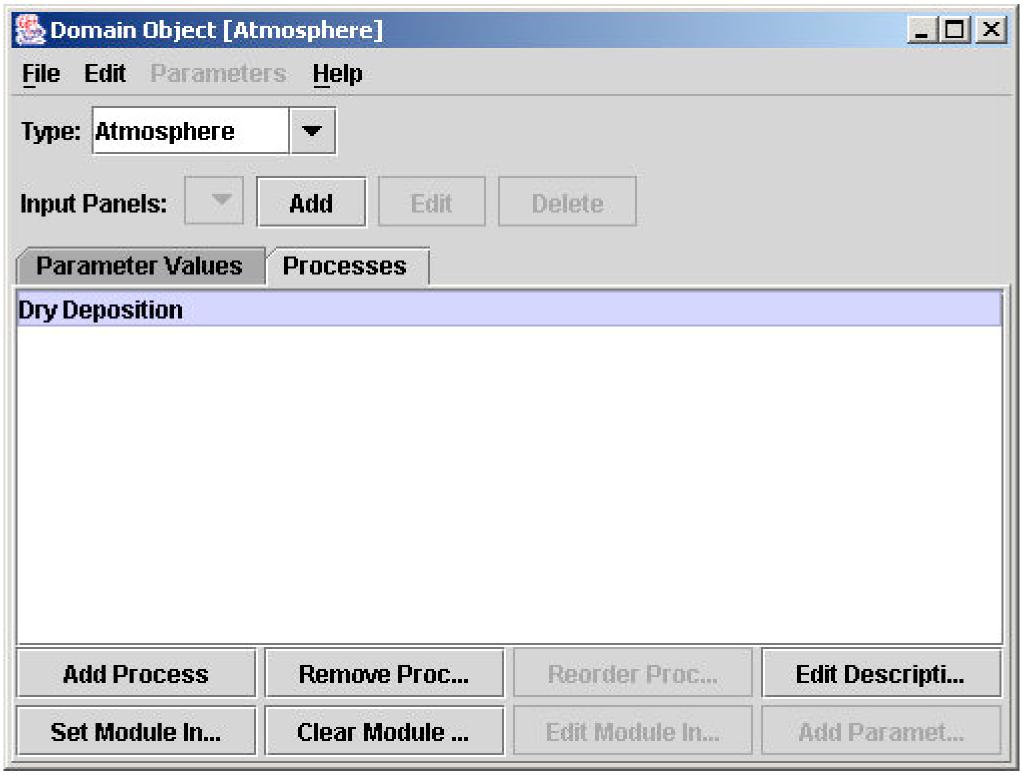
2. Enter "Atmosphere" in the *Type* field. Select the **Parameters/Copy from Module** menu item to invoke the *Select Module* dialog box. Choose Dep_Module and then click **OK**. This action copies into the domain object all relevant parameters (State and O3 Deposition Velocity) from the Dep_Module module instance.



Type: Atmosphere

Parameter	Value		Source	Status
State		Browse		
O3 Deposition Velocity		Set		

3. To specify the domain object's processes, select the **Processes** tab and then click the **Add Process** button at the bottom to invoke the *New Process* dialog box. Enter "Dry Deposition" and then click **OK** on the *New Process* dialog box. The field under the **Processes** tab now displays "Dry Deposition".



Type: Atmosphere

Dry Deposition

4. Close the *[Domain Object] Atmosphere* window by clicking on the X in the upper right corner of the window.

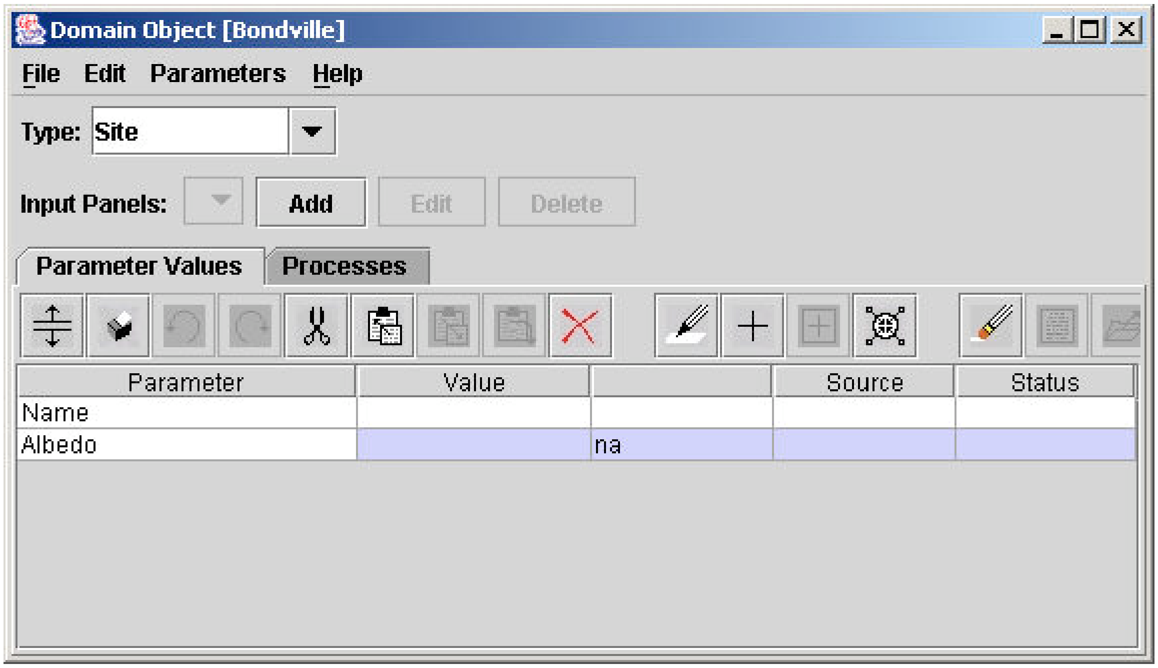
Create the Bondville MIMS Domain Object (QuickStart)

You will create the Bondville domain object of type "Site" and then copy two parameters (Name and Albedo) from the Dep-Module into the Bondville domain object.

1. In the *My QuickStart* window, click **New** to invoke the *Create Member Dialog* box. Select Domain Object from the tree, enter "Bondville" in the *Name* field, then click **Create**. The *New Members Created* dialog box is presented prompting you to open a new member, "Bondville". Click **Yes** to open the *[Domain Object] Bondville* window.

2. Select "Site" from the choices for the *Type* field. Select the **Parameters/Copy from Module** menu item to invoke the *Select Module* dialog box. Choose Dep_Module, then click **OK**. This action copies into the domain object all relevant parameters (Name and Albedo) from the Dep_Module module instance.

   Note: No processes are associated with the Bondville domain object.



Type: Site

Parameter	Value		Source	Status
Name				
Albedo		na		

3. Close the *[Domain Object] Bondville* window by clicking on the X in the upper right corner of the window.

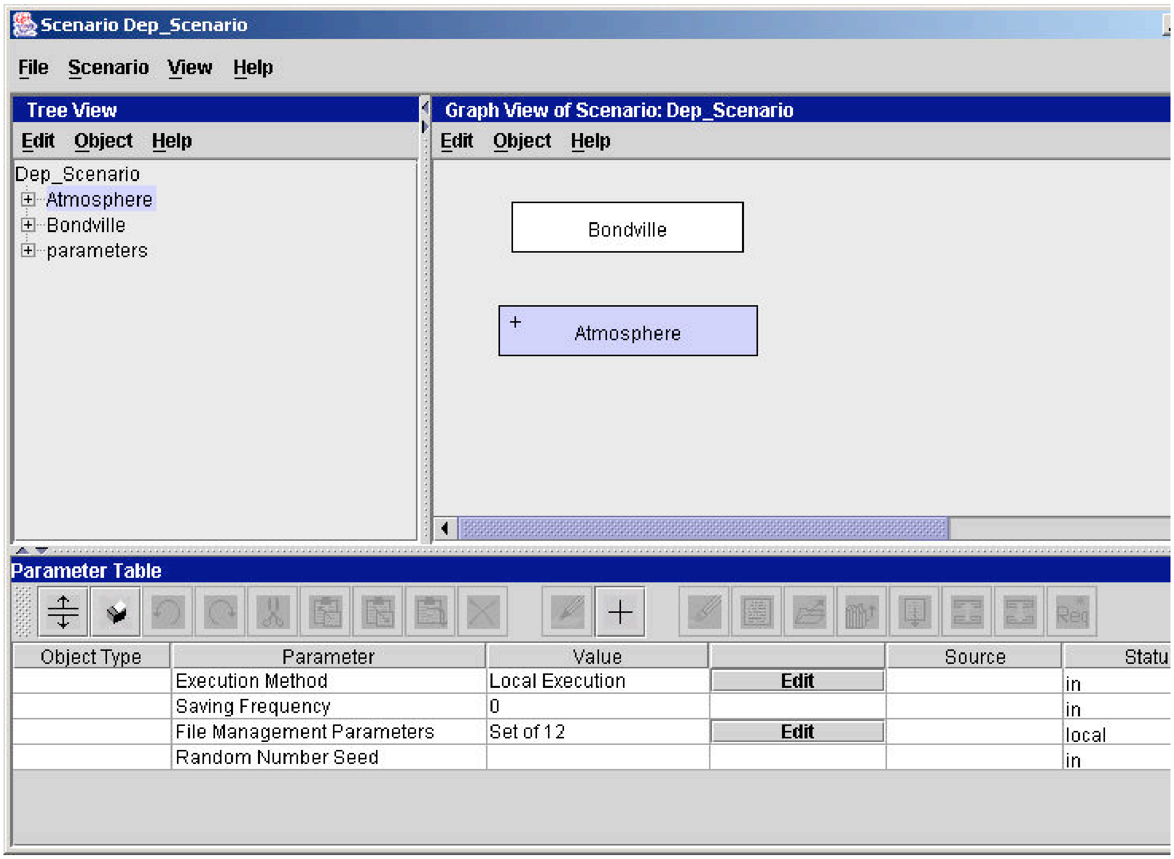
Create the Dep_Scenario MIMS Scenario (QuickStart)

You will create a MIMS Scenario and place the two domain objects (Atmosphere and Bondville) you just created in the scenario. AMIMS scenario defines the operational sequence and settings for a given simulation.

1. In the *My QuickStart* window, click **New** to invoke the *Create New Member Dialog*. Select Scenario, enter "Dep_Scenario" in the *Name* field, then click **Create**. The *New Members Created* dialog box is presented prompting you to open a new member, "Dep_Scenario". Click **Yes** to open the *Scenario Dep_Scenario* window. This window provides three views of the scenario's contents. On the left is an outline listing the scenario's domain objects in a tree view. The right side of the window shows a diagram with a graphical view of the scenario's domain objects and their connections (blank at first). The table at the bottom of the scenario window shows the scenario parameters.

2. Add a domain object to the scenario by right-clicking in the graph view. A menu appears. Choose **Add Domain Object** to invoke the *Select Domain Object* dialog box. Select **Bondville**, then click **OK**. A box with the name Bondville appears where the mouse was clicked. Left-click on a domain object and hold the button down to drag the domain object to the upper left corner of the graph view.

3. Right click on an empty spot in the graph view. When the menu appears, choose **Add Domain Object** again to invoke the *Select Domain Object* dialog box. Select **Atmosphere**. Drag the Atmosphere domain object to the area just below the Bondville domain object.



Object Type	Parameter	Value		Source	Statu
	Execution Method	Local Execution	Edit		in
	Saving Frequency	0			in
	File Management Parameters	Set of 12	Edit		local
	Random Number Seed				in

Connect the Domain Objects in the Scenario (QuickStart)

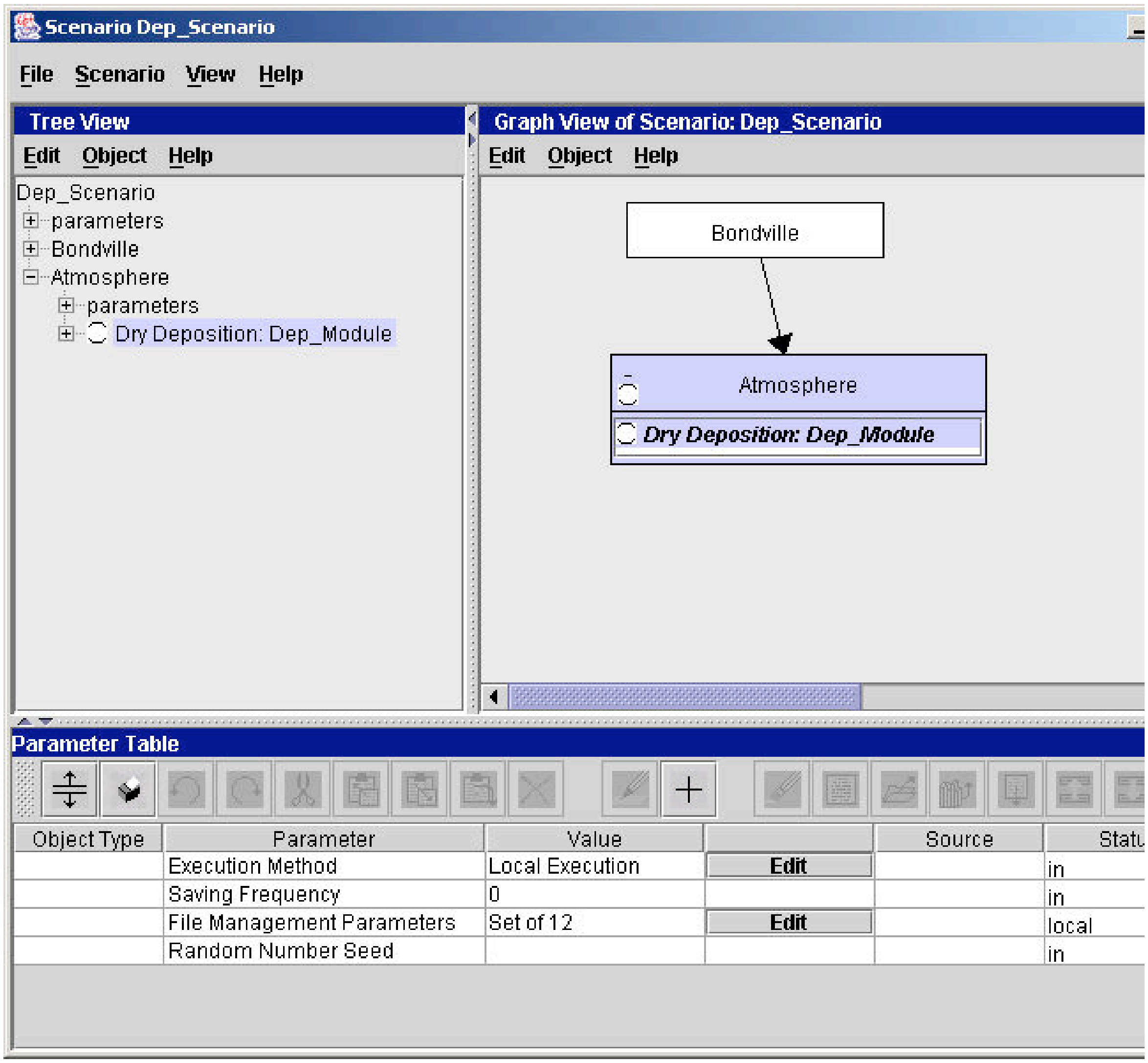
In the Scenario window, follow the steps below to connect the two domain objects and their input parameters.

1. In the graph view of the Dep_Scenario scenario window, right-click on Atmosphere and choose **Add Parameter Source** from the menu. A blue arrow appears pointing to Atmosphere. Then move the mouse to Bondville and left-click. A black arrow appears between the two domain objects with the arrow pointing to Atmosphere. This arrow indicates that the Bondville domain object provides parameters to Atmosphere and its processes.

2. Expand the domain objects by clicking on the **Object/Expand All** menu item in the graph view. The Expand All command shows the processes in each domain object. To attach a module to implement dry deposition, right-click on Dry Deposition (in either the tree or graph views) and choose **Set Module Instance** from the menu. A list of the applicable modules appears (based on domain object type and the process) in the *Select Module* dialog box. Choose Dep_Module, then click **OK**.

3. The *Select Parameter Connections* dialog box appears offering to connect some of the model's input parameters. Click the **Create Connections** button. A module instance is created that represents this invocation of the model.

4. In the scenario window, the Dry Deposition process now has a broken circle on its left and the phrase Dep_Module on its right.



Object Type	Parameter	Value		Source	Statu
	Execution Method	Local Execution	Edit		in
	Saving Frequency	0			in
	File Management Parameters	Set of 12	Edit		local
	Random Number Seed				in

Assign Values to the Parameters (QuickStart)

In the *Scenario Dep_Scenario* window, the partial circle to the left of the Dry Deposition process indicates that some information required to run Dep_Module is missing. Follow the steps below to provide that information.

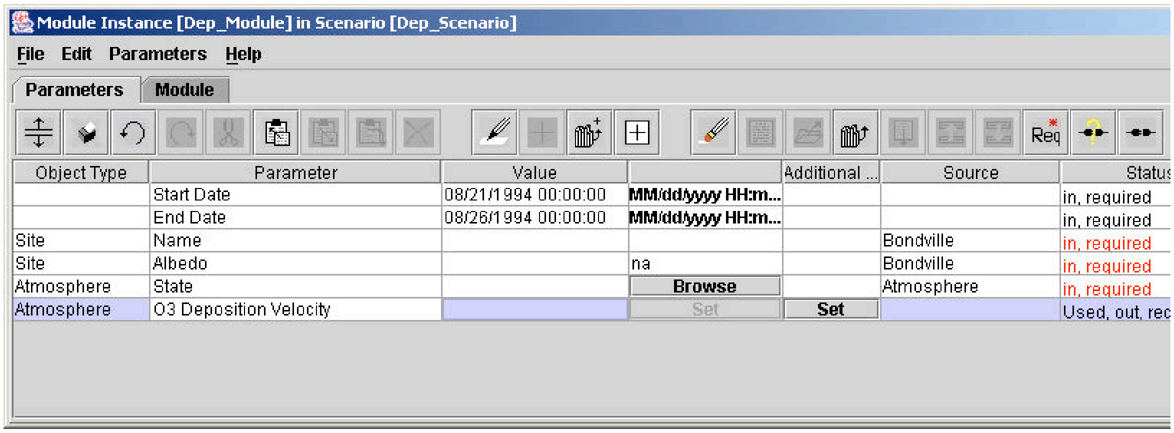
1. Double-click the Dry Deposition process on the Graph View of Scenario: Dep_Scenario in the *Scenario Dep_Scenario* window to invoke the *Module Instance [Dep_Module] in Scenario [Dep_Scenario]* window. See the parameters which initially are copies of the parameters in the module. The red text in the Status column on the right indicate parameters that must be set. Set the following values by double-clicking in the Value column and entering the following values:

   For the Start Date parameter, type "08/21/1994 00:00:00"

   For the End Date parameter, type "08/26/1994 00:00:00"

2. To use the model's result, highlight the O3 Deposition Velocity row and then select the **Parameters/Connect to Parameter** menu item to invoke the *Select Parameter Sources* dialog box. A list shows the scenario and domain objects that could provide the parameter. Select "Atmosphere" and click **OK**. The *Select Connections to Make* dialog box is presented. It shows the potential connection selected by default. Click **Create Connections**. This indicates that the output of the model should be passed to the Atmosphere domain object. Note: The status column for the O3 Deposition Velocity parameter on the module instance window now indicates that the parameter is "Used".

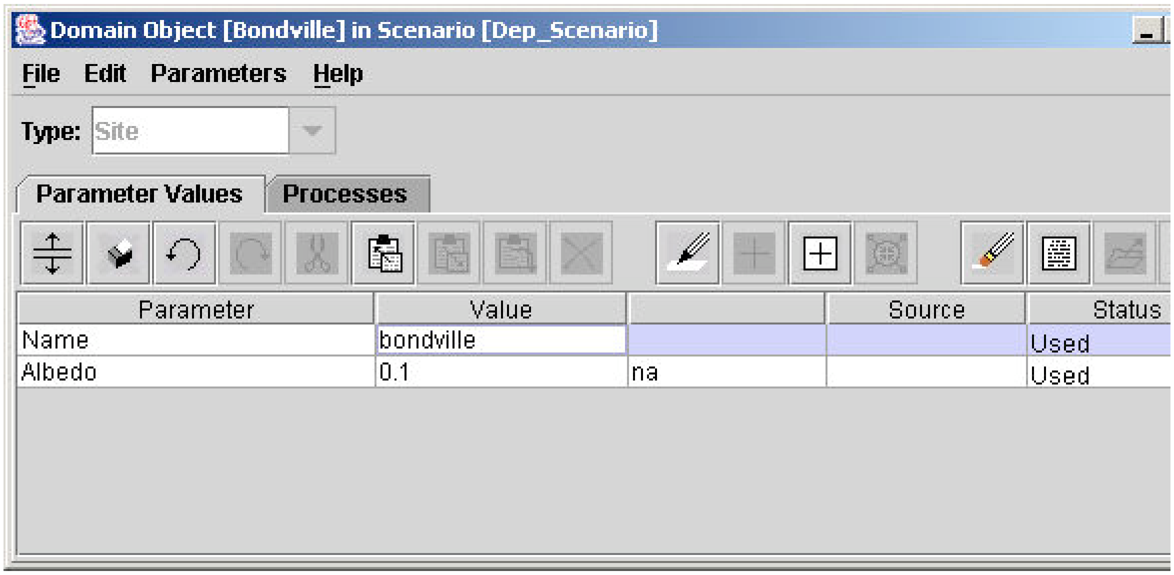
3. The other input parameters cannot be set here because they come from domain objects. The values must be set by those domain objects. Close the *Module Instance [Dep_Module] in Scenario [Dep_Scenario]* window.



Module Instance [Dep_Module] in Scenario [Dep_Scenario]

Object Type	Parameter	Value		Additional ...	Source	Status
	Start Date	08/21/1994 00:00:00	MM/dd/yyyy HH:m...			in, required
	End Date	08/26/1994 00:00:00	MM/dd/yyyy HH:m...			in, required
Site	Name				Bondville	in, required
Site	Albedo		na		Bondville	in, required
Atmosphere	State		Browse		Atmosphere	in, required
Atmosphere	O3 Deposition Velocity		Set	Set		Used, out, rec

4. In the *Scenario Dep_Scenario* window, open the Bondville domain object window by double-clicking on its box to invoke the *Domain Object [Bondville] in Scenario [Dep_Scenario]* window. Enter "bondville" and "0.1" in the Value column for the Name and Albedo parameters. Close the *Domain Object [Bondville] in Scenario [Dep_Scenario]* window.

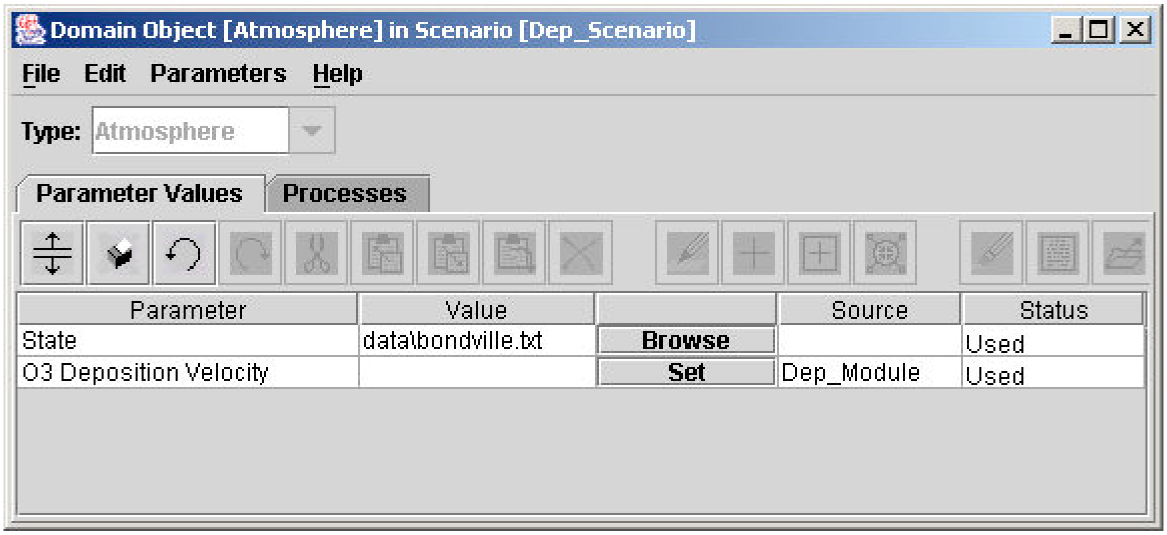


Domain Object [Bondville] in Scenario [Dep_Scenario]

Type: Site

Parameter	Value		Source	Status
Name	bondville			Used
Albedo	0.1	na		Used

5. Open the Atmosphere domain object window by double-clicking on the word Atmosphere to present the *Domain Object [Atmosphere] in Scenario [Dep_Scenario]* window. Enter "data\bondville.txt" for the State value. Close the *Domain Object [Atmosphere] in Scenario [Dep_Scenario]* window.



Domain Object [Atmosphere] in Scenario [Dep_Scenario]

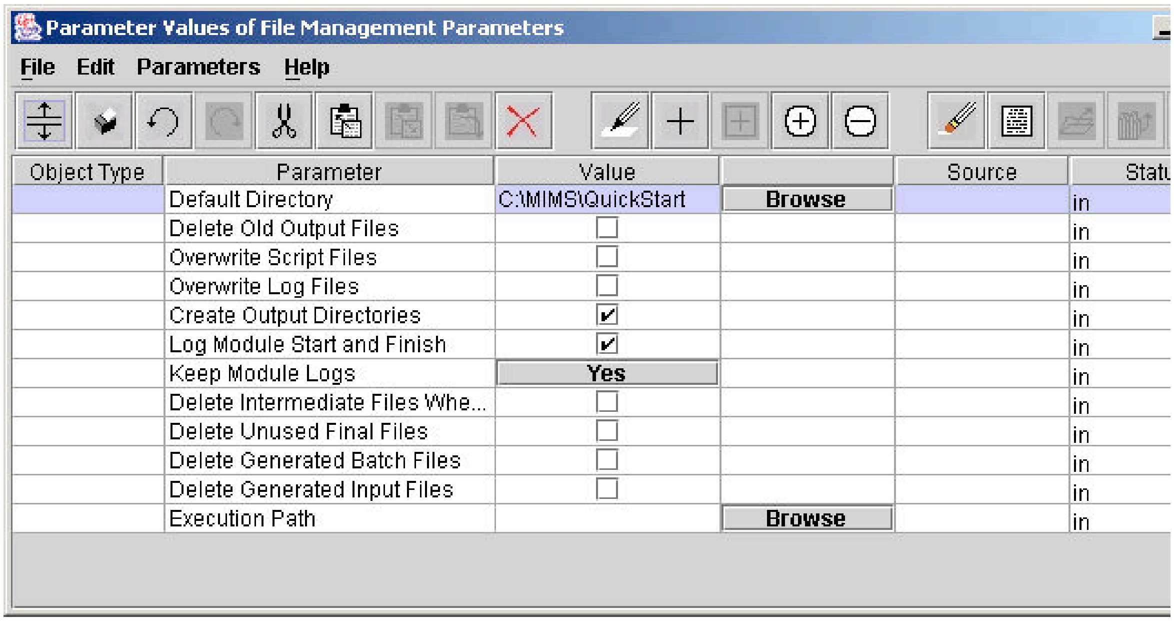
Type: Atmosphere

Parameter	Value		Source	Status
State	data\bondville.txt	Browse		Used
O3 Deposition Velocity		Set	Dep_Module	Used

6. In the *Scenario Dep_Scenario* window there should now be an open green circle to the left of the Dry Deposition process. This green circle indicates that the module is ready to run.

7. Before you can run the scenario, you need to indicate the default directory for this work. Look at the Scenario Parameters table at the bottom of the *Scenario Dep_Scenario* window (if the panel is not showing, click on the small triangle at the bottom left of the window). Beside the File Management Parameters parameter, click the **Edit** button.

8. The *Parameter Values of File Management Parameters* dialog box is presented showing the 12 variables in the group called File Management Parameters. In the Default Directory row of this dialog, click the **Browse** button to invoke the *Select a Directory* dialog box. Find the QuickStart subdirectory within the MIMS directory, double-click on it in the file list, then click the **Select** button. The Default Directory value should now match the complete directory path (e.g., C:\MIMS\QuickStart). Close the *Parameter Values of File Management Parameters* dialog box.



Object Type	Parameter	Value		Source	Statu
	Default Directory	C:\MIMS\QuickStart	Browse		in
	Delete Old Output Files	☐			in
	Overwrite Script Files	☐			in
	Overwrite Log Files	☐			in
	Create Output Directories	☑			in
	Log Module Start and Finish	☑			in
	Keep Module Logs	Yes			in
	Delete Intermediate Files Whe...	☐			in
	Delete Unused Final Files	☐			in
	Delete Generated Batch Files	☐			in
	Delete Generated Input Files	☐			in
	Execution Path		Browse		in

Execute the Scenario (QuickStart)

Now you are ready to execute the model.

1. On the *Scenario Dep_Scenario* window, select the **Scenario/Execute Ready Modules** menu item. The Dep_Module will be executed; as it operates, a filled green circle appears beside the Dry Deposition process. A check mark in the green circle indicates that the execution completed. Some models indicate failure by returning a status code that MIMS recognizes. The demonstration model does not return such a status code, so the Dep_Module always appears to complete successfully.

2. To see if the model did execute successfully, right-click on the Dry Deposition process and choose **View Output Log**. Microsoft Notepad starts and shows the log file created by the model (if not, set the viewer to Notepad in the Edit Administration Information Window). A log file for a successful execution should be similar to the following text:

   ```
   Reading config.txt
   Reading bondvill.lai
   Reading plants\PLANT.DAT
   Reading plants\PADPROF1.20
   Reading plants\PADPROF2.20
   Reading plants\PADPROF3.20
   Reading plants\LANG.SPH
   Running Dry Deposition Model for bondvill
   Output will be to data\output.txt
   All data will be output, with bad values blanked.
   Week 1: 08/21/94 08/22/94 08/23/94 08/24/94 08/25/94 08/26/94
   Done: 288 records in 8.0 seconds: 36.0 records/second
   ```

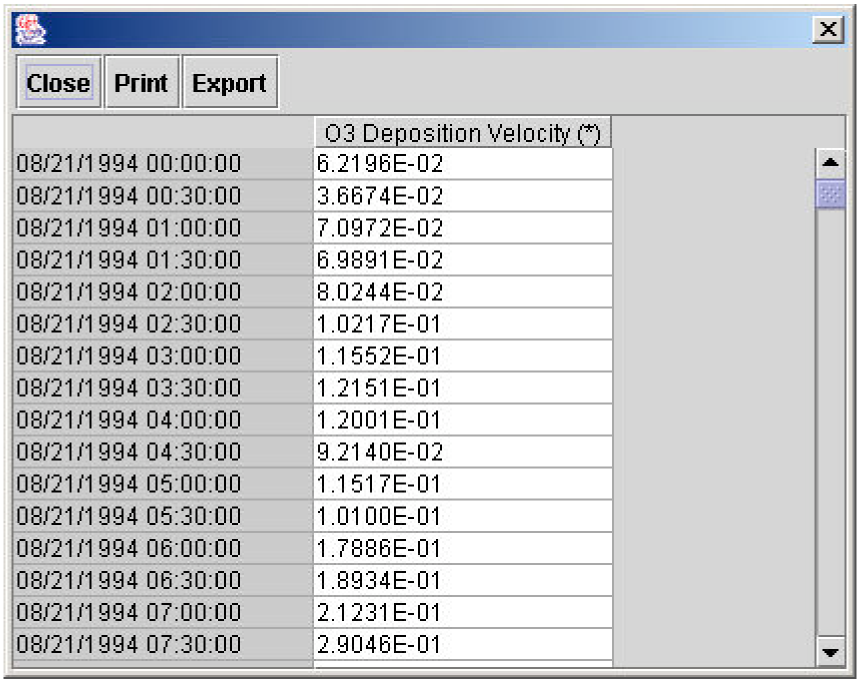
   If the scenario has been run by a previous user, the following line also appears in the output log:

   ```
   WARNING: Output File already exists and will be overwritten.
   ```

3. Close the Notepad window after examining the file.

4. Double click the mouse on the Dry Deposition process in the *Scenario Dep_Scenario* window. The *Module Instance [Dep_Module]* window should appear. Right-click the mouse on the row containing the parameter O3 Deposition Velocity and select **View Series of Values**. A list of the output

variables with the O3 Deposition Velocity will be displayed.



	O3 Deposition Velocity (*)
08/21/1994 00:00:00	6.2196E-02
08/21/1994 00:30:00	3.6674E-02
08/21/1994 01:00:00	7.0972E-02
08/21/1994 01:30:00	6.9891E-02
08/21/1994 02:00:00	8.0244E-02
08/21/1994 02:30:00	1.0217E-01
08/21/1994 03:00:00	1.1552E-01
08/21/1994 03:30:00	1.2151E-01
08/21/1994 04:00:00	1.2001E-01
08/21/1994 04:30:00	9.2140E-02
08/21/1994 05:00:00	1.1517E-01
08/21/1994 05:30:00	1.0100E-01
08/21/1994 06:00:00	1.7886E-01
08/21/1994 06:30:00	1.8934E-01
08/21/1994 07:00:00	2.1231E-01
08/21/1994 07:30:00	2.9046E-01

You may print or export the values for further study. Press the **Close** button to close the list interface, and then close the *Module Instance [Dep_Module]* window.

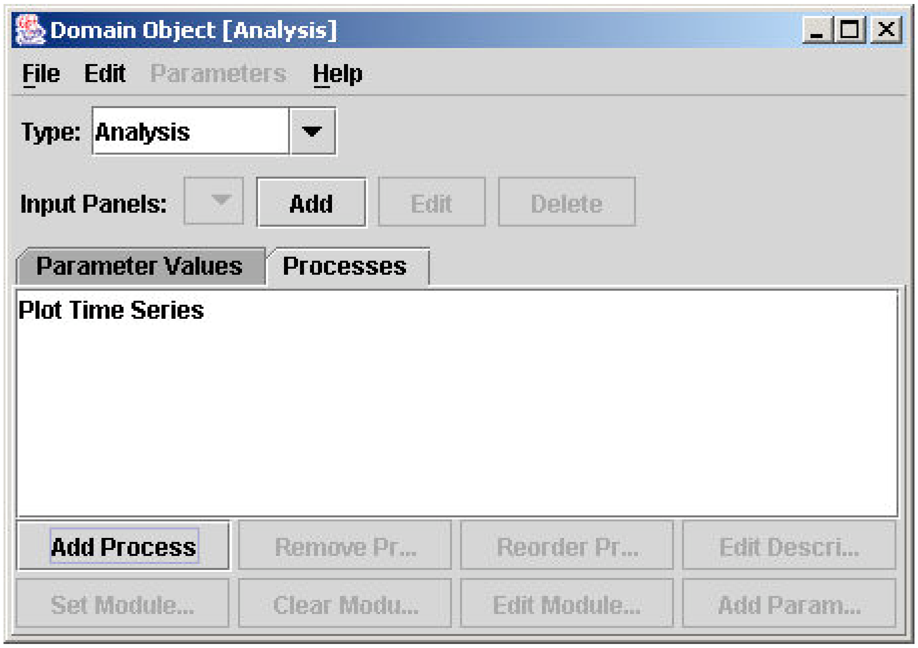
Add the Analysis MIMS Domain Object (QuickStart)

Follow the steps below to add the Analysis MIMS Domain Object to QuickStart project.

1. In the *My QuickStart* project window, click **New**. The *Create Member Dialog* box is presented. You will use a time series plot to view the results. Select Time Series Plotter from the Modules folder. Type "Time Series Plotter" in the *Name* field as the name for the module and then click the **Create** button. The *New Member Created* dialog box appears. Click **No** to avoid opening the *Time Series Plotter [Time Series Plotter]* window at this time.

2. You will use a domain object to hold the plot routine. (In a future version, a framework object may be available for this purpose so that the domain objects will not be used. The tasks of framework objects are typically independent of the environmental models and instead represent computational functions.) In the *My QuickStart* project window, click **New** to invoke the *Create Member Dialog* box. Select Domain Object. Type "Analysis" in the *Name* field as the name for the domain object, then click **Create**. The *New Member Created* dialog box appears. Click **Yes** to open the *Domain Object [Analysis]* window.

3. In the *Type* field, select "Analysis" from the pulldown menu and click the **Add Process** button at the bottom of the Processes tab to invoke the *New Process* dialog box where you may add a Process called "Plot Time Series." Close the *Domain Object [Analysis]* window.



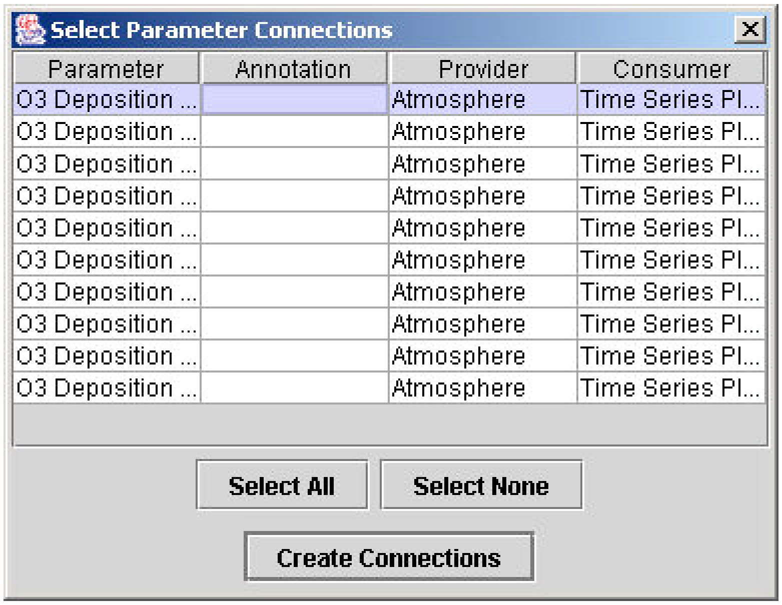
Type: Analysis

Plot Time Series

4. In the *Scenario Dep_Scenario* window, right-click on the background in the graph view. Select **Add Domain Object** from the menu. The *Select Domain Object* dialog box is presented. Choose Analysis. This process brings the Analysis domain object into the scenario.

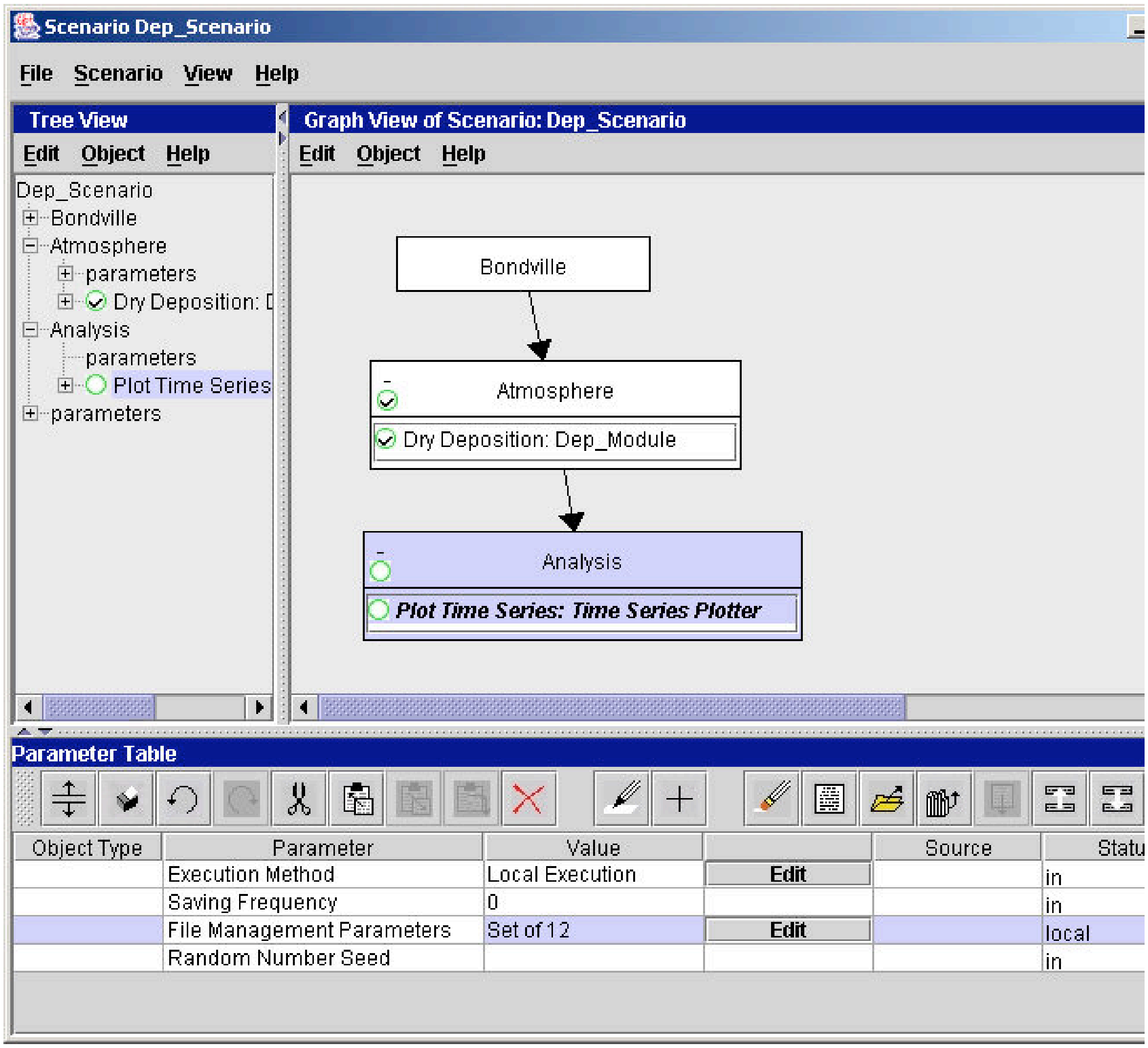
5. Right click on the Analysis domain object in the graph view, choose **Add Parameter Source** from the menu. A blue arrow appears. Left-click on the Atmosphere domain object. Now a black arrow points from the Atmosphere domain object to the Analysis domain object. Click on the "+" in Analysis to see its process (Plot Time Series). Right-click on Plot Time Series, choose the **Set Module Instance** command on the right-click menu to invoke the *Select Module* dialog box. Choose Time Series Plotter, then click **OK**.

6. The *Select Parameters Connections* dialog box appears with a list of possible parameter connections. The MIMS Time Series Plotter can plot 10 variables simultaneously, so multiple possible connections are shown. Click on the top row to highlight only that row, then click **Create Connections**. The time series plotter automatically uses the output specified by the Atmosphere domain object.



Parameter	Annotation	Provider	Consumer
O3 Deposition ...		Atmosphere	Time Series Pl...
O3 Deposition ...		Atmosphere	Time Series Pl...
O3 Deposition ...		Atmosphere	Time Series Pl...
O3 Deposition ...		Atmosphere	Time Series Pl...
O3 Deposition ...		Atmosphere	Time Series Pl...
O3 Deposition ...		Atmosphere	Time Series Pl...
O3 Deposition ...		Atmosphere	Time Series Pl...
O3 Deposition ...		Atmosphere	Time Series Pl...
O3 Deposition ...		Atmosphere	Time Series Pl...
O3 Deposition ...		Atmosphere	Time Series Pl...

7. In the *Scenario Dep_Scenario* window, right-click on the Plot Time Series process and select **Update Module Instance** from the right-click menu. Note: If you do not do this, the parameters in the output file will not be properly read by the Time Series Plotter.



Object Type	Parameter	Value		Source	Statu
	Execution Method	Local Execution	Edit		in
	Saving Frequency	0			in
	File Management Parameters	Set of 12	Edit		local
	Random Number Seed				in

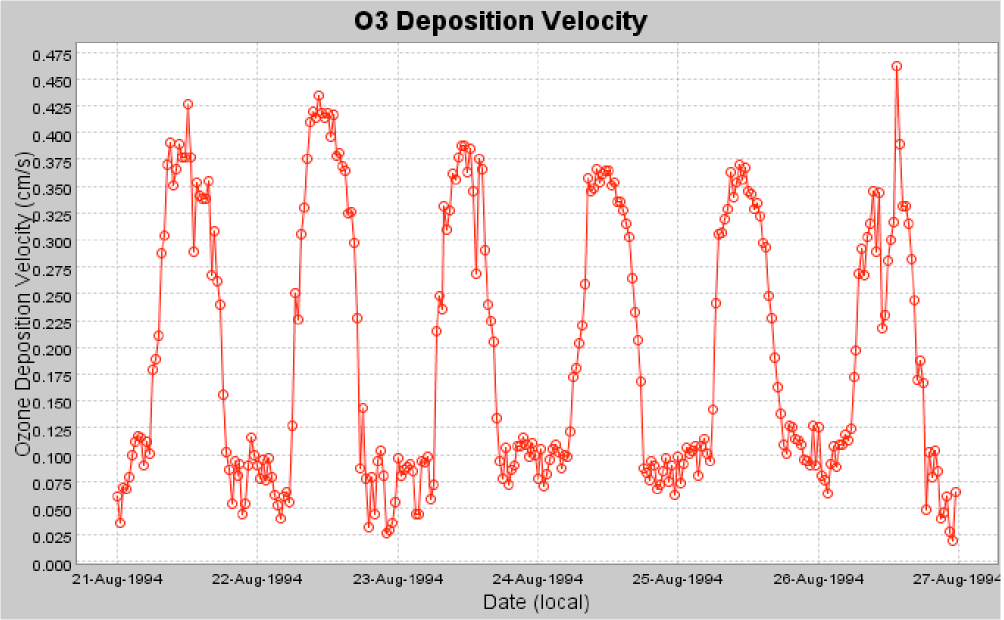
8. In the *Scenario Dep_Scenario* window, choose the **Scenario/Reset All** then **Scenario/Execute Ready Modules** menu items. When the execution completes, you should see a time series plot of the deposition velocities.

Interpret the Results Using the Time Series Plotter (QuickStart)

The window that plots your modeled results is titled *Analysis Time Series*. The x-axis shows the dates from August 21-27, 1994 (the dates are centered on midnight at the start of the day), and the y-axis shows deposition velocities in cm/s. The plot shows the model output. Note that the ozone deposition velocities reach their maxima during the day and drop at night.

1. You can customize the appearance of the axes, labels, and background by right-clicking the mouse on the window to invoke a menu and then click on **Properties**. On the *Chart Properties* dialog box, select the **Plot** tab and the **Vertical Axis** tab within that. Change the label from "cm/s" to "Ozone Deposition Velocity (cm/s)" and click **OK**.

2. Right-click again on the plot to invoke a menu and then click **Print**. Choose the appropriate printer in landscape mode, then click **OK**. Your printed chart should look like this:



Congratulations on completing the basic QuickStart tutorial!

Now close the plot window.

Expand to Employ a Monte Carlo Approach (QuickStart)

In this expansion, you will add a new parameter to the Dep_Scenario scenario and then copy the scenario to a simulation called Dep_Monte. The Analysis object will be removed from the Dep_Monte scenario, in order to allow the Dep_Monte scenario to act as a single module within another scenario (Scen_Monte Carlo).

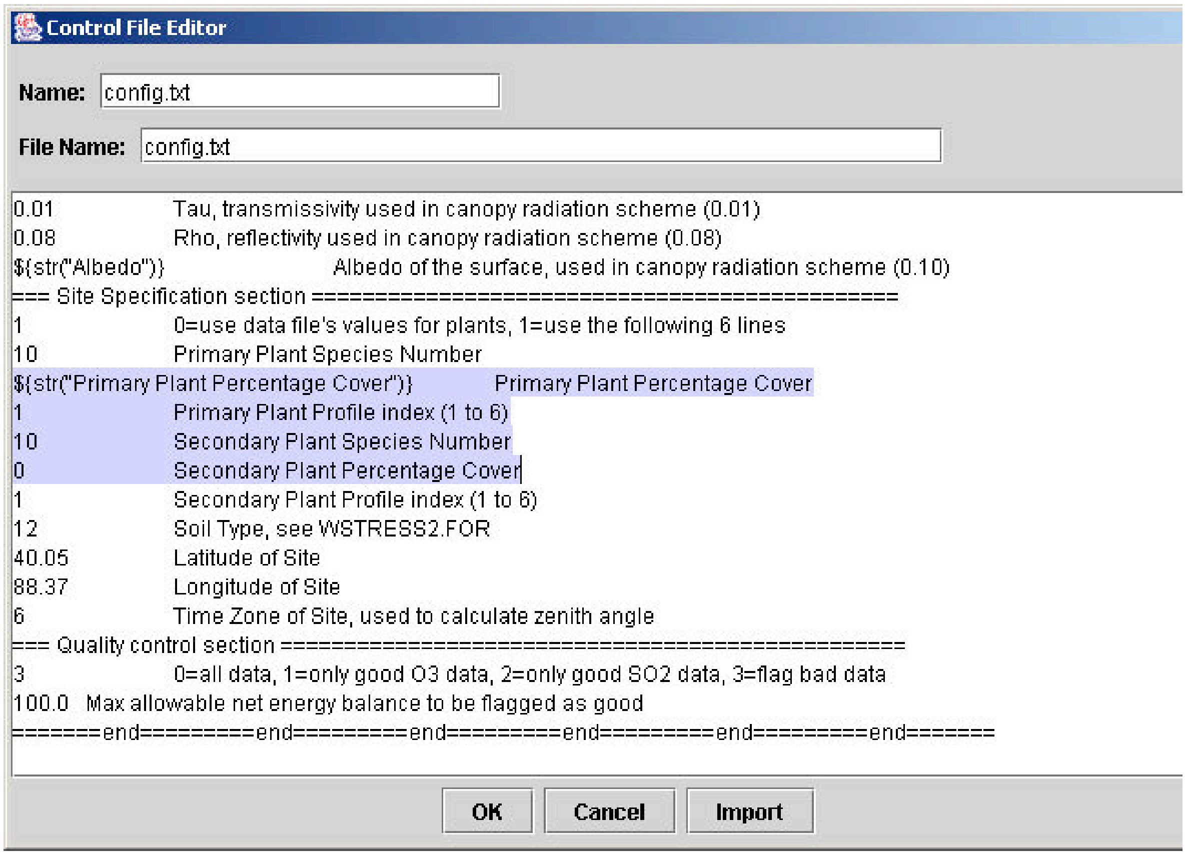
The Scen_Monte Carlo scenario will use a Monte Carlo Framework Object to feed the new parameter to the Dep_Monte module and will read back the maximum ozone deposition velocity. The output from the framework object will be displayed by a histogram plotter.

Step 1. Create the Dep_Monte Scenario

a. From the *My QuickStart* project window, select Dep_Module from the Default External Module list, then click **Open** to open the *Module [Dep_Module]* window.

Click the **Add New Parameter** icon (+) to add a "Primary Plant Percentage Cover" floating point parameter. It should be assigned a domain object type of Atmosphere, as input, and as required. This parameter represents the fraction of the land covered by the primary plant type and is often subject to some uncertainty. After the parameter is created, click the **Edit Parameter Type** icon (✎) to set the units to %.

b. On the *Module Dep_Module* window, click **Edit** to edit the config.txt control file. Near the end of the file, replace the explicit number "50" beside "Primary Plant Percentage Cover" with ${str("Primary Plant Percentage Cover")}. Change the Secondary Plant Percentage Cover from "50" to "0". Click **OK** to accept the change. Then close the *Module Dep_Module* window.



Name: config.txt

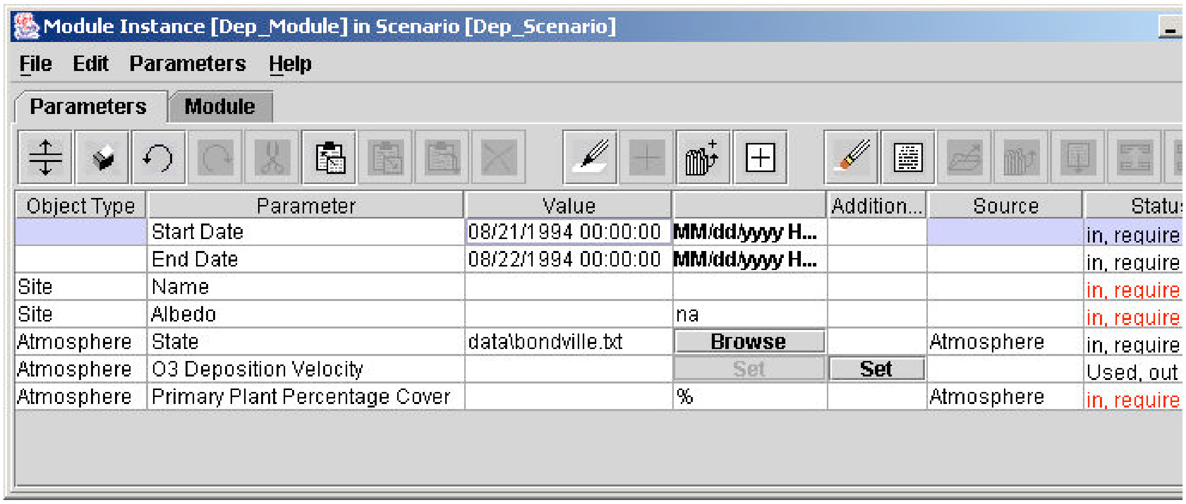
File Name: config.txt

```
0.01            Tau, transmissivity used in canopy radiation scheme (0.01)
0.08            Rho, reflectivity used in canopy radiation scheme (0.08)
${str("Albedo")}                Albedo of the surface, used in canopy radiation scheme (0.10)
=== Site Specification section ============================================
1               0=use data file's values for plants, 1=use the following 6 lines
10              Primary Plant Species Number
${str("Primary Plant Percentage Cover")}        Primary Plant Percentage Cover
1               Primary Plant Profile index (1 to 6)
10              Secondary Plant Species Number
0               Secondary Plant Percentage Cover
1               Secondary Plant Profile index (1 to 6)
12              Soil Type, see WSTRESS2.FOR
40.05           Latitude of Site
88.37           Longitude of Site
6               Time Zone of Site, used to calculate zenith angle
=== Quality control section ===============================================
3               0=all data, 1=only good O3 data, 2=only good SO2 data, 3=flag bad data
100.0  Max allowable net energy balance to be flagged as good
=======end=========end=========end=========end=========end=========end=======
```

c. From the *My QuickStart* project window, select Atmosphere from the Domain Object list, then click **Open** to open the Atmosphere domain object. In the *[Domain Object] Atmosphere* window, click on the **Copy from Module** () icon. Choose the Dep_Module when prompted for the module name. This adds Primary Plant Percentage Cover to Atmosphere's parameter list. Close the *[Domain Object] Atmosphere* window.

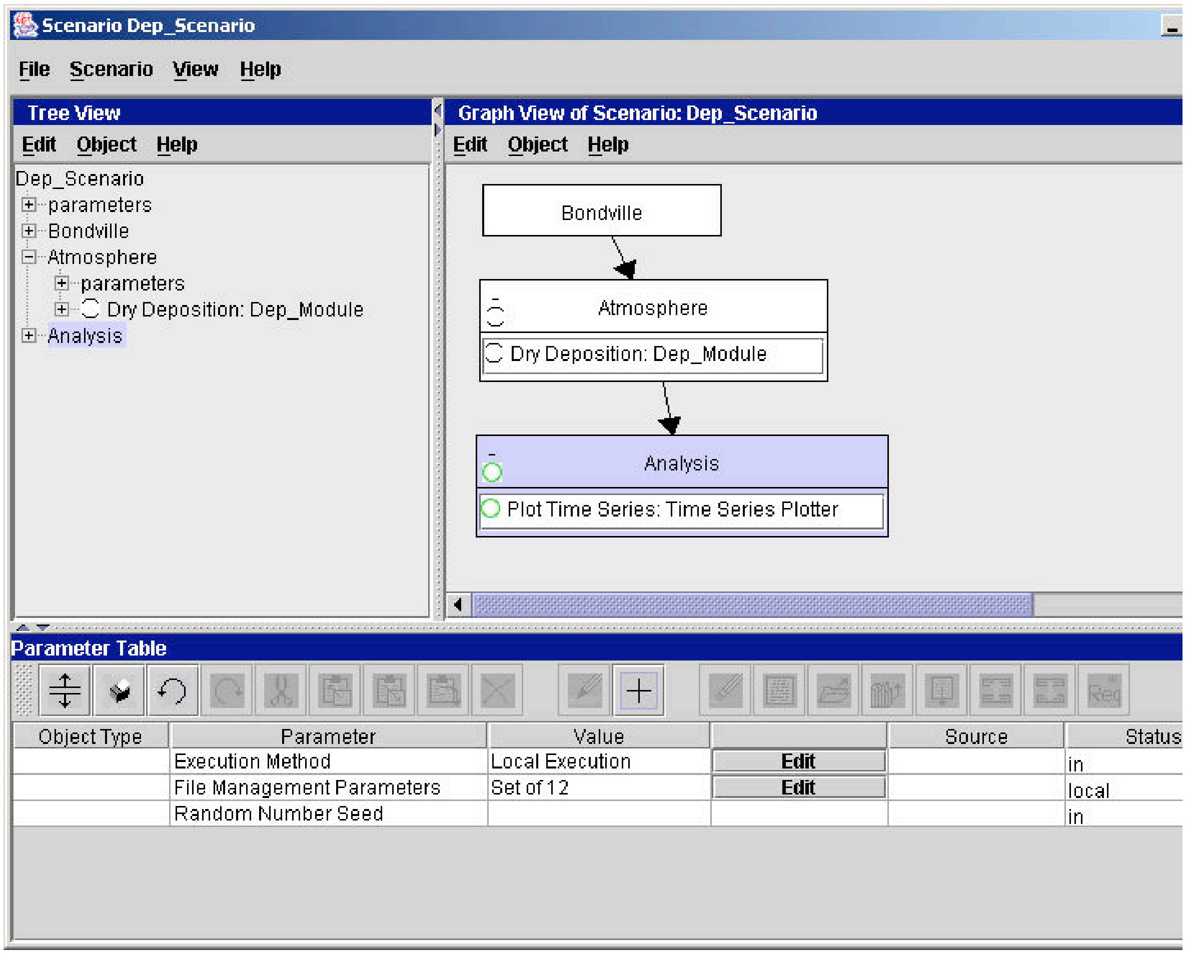
d. In the *Scenario Dep_Scenario* window, you will have to delete the existing Atmosphere domain object. Right-click on Atmosphere to invoke a menu, then choose the **Delete Object** menu item. To add the new Atmosphere domain object, right-click in an open area of the graph view to invoke a menu, then choose the **Add Domain Object** menu item. To set the module instance to Dep_Module, left-click on the "+" sign in the Atmosphere box, followed by a right-click on Dry Deposition to invoke a menu. Then choose the **Set Module Instance** menu item. Set all three parameter connections when prompted.

e. To open the Atmosphere domain object window, double-click on Atmosphere in the graph view, then set the value for the State to "data\bondville.txt". Double click on the Dry Deposition process in the domain object window to open the module instance window. Set the values for the start and end dates to "08/21/1994 00:00:00" and "08/22/1994 00:00:00" in the module instance window. Close the module instance window.



Object Type	Parameter	Value		Addition...	Source	Statu
	Start Date	08/21/1994 00:00:00	MM/dd/yyyy H...			in, require
	End Date	08/22/1994 00:00:00	MM/dd/yyyy H...			in, require
Site	Name					in, require
Site	Albedo		na			in, require
Atmosphere	State	data\bondville.txt	Browse		Atmosphere	in, require
Atmosphere	O3 Deposition Velocity		Set	Set		Used, out
Atmosphere	Primary Plant Percentage Cover		%		Atmosphere	in, require

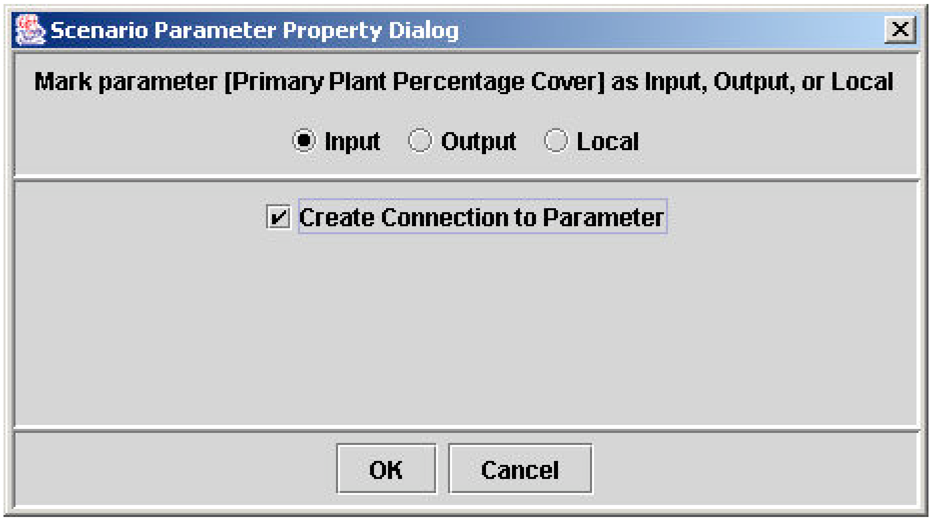
f. In the *Scenario Dep_Scenario* window, to reattach the parameter connection to the Bondville domain object, right-click on the Atmosphere domain object in the graph view, choose the **Add Parameter Source** menu item, drag the arrow tail to the Bondville domain object, select both Albedo and Name, then click **Create Connections**. Next, to reattach the parameter connection to the Analysis domain object right-click on the Analysis domain object in the graph view, choose the **Add Parameter Source** menu item, drag the arrow tail to the Atmosphere domain object, select only the first variable, and click **Create Connections**.



Dep_Scenario
- parameters
- Bondville
- Atmosphere
  - parameters
  - Dry Deposition: Dep_Module
- Analysis

Object Type	Parameter	Value		Source	Status
	Execution Method	Local Execution	Edit		in
	File Management Parameters	Set of 12	Edit		local
	Random Number Seed				in

g. From within the Atmosphere domain object window (still opened from within the *Scenario Dep_Scenario* window), highlight the Primary Plant Percentage Cover parameter, then click the **Add to Scenario** (⊞) icon. Choose the **Input** radio button and check the *Create Connection to Parameter* check box in the *Scenario Parameter Property* dialog box.

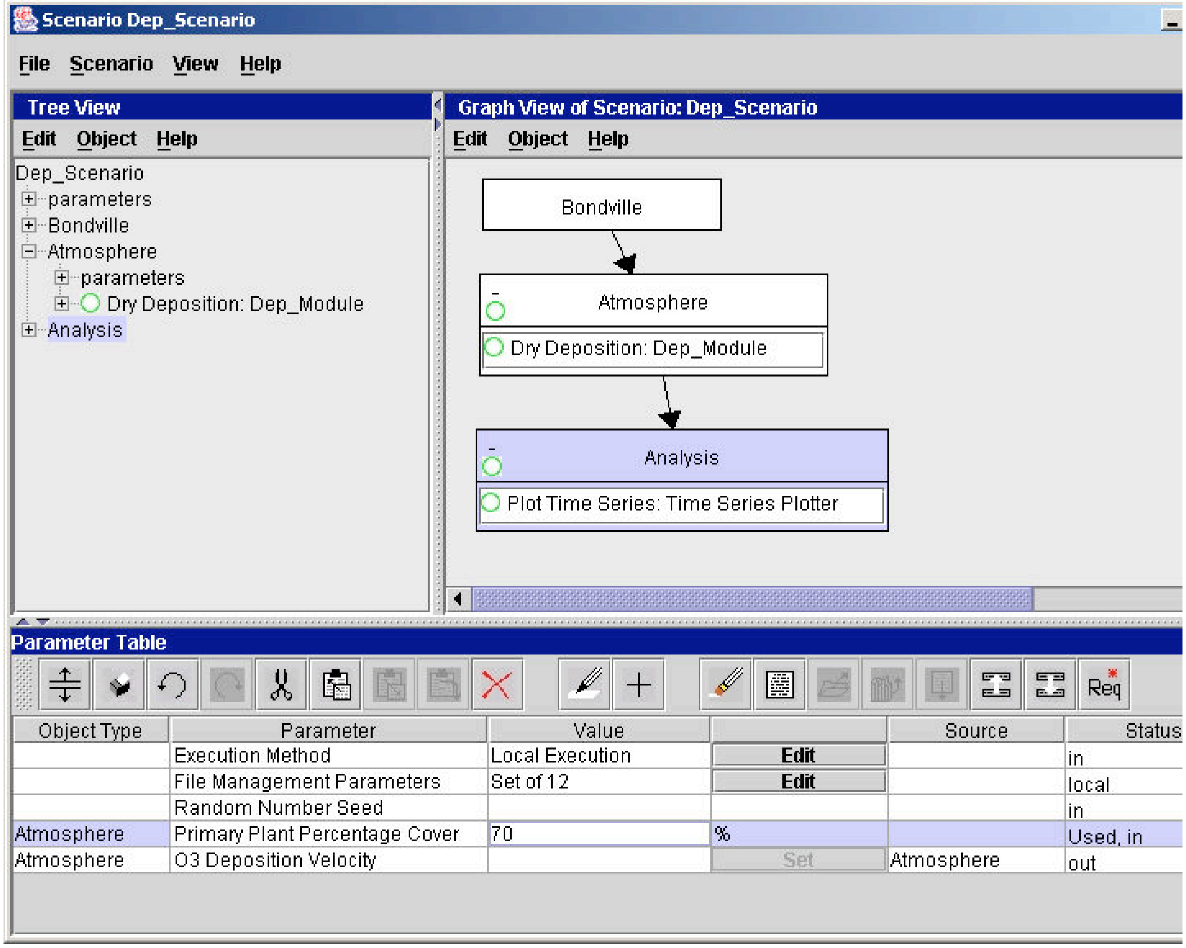


Mark parameter [Primary Plant Percentage Cover] as Input, Output, or Local

● Input ○ Output ○ Local

☑ Create Connection to Parameter

h. Do the same for the O3 Deposition Velocity parameter--choose the **Output** radio button and check the Create Connection to Parameter check box in the *Scenario Parameter Property* dialog box. The scenario parameters table in the scenario window should now show the Primary Plant Percentage Cover with a "Used, in", in status and the O3 Deposition Velocity parameter with a source of "Atmosphere" and status as "Out". Type the value "70" into the value column for Primary Plant Percentage Cover in the Scenario Parameters table. Execute the scenario to be sure that the model still runs properly and produces an appropriate time series (two daily ozone peaks).

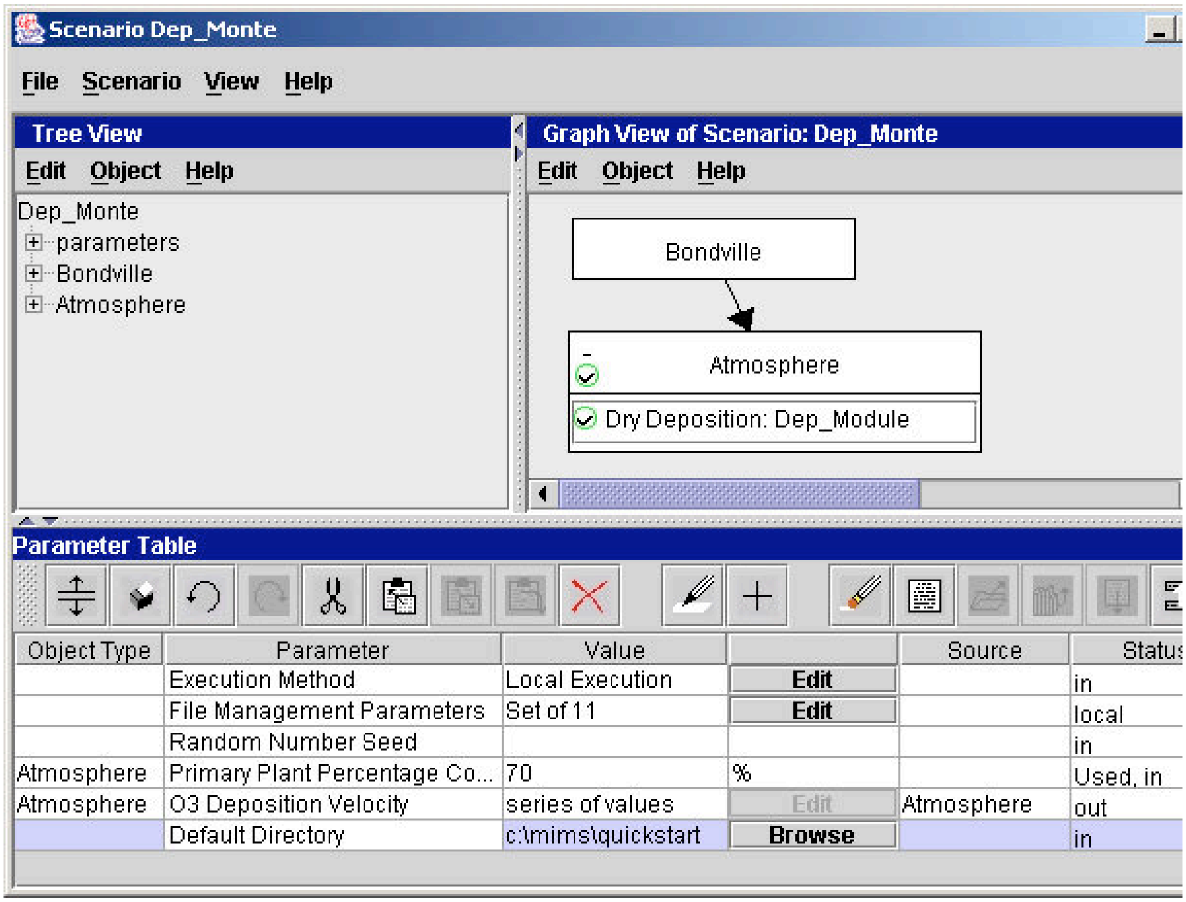


Object Type	Parameter	Value		Source	Status
	Execution Method	Local Execution	Edit		in
	File Management Parameters	Set of 12	Edit		local
	Random Number Seed				in
Atmosphere	Primary Plant Percentage Cover	70	%		Used, in
Atmosphere	O3 Deposition Velocity		Set	Atmosphere	out

i. Close the Dep_Scenario and Atmosphere windows.

j. From the *My QuickStart* project window, highlight the Dep_Scenario and click on the **Duplicate** button. Name the new scenario "Dep_Monte". Then, open the *Scenario Dep_Monte* window. In the *Scenario Dep_Monte* window, right-click on the Analysis domain object and delete it. The Dep_Monte scenario will act as the module that operates repeatedly for the Monte Carlo simulation.

k. The Scenario Parameter Table should appear at the bottom of the scenario window (if not, select its check box under the **View** menu). Click on the **Edit** button beside the File Management Parameters parameter in the Scenario Parameters table. This opens the *Parameter Values of File Management Parameters* window. In this new window, select the Default Directory parameter and then click on the **Remove from Group** (⊖) icon, or select **Remove from Group** in the **Parameters** menu. This action transfers the Default Directory parameter up to the Scenario Parameters table.



Object Type	Parameter	Value		Source	Status
	Execution Method	Local Execution	Edit		in
	File Management Parameters	Set of 11	Edit		local
	Random Number Seed				in
Atmosphere	Primary Plant Percentage Co...	70	%		Used, in
Atmosphere	O3 Deposition Velocity	series of values	Edit	Atmosphere	out
	Default Directory	c:\mims\quickstart	Browse		in

l. Close the window called *Parameter Values of File Management Parameters*.

m. Close the *Scenario Dep_Monte* window.

Expand to Employ a Monte Carlo Approach (QuickStart)

The Simple Monte Carlo Iterator (max) will send a single Monte Carlo parameter into a module (at the scenario parameters level) and output the maximum value for the output parameter for each Monte Carlo simulation.

Step 2. Create the Monte Carlo Iterator

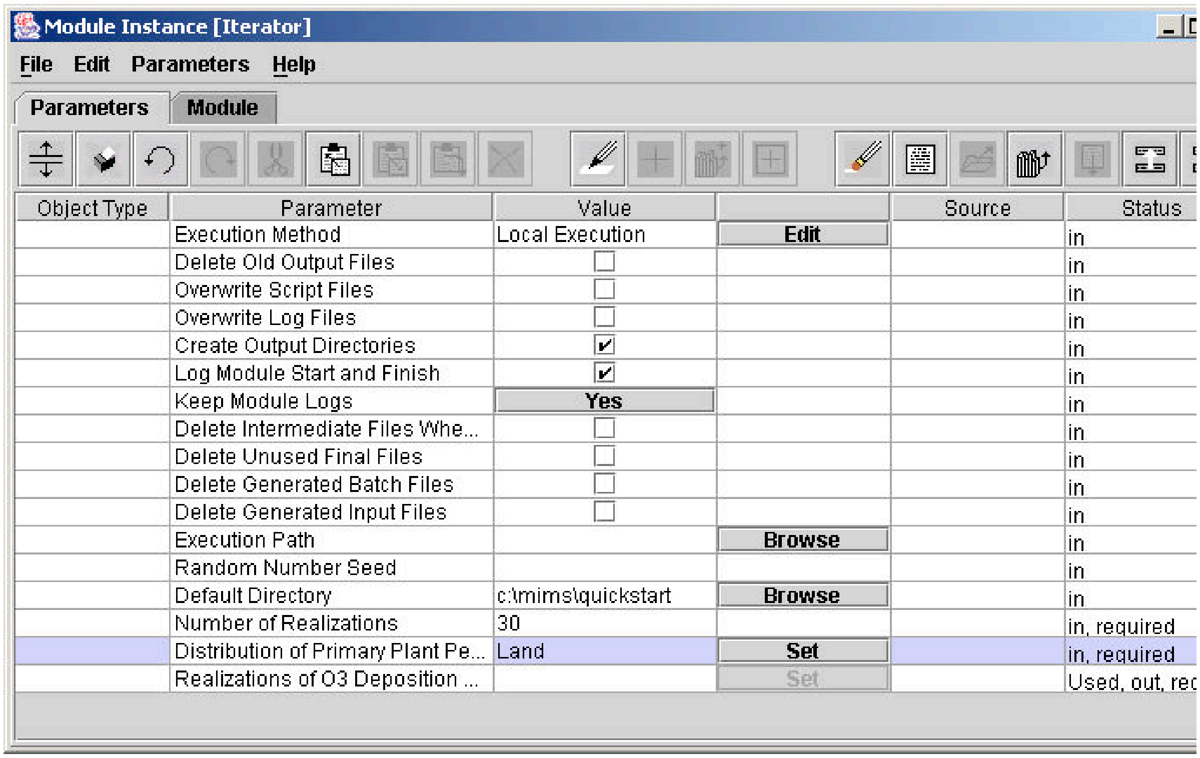
a. To create a Simple Monte Carlo Iterator (Max) in the *My QuickStart* project window, click the **New** button and choose the Simple Monte Carlo Iterator (Max) member found in the Iterators folder and name it Monte Carlo. When prompted to open the new member, click **Yes**. The *Framework Object* window appears for the iterator. Click on the **Processes** tab. Highlight the Iterate process and double-click on it to make the module instance window appear.

b. When prompted "Do you want to add an iterand at this time?", click **Yes**. Choose Dep_Monte to act as the iterating module, then click **OK**. A Dep_Monte (Inputs/Outputs) dialog box appears.

c. On the **Dep_Monte Inputs** tab, select Primary Plant Percentage Cover in the Unselected Items list and click **>>>Add>>>**. The parameter is now listed in the Selected Items List. On the **Dep_Monte Outputs** tab, select O3 Deposition Velocity in the Unselected Items list and click **>>>Add>>>**. The parameter is now listed in the Selected Items List. Click **OK**. The dialog box closes. In addition to the file management parameters, six new parameters are added to the parameters list in the Module Instance window:

- Execution Method (value of Local Execution)
- Random Number Seed
- Default Directory
- Number of Realizations
- Distribution of Primary Plant Percentage Cover
- Realizations of O3 Deposition Velocity.

d. Set the value for the number of realizations to 30. Click the **Set** button beside the Distribution of Primary Plant Percentage Cover parameter. Choose a normal distribution (named Land) with mean of 50 and standard deviation of 10. After choosing the standard deviation, click **OK**. The Status column should now show that all of the parameters have been specified for the module instance. Only the checkboxes beside Create Output Directories and Log Module Start and Finish should be checked. Minimize the *Module Instance [Iterator]* window.

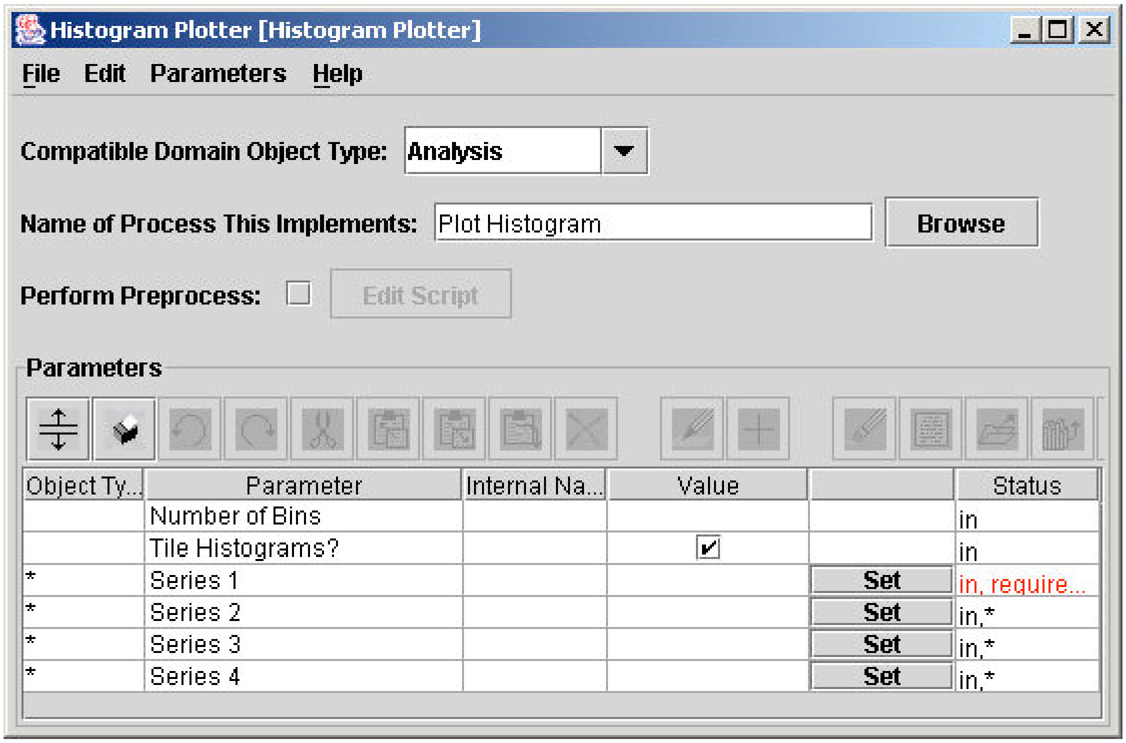


Object Type	Parameter	Value		Source	Status
	Execution Method	Local Execution	Edit		in
	Delete Old Output Files	□			in
	Overwrite Script Files	□			in
	Overwrite Log Files	□			in
	Create Output Directories	☑			in
	Log Module Start and Finish	☑			in
	Keep Module Logs	Yes			in
	Delete Intermediate Files Whe...	□			in
	Delete Unused Final Files	□			in
	Delete Generated Batch Files	□			in
	Delete Generated Input Files	□			in
	Execution Path		Browse		in
	Random Number Seed				in
	Default Directory	c:\mims\quickstart	Browse		in
	Number of Realizations	30			in, required
	Distribution of Primary Plant Pe...	Land	Set		in, required
	Realizations of O3 Deposition ...		Set		Used, out, rec

Expand to Employ a Monte Carlo Approach (QuickStart)

Step 3. Create the Histogram Plotter

a. From the *My QuickStart* window, click **New** and select Histogram Plotter from the Modules folder and name it Histogram Plotter. Click **Yes** when prompted to open the new member, and the compatible domain object type will have automatically been set to "Analysis".



Compatible Domain Object Type: Analysis

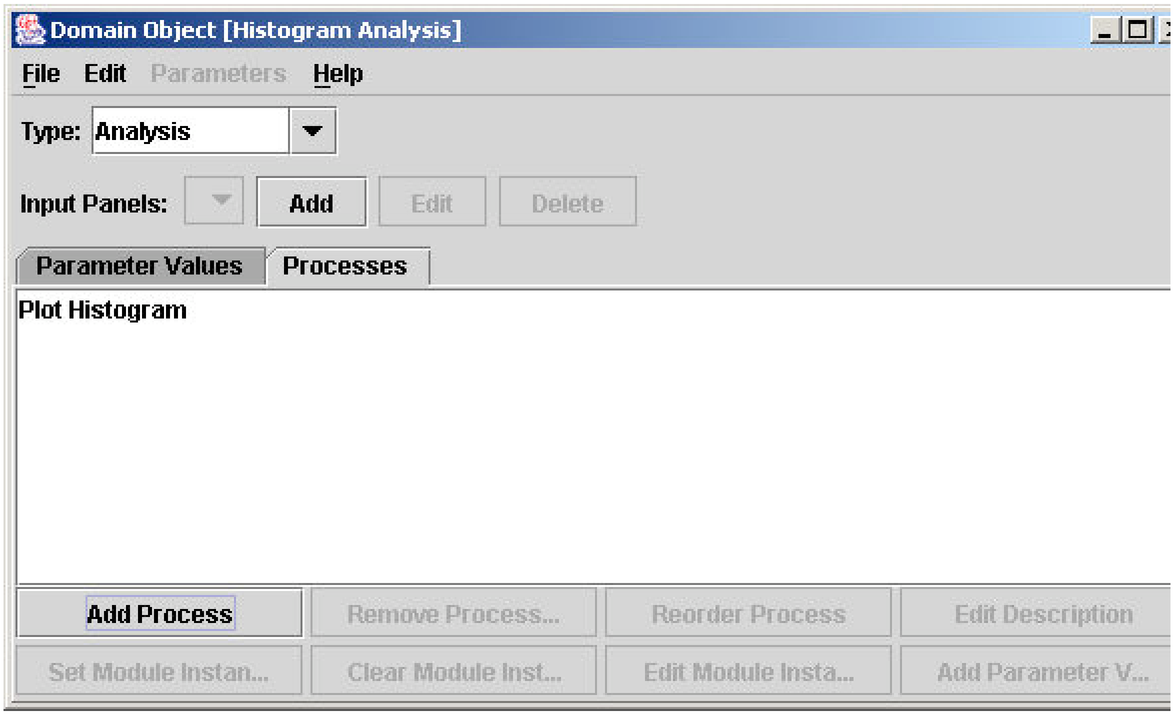
Name of Process This Implements: Plot Histogram

Perform Preprocess: □

Parameters

Object Ty...	Parameter	Internal Na...	Value		Status
	Number of Bins				in
	Tile Histograms?		☑		in
*	Series 1			Set	in, require...
*	Series 2			Set	in,*
*	Series 3			Set	in,*
*	Series 4			Set	in,*

b. Then click **New** to create a new domain object named Histogram Analysis with type Analysis. Click on its **Processes** tab and add a process called Plot Histogram to this domain object.

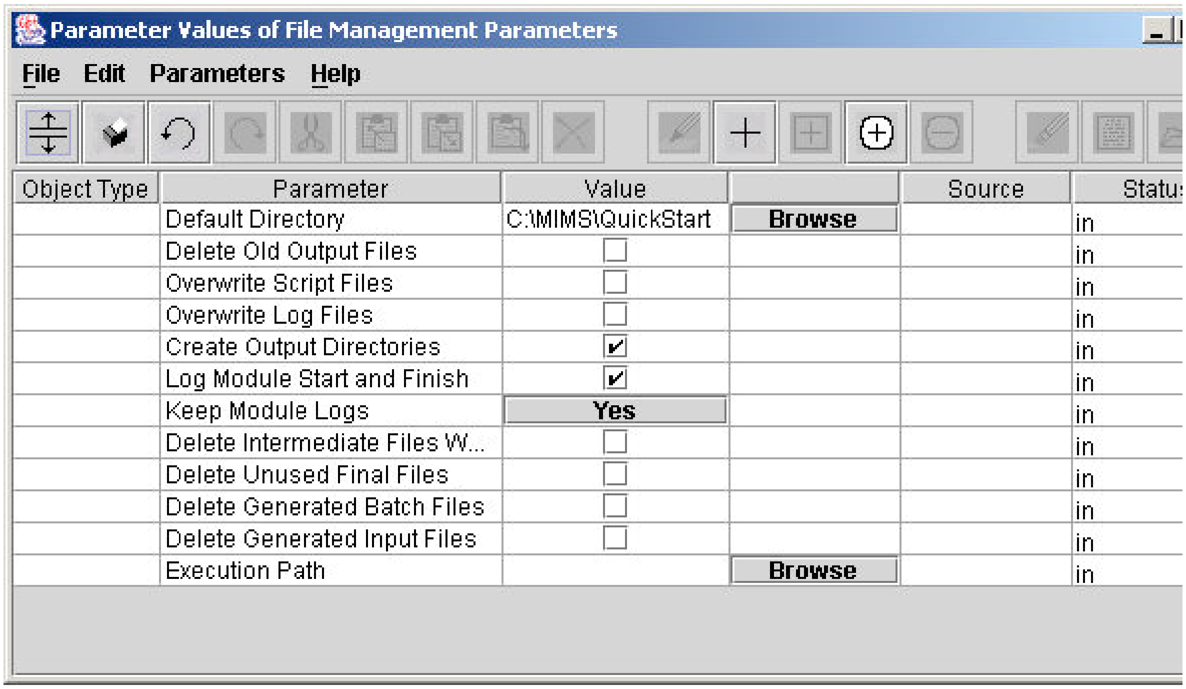


c. Minimize the Histogram Plotter, Histogram Analysis, and Monte Carlo windows.

Expand to Employ a Monte Carlo Approach (QuickStart)

Step 4. Create the Scen_Monte Carlo Scenario

a. Click **New** on the *My QuickStart* window to create a new scenario named Scen_Monte Carlo for the Monte Carlo study. Click **Yes** to open the new member. Click **Edit** on the File Management Parameters group to set the default directory to match the QuickStart directory used for the Dep_Scenario scenario (click the **Browse** button beside the default directory parameter). Make sure the box beside Delete Old Output Files is not checked. Then close the dialog box showing the File Management Parameter group.



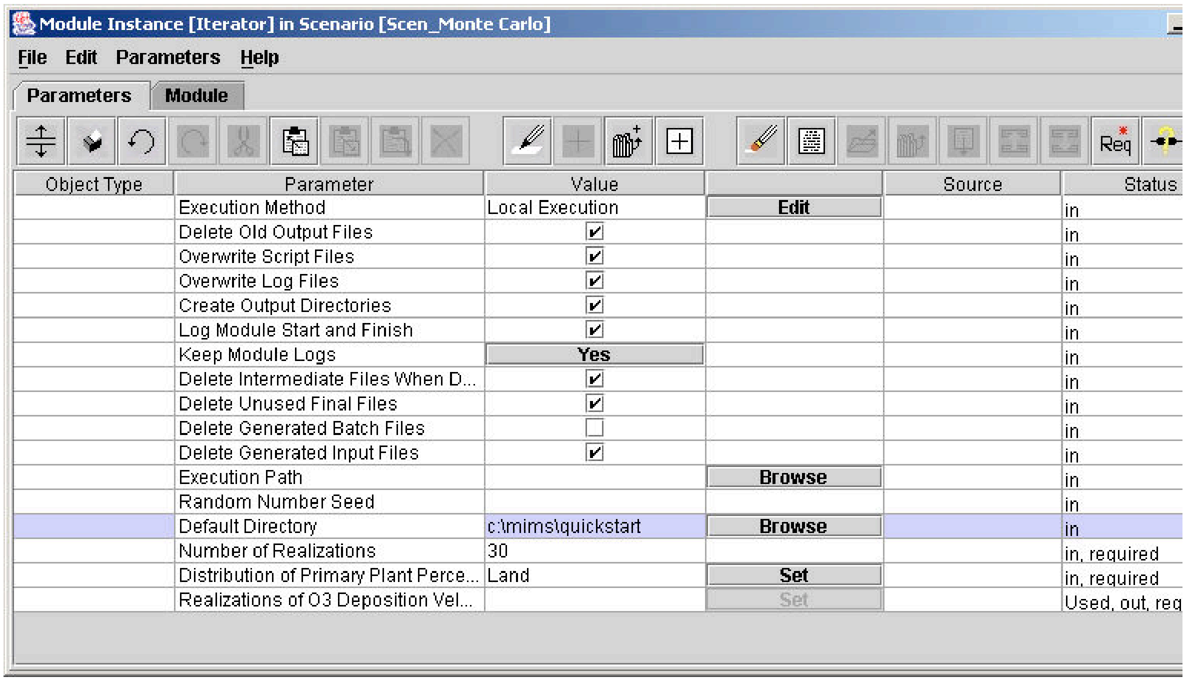
Object Type	Parameter	Value		Source	Statu
	Default Directory	C:\MIMS\QuickStart	Browse		in
	Delete Old Output Files	☐			in
	Overwrite Script Files	☐			in
	Overwrite Log Files	☐			in
	Create Output Directories	☑			in
	Log Module Start and Finish	☑			in
	Keep Module Logs	Yes			in
	Delete Intermediate Files W...	☐			in
	Delete Unused Final Files	☐			in
	Delete Generated Batch Files	☐			in
	Delete Generated Input Files	☐			in
	Execution Path		Browse		in

b. To add the Monte Carlo framework object to the scenario, right-click in the graph view of the scenario window and choose the **Add Framework Object** from the menu. An incomplete circle appears, indicating that the Default Directory parameter of the Iterator process has not yet been set.

c. To set the Default Directory parameter to match the default directory for the Scen_Monte Carlo scenario, click on the "+" sign in the Monte Carlo object. Double-click on the Iterate process to create a window titled *"Module Instance [Iterator] in Scenario [Scen_Monte Carlo]*. On the Parameters tab of this new window, click on the Default Directory parameter and then on the **Connect to**

Parameter (-■-) icon.

Select Scen_Monte Carlo in the dialog box that appears and prompts you for the connection, then click **OK**. A new dialog box appears, and you should click on the **Create Connections** button. The Default Directory for the nested scenario and the Scen_Monte Carlo scenario is now identical. Close the module instance window. You will now see an open green circle beside the Monte Carlo framework object.

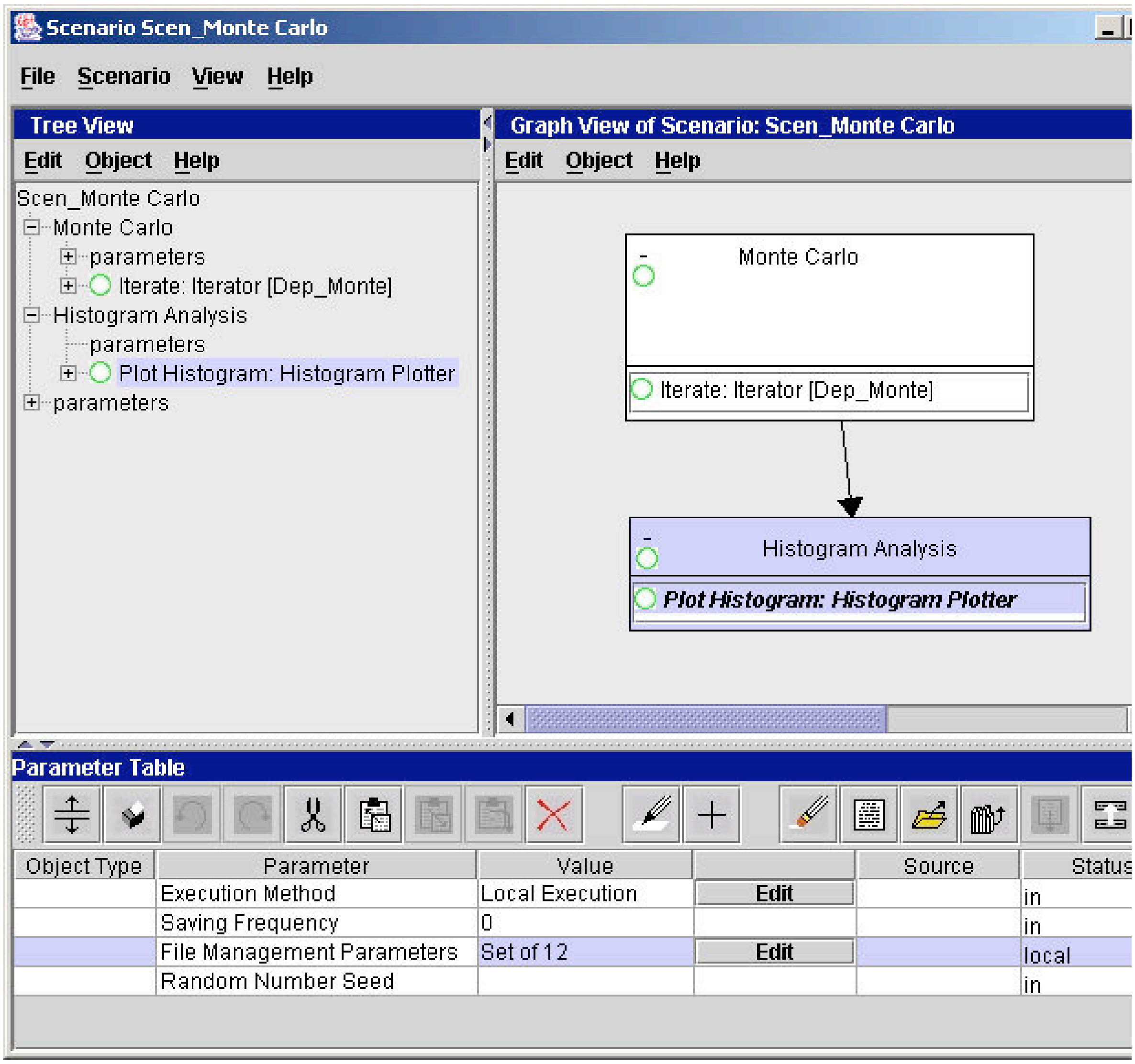


Module Instance [Iterator] in Scenario [Scen_Monte Carlo]

Object Type	Parameter	Value		Source	Status
	Execution Method	Local Execution	Edit		in
	Delete Old Output Files	☑			in
	Overwrite Script Files	☑			in
	Overwrite Log Files	☑			in
	Create Output Directories	☑			in
	Log Module Start and Finish	☑			in
	Keep Module Logs	Yes			in
	Delete Intermediate Files When D...	☑			in
	Delete Unused Final Files	☑			in
	Delete Generated Batch Files	□			in
	Delete Generated Input Files	☑			in
	Execution Path		Browse		in
	Random Number Seed				in
	Default Directory	c:\mims\quickstart	Browse		in
	Number of Realizations	30			in, required
	Distribution of Primary Plant Perce...	Land	Set		in, required
	Realizations of O3 Deposition Vel...		Set		Used, out, req

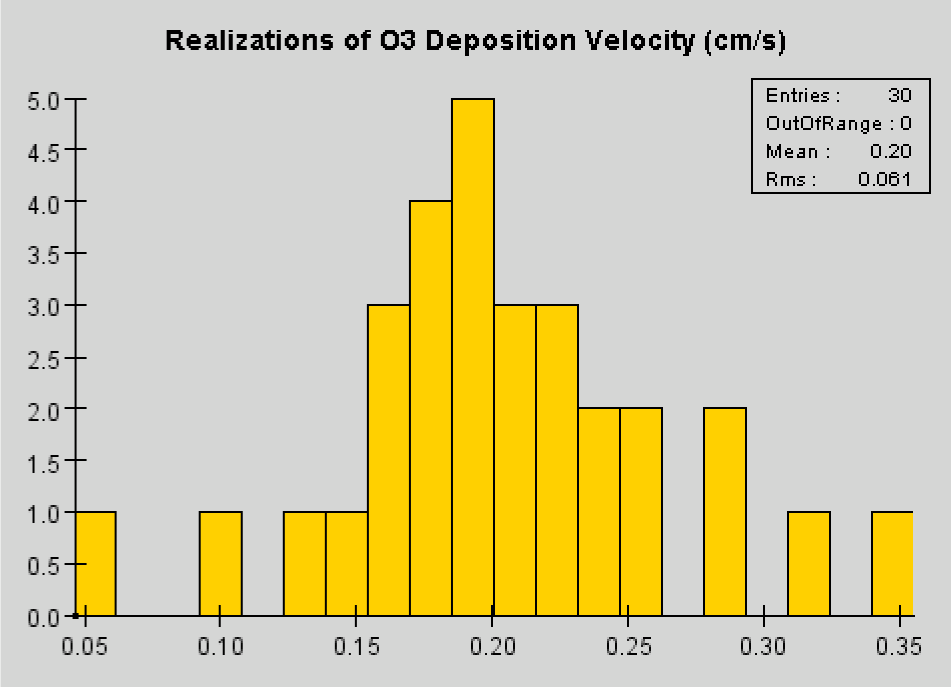
d. Also add the Histogram Analysis domain object to the Scen_Monte Carlo scenario (right-click on the open area of the graph view). Click on the "+" sign in the Histogram Analysis domain object. To set the module instance to Histogram Plotter for the process Plot Histogram, right-click on the domain object and choose **Set Module Instance** from the menu.

Right-click on the Histogram Analysis domain object, click **Add Parameter Source**, and drag the arrow down to the Monte Carlo framework object. When the *Select Connections* dialog box appears, only highlight the first row, then click the **Create Connections** button.


Scenario Scen_Monte Carlo
File Scenario View Help
Tree View
Edit Object Help
Scen_Monte Carlo
Monte Carlo
parameters
Iterate: Iterator [Dep_Monte]
Histogram Analysis
parameters
Plot Histogram: Histogram Plotter
parameters
Graph View of Scenario: Scen_Monte Carlo
Edit Object Help
Monte Carlo
Iterate: Iterator [Dep_Monte]
Histogram Analysis
Plot Histogram: Histogram Plotter
Parameter Table

Object Type	Parameter	Value		Source	Status
	Execution Method	Local Execution	Edit		in
	Saving Frequency	0			in
	File Management Parameters	Set of 12	Edit		local
	Random Number Seed				in

e. Choose the **Scenario/Execute Ready Modules** menu item and examine the histogram plot that is created. Right-click the mouse on the plot, and uncheck the box beside *Show Error Bars*. Then the plot should appear similar to this:



f. To produce precisely the same histogram as the one shown above (bar colors may still vary), set the value for the Random Number Seed parameter in the Scen_Monte Carlo scenario window to 5. Then **Reset All** and **Execute the Ready Modules** again through the **Scenario** menu items.

When the directory paths are properly set, the Scen_Monte Carlo scenario in the Demo QuickStart should produce the same histogram.

Terms and Concepts

This chapter details how to make specifications for the members available within MIMS, and the Terms and Concepts chapter describes the functions for these members. Within this chapter, you will find pages about the following:

Projects and Scenarios

- MIMS Project
- Scenario

Objects

- Domain Object
- Framework Object
- Iterator Framework Object
    - *Simple Monte Carlo*
    - *Simple Monte Carlo Max*
    - *Synchronized List*
- Temporal Iterator
- Plot Page Object

Modules

- Module
- Module Instance
- Process
- Embedded Analysis Modules
    - *Histogram Plotter*
    - *Time Series Plotter*
- External Analysis Modules
    - *Plot Page Object*
    - *Scatter Plot Module*
    - *Time Series Plot Module*
- Generate Random Time Series
- Columnar File Time Series Module

Parameters

- Parameter Concept
- Parameter Connection
- Parameter Group
- Chemical Mechanism
- Chemical Mechanism Family
- IsRequired Formula
- Status Field

Projects and Scenarios

MIMS Project (presented in the Project Window)

A MIMS Project is the highest level that you view within the MIMS framework and serves as a library of all of the information involved in a study. The MIMS project holds all of the scenarios, domain objects, framework objects, modules, parameters, and other members that are used within a study. Within a given MIMS project, all members are available for any scenario and do not have to be re-created.

The MIMS project for a session is created or chosen from the Project Selection Window (first screen displayed when the MIMS program begins).

Scenario (presented in the Scenario Window)

A Scenario represents a complete simulation and holds all of the information necessary to reproduce the simulation. A scenario may start by creating a model's input files, then execute different models, and end with the analysis of the output. The Scenario is the second hierarchical level in MIMS; most MIMS Projects are composed of one or more Scenarios.

A MIMS Scenario arranges all of the relationships among the MIMS building blocks (MIMS Domain Objects, MIMS Framework Objects, MIMS Processes, MIMS Modules, and Parameters) so they can be executed. To choose a MIMS Scenario, look under the Project Window. Under the **Category** menu, select **Scenario** to view the available scenarios.

You should save each scenario from which you plan to gather results. Then you should duplicate and modify the copy of that scenario when you wish to change any element:

- framework object (e.g., MIMS Iterator),
- domain object,
- process,
- module, or
- parameter.

Use the **Duplicate** button in the project window to copy scenarios.

A scenario may also behave as a module within another scenario. On the Scenario window, choose the **Scenario/Edit Parameter Values** menu option for the nested scenario. A dialog box appears. Set the object type and process name to the values that will be used in the outer scenario. When you set a module in the outer scenario, the list of choices will now include the scenario you wish to nest. Remember that the only parameters that can be passed into and out of the nested scenario are the ones specified as scenario parameters (for the nested scenario).

Objects

Domain Object (presented in the Domain Object Window)

A Domain Object represents a grouping of MIMS Processes, MIMS Modules, and Parameters that all share a common theme or area of interest within a simulation. The most common domain objects are individual media (Atmosphere, Land, Lake Superior, Groundwater, etc.), but domain objects also often represent interfaces (Atmosphere-Land, Groundwater-Humans, etc.). In the My QuickStart tutorial, a domain object is also used to represent the Analysis area of the simulation.

A MIMS Scenario is made up of one or more domain objects and Framework Objects. The Domain and Framework Objects represent the third hierarchical level of the MIMS framework (under projects and scenarios). Each domain object holds information describing its MIMS Processes, MIMS Modules, and Parameters. Each domain object may hold one or more processes that are used to describe the functions within the object's area of interest within the simulation. All parameters that are output from the domain object's processes are also described at the object level, and input parameters may also be specified at the object level.

In the Scenario Window, the domain objects are displayed. The domain objects are shown as subunits on the Tree View section of this window and as boxes on the Graph View section. Domain objects can be created directly from the Project Window. The specifications about domain object type, the domain object parameters, and the values can be set from the domain object window opened from the project window, but parameter connections cannot be established to modules or scenarios. The parameter values and connections to modules and scenarios can be set when the you open the domain object window from within the scenario, but the other specifications may not be altered.

Sample Domain Object: Atmosphere		
Sample Modules	**Sample Processes**	**Sample Parameters**
MM5 meteorology preprocessor	Calculate meteorology	Initial conditions Boundary conditions
CMAQ air quality model	Calculate air quality	Daily emissions Receptor latitude, longitude, and elevation Output file name

When a domain object is created, you specify its type. Only modules with that same type may be included in that domain object.

Framework Object (presented in the Framework Object Window)

A Framework Object represents a grouping of MIMS Processes, MIMS Modules, and Parameters that all share a common functionality within a simulation. The tasks of framework objects are typically independent of the environmental models and instead represent computational functions, such as advancing time steps or repeatedly conducting a process.

A MIMS Scenario is made up of one or more Domain Objects and framework objects. The domain and framework objects represent the third hierarchical level of the MIMS framework. Each framework object (within a scenario) holds all of the information describing its Processes, Modules, and Parameters. Each framework object may hold one or more processes that are used to describe the functions within that grouping of the simulation.

In the Scenario Window, the MIMS framework objects are illustrated. The framework objects are shown as subunits on the Tree View section of this window and as boxes on the Graph View section.

The following framework objects have been created for use in MIMS:

- Iterator Framework Object
- Simple Monte Carlo Iterator - reads the last value from a time series
- Simple Monte Carlo Iterator (max) - reads the highest value from a time series
- Synchronized List Iterator

Iterator Framework Objects

The MIMS Iterator Framework Objects can be useful tools for sequencing multiple runs of:

- Sequential time periods (e.g., multiple simulations of weeks in a year)
- Different time periods (e.g., a time period in July 1995 and one in August 1997)
- Different emissions control strategies and a baseline
- Projections of control strategies out to multiple future years
- Different chemical mechanisms so that their output files can be compared
- Monte Carlo simulations
- Calibrations and optimizations

When you create an Iterator framework object, an object window appears. If you click on the **Processes** tab in the object window, a process called "Iterate" appears with a module instance called "Iterator". Double-click the mouse on the module instance to open a module instance window.

The *Module Instance Iterator* window appears, and the **Parameters** menu lists two additional items not found in other object windows. The **Set Iterand...** menu item is used to set the module/scenario that you want to run repeatedly and also prompts you for the parameters that should be passed in and out of the nested module/scenario. Select the **Show Module** menu item to display the associated module.

In addition to these two menu items, the *Module Instance Iterator* window includes an additional tab titled **Iterating Lists**. This tab currently includes buttons for the **Set Iterand...** and **Show Module** commands. The **Set Iterand** button will lead you through panels to specify the iterative procedure. Unlike other module instance windows, the **Module** tab on the *Module Instance Iterator* window does not include lines for the execution command, control files, or a wait-for-completion flag.

Below you will find more details on the following MIMS iterators:

- TemporalIterator (controls time stepping and is detailed separately)
- Simple Monte Carlo Iterator - reads the last value from a time series
- Simple Monte Carlo Iterator (max) - reads the maximum value from a time series
- Synchronized List Iterator - changes an input parameter with each successive run based on values in a list

When iterators appear as objects in the Graph View of scenario windows, a progress bar is included that tracks the number of iterations during an execution.

Histograms are often an effective way to display results from a Monte Carlo simulation, so MIMS provides a Histogram Plotter module.

The ***Simple Monte Carlo Iterator*** object can be used to send random numbers to a module/scenario for a particular parameter and track the effect on the output values. When you create a Simple Monte Carlo Iterator object with a time series output, MIMS will track the final value from the time series. To track the maximum value in the time series, you should employ the Simple Monte Carlo Iterator (Max).

To generate this iterator, create a Simple Monte Carlo Iterator in the project window by clicking on the **New** button. The framework object window will appear for the iterator. Click on the **Processes** tab. Highlight the Iterate module, then press the **Edit Module Instance** button to make the module instance window appear.

From the *Module Instance Iterator* window, click on the **Parameters/Set Iterand** menu item. Choose the desired module/scenario as the member. the *Inputs/Outputs* dialog box appears. On the **Inputs** tab, select the variable(s) you would like to randomize in the Unselected Items list and click **Add**. The parameter will now be listed in the Selected Items List. On the **Outputs** tab, select the desired output parameter(s) in the Unselected Items list and click **Add**. The parameter will now be listed in the Selected Items List. Click **OK**. The dialog box closes. At least three new parameters are added to the parameters list in the module instance window:

- Number of Realizations
- Distribution of "input parameter"
- Realizations of "output parameter"

The number of realizations should be set to the desired number of iterations. Click the **Set** button beside the Distribution of "input parameter". Choose a uniform or normal distribution, and then assign a name and parameters to the distribution. After choosing the parameters, click **OK**.

The ***Simple Monte Carlo Iterator (Max)*** object can be used to send random numbers to a module/scenario for a particular parameter and track the effect on the maxima of the output values. When you create a Simple Monte Carlo Iterator (Max) object with a time series output, MIMS will track the maximum value from the time series.

To generate this iterator, create a Simple Monte Carlo Iterator (Max) in the project window by clicking on the **New** button. The framework object window will appear for the iterator. Click on the **Processes** tab. Highlight the Iterate module, and press the **Edit Module Instance** button to make the module instance window appear.

From the *Module Instance Iterator* window, click on the **Parameters/Set Iterand** menu item. Choose the desired module/scenario as the member. An *Inputs/Outputs* dialog box appears. On the **Inputs** tab, select the variable(s) you would like to randomize in the Unselected Items list and click **Add**. The parameter will now be listed in the

Selected Items List. On the **Outputs** tab, select the desired output parameter(s) in the Unselected Items list and click **Add**. The parameter will now be listed in the Selected Items List. Click **OK**. The dialog box closes. At least three new parameters are added to the parameters list in the module instance window:

- Number of Realizations
- Distribution of "input parameter"
- Realizations of "output parameter"

The number of realizations should be set to the desired number of iterations. Click the **Set** button beside the Distribution of "input parameter". Choose a uniform or normal distribution, and then assign a name and parameters to the distribution. After choosing the parameters, click **OK**.

Instead of iterating on random numbers, the ***Synchronized List Iterator*** will repeat a module using an input parameter whose values are taken from a list. Each successive iteration represents a different list item. The list could specify numbers, text strings, or even file names and directories. This framework object might be used to configure a series of model simulations that consider different cases. For example, each string might hold the name of a different chemical that should be considered in a hazardous waste project. For an air quality strategy design project, each string might hold the name of a control strategy approach (e.g., "Baseline," "Bold Reductions," and "Bold with Constraints").

To generate this iterator, create a Synchronized List Iterator in the project window by clicking on the **New** button. The framework object window will appear for the iterator. Click on the **Processes** tab. Highlight the Iterate module, and press the **Edit Module Instance** button to make the module instance window appear.

From the *Module Instance Iterator* window, click on the **Parameters/Set Iterand** menu item. Choose the desired module/scenario as the member. An *Inputs/Outputs* dialog box appears. On the **Inputs** tab, select the variable(s) you would like to randomize in the Unselected Items list and click **Add**. The parameter will now be listed in the Selected Items List. On the **Outputs** tab, select the desired output parameter(s) in the Unselected Items list and click **Add**. The parameter will now be listed in the Selected Items List. Click **OK**. The dialog box closes. The module parameters are then added to the parameters list in the module instance window.

After the input/output parameters are selected *or* when you click on the button marked **Edit Synchronized Lists** on the Iterating Lists tab, the Synchronized Input Lists Editor appears. The purpose of this window is to allow you to match and order the parameter values that will be used in the iterations.

Temporal Iterator

The Temporal Iterator framework object can provide time stepping for modules that do not have their own time stepping devices or when an additional level of time stepping is required. Examples include the following:

- A set of models that step forward in time together might be incorporated into the MIMS framework by allowing the Temporal Iterator to control the time stepping for all of the models instead of trying to coordinate the time step loops in each model separately.
- A multiple-day air quality episode might be divided into 1-day segments to reduce file sizes. In such a case, even though the air quality model has its own internal time steps, it must be restarted for each day in the episode. The Temporal Iterator could initiate the daily restart.

The inputs to the Temporal Iterator are the *starting time*, the *duration* of the iterator's time step, and the *number of time steps* to consider. For instance, a starting time of 1:00 pm, a time step of 1 hour, and two steps would invoke the dependent models for times corresponding to 1:00 and 2:00 pm.

For a module to use the iterator's output value, you must establish a Parameter Connection to the temporal iterator's output parameter and include this variable in the input to at least one module instance.

Other standard iterators are described in the section on Iterator Framework Objects.

Plot Page Object

See the discussion within the External Analysis Module section.

Modules

Module (presented in the Module Window)

A MIMS module is a software program that the MIMS framework invokes. Each module can only implement one MIMS Process and performs according to the module instance’s parameter specifications. A Module Instance describes a single invocation of a MIMS module, according to the Parameters specified in the MIMS module instance window.

User-created MIMS modules can focus on a particular environmental media. The environmental media would be specified as the module's type, and the module instance could then be included in domain objects of the same type. Sample modules (and their types) include:

- BASINS (surface water)
- CCTM (atmosphere-land)
- MM5 (atmosphere)
- PRSYM (groundwater)
- SMOKE (atmosphere)
- SWOOMS (surface water)
- TOPLATS (land surface)

To view the details about a module, you may access the Module window through the Project window or the Module Instance window. As an example, from the Project Window, choose Default External Module as the category and then double click on the desired module. From the Module Instance Window, select the **Parameters/Show Module** menu item. The Module Window appears and describes the settings for this module.

Many MIMS modules are incorporated into the MIMS framework for ready user access. They include:

- Embedded Analysis Modules (histogram, time series, and x-y plotter modules)
- External Analysis Modules (time series and scatter plot modules)
- Generate Random Time Series Module
- Columnar File Time Series Module
- Custom GUI External Module

In addition, the Default External Module allows you to invoke external programs. The Models3 Module acts as a default external module to invoke the programs associated with the Models-3 Air Quality Modeling System.

All modules include a compatible object type, a process that they implement, and the associated parameters. Many modules also include preprocessing instructions, execution commands, control files, custom GUI objects, columnar files, and/or instructions to wait for process completion.

Module Instance (presented in the Module Instance Window)

A module instance describes a single operation of a MIMS Module, according to the Parameters specified for that particular MIMS Process. To execute a module instance, connect all the necessary parameters so that a green open circle appears beside the process in the scenario window. Then choose the **Scenario/Execute Ready Modules** menu item in the Scenario window. After a module instance has successfully executed, a check mark appears beside the process in the Scenario Window.

Process

A MIMS process describes a single action within the MIMS framework. In most cases each process executes a single module instance with the specifications determined by the input parameters for the module instance. The MIMS module instance will gather the input Parameter values, operate on them using a MIMS Module, and export the data to the specified output parameter files. This execution series of a MIMS process is referred to as a Module Instance.

One or more processes may be contained in a MIMS Domain Object.

To locate a particular process within the MIMS framework, go to the MIMS Scenario window (the window with Tree and Graph views). Click on the "+" next to the domain object to see its list of processes (or double-click on the domain object name to view all of the processes and parameters associated with that domain object).

The process name appears before the colon, and the associated module name is shown after that. Right-click on the process to set, update, or delete the associated module instance. You may also choose to assign a process to a domain object but not assign a module instance. In this case, computations will not be performed, but the process placeholder will describe the physical/chemical characteristics and operations of the domain object.

If you double-click on the process in the Scenario window (and a module instance is associated with the process), a new window appears. The new window is named Module Instance <*module name*> and allows you to view the module and the parameter specifications.

Embedded Analysis Modules

The MIMS framework provides some simple analysis tools that allow you to plot and print their output information. These modules are easy to distribute along with the framework but will not have all of the functionality of the External Analysis Modules. One weakness of the embedded analysis modules is that they do not handle missing data points.

To use one of the embedded analysis modules, first create the module from the Project Window by selecting the desired plot type (histogram, time series plotter, etc.). Then, create a domain object with a type of "Analysis" from the project window. In the domain object window, click on the **Processes** tab, add a process, and set the process's module instance to the desired plot type.

You may choose from the following embedded analysis modules and see them described further below:

Histogram Plotter

Time Series Plotter

The ***Histogram Plotter*** is a pre-programmed module within MIMS. The module will create a screen showing up to four histograms simultaneously. It implements the Plot Histogram process and has an Analysis object type.

In the Parameters table, you must set at least the first series of data through a parameter connection to another object. Other parameters that can be specified in the module instance window are:

- Number of Bins -- Select the number of bins for the data. If the number of bins is not specified, the program will determine a reasonable default number.
- Tile Histograms? -- When the box is checked, multiple histograms will appear on the same page of the plot window. When the box is not checked, each histogram appears on its own tab.
- Series 2, 3, and 4 -- Show other histograms in the same plot window.

When you execute the scenario, a histogram plot window appears. You may right-click on the plot to change the error bars, colors, axis labels, titles, etc. Right-clicking on the plot also presents the choices to print or save the image to a file.

The ***Time Series Plotter*** is a pre-programmed module within MIMS. The module creates a screen showing up to 10 time series simultaneously. It implements the Plot Time Series process and has an Analysis object type.

In the Parameters table, you must set at least the Time Series 1 by a parameter connection to another object. Other parameters that may be specified in the module instance window are:

- Window Title -- Set the title for the window. This does not set the title for the plot but just the window that holds it.
- Time Series 2 and higher -- Show other time series on the same plot window.

When you execute the scenario, a time series plot window appears. You may right-click on the plot to change the ranges, colors, axis labels, titles, etc. Right-clicking on the plot also presents the choices to print or save the image to a file.

External Analysis Modules

The MIMS framework provides some complex analysis tools that allow you to plot and print their output information. Some external analysis modules allow direct output (no user interaction) to graphic files and printers. These external analysis modules are distributed separately from the framework (unlike the embedded analysis modules).

To use one of the embedded analysis modules, first create the module from the Project Window by selecting the desired plot type (e.g., Plot Page Object). Then, create a domain object with a type of "Analysis Engine" from the project window. In the domain object window, click on the **Processes** tab, add a process, and set the process's module instance to the desired plot type.

You may choose from the following external analysis members and see them described further below:

- Plot Page Object
- Time Series Plot Module
- Scatter Plot Module

These MIMS members cannot operate until you have installed the R statistical software.

The ***Plot Page Object*** is a Framework Object within MIMS. The Plot Page object has an Analysis Engine object type, so it enables you to activate the R statistical software for creating graphs. This framework object is permitted to contain Module Instances that operate with the external R software (e.g., Time Series Plot Module and Scatter Plot Module). Up to 12 plots may be associated with a Plot Page object; a graph is added to the single output page for each plot that is included.

Four parameters are associated with the Plot Page object:

- Page Title: Specify the text that should appear at the top of the plot page.
- Output Type: Choose to display the results to the screen, a PDF file, or a JPEG file.
- Output File: Specify the path, file name, and extension where the output file should be stored.
- Number of Columns: Specify the number of plots across the window.

The ***Time Series Plot Module*** represents a MIMS module with an Analysis Engine object type. Use the R statistical software to execute this module to create a graph and display it as a pdf file.

The Time Series Plot Module window should be set as follows:

- Compatible Domain Object Type to Analysis Engine.
- Name of Process This Implements to Plot*. The Plot* instruction tells the MIMS Framework to use the internal process associated with this module.
- Perform Preprocess (allows preliminary execution of arbitrary script code [e.g., to perform consistency checks among parameters]).

The input Parameters for the Time Series Plot Module specify the following:

- Plot Title appears directly above the graph.
- S1, S2, S3, and S4 Values specify the time series data (up to 4 time series) to be plotted. Click on the **Set** button in this parameter row. You are presented with choices to plot CASTNET Time Series Values, Time Series from File Values, Time Series from User Defined Function, Time Series Constant Values, or Time Series in Memory Values.
- Plot Style lets you choose to show lines, points, or both.
- Log Y Scale is a Boolean expression that can be turned on or off, based on the desired display.

The ***Scatter Plot Module*** represents a MIMS module with an Analysis Engine object type. Using the R statistical software, an execution of this module will create a graph and display it as a Adobe pdf file.

The Scatter Plot Module Window should be set:

- Compatible Domain Object Type should be set to Analysis Engine.
- Name of Process This Implements should be set to Plot*.
- Perform Preprocess allows preliminary execution of arbitrary script code (e.g., to perform consistency checks among parameters).

The input Parameters for the Scatter Plot Module specify the following:

- Plot Title appears directly above the graph.
- X Values specifies the list of numbers to be used on the ordinate axis. Currently the values must be specified individually as Double Series in Memory Values.
- Y1, Y2, Y3, and Y4 Values specify the abscissa data (up to 4 series) to be plotted. Currently the values must be specified individually as Double Series in Memory Values and must match the X Values.
- Connect Points is a Boolean expression that can be turned on or off.
- Minimum X, Maximum X, Minimum Y, and Maximum Y specify the extents of the coordinates on the graph.
- Log Y Scale is a Boolean expression that can be turned on or off, based on the desired graph.

Generate Random Time Series (GRTS) Module

A MIMS Module is a software program that the MIMS framework invokes. Each module can only implement one MIMS Process and performs according to the Module Instance's parameter specifications. The Generate Random Time Series (GRTS) module enables you to assign random numbers within a certain distribution to every value in a time series.

The GRTS module was designed to add a random element to data sets collected over a period of time. The GRTS module is a particularly useful add-on when a data set is simulated by the computer and needs to express potential measurement errors. The GRTS module generates numbers that may be added or subtracted from the base simulation.

The GRTS module window (titled "Generate Random Time Series"+module name) lists the MIMS Process and Parameters associated with the module. The process is fixed to be Generate Time Series, and the **Browse** button beside the process cannot be used to select another process.

In addition to the process and parameters list, the GRTS module window presents the Compatible Domain Object Type (the domain object type that may contain the GRTS module, not the name of the domain object). By default, the type is Noise.

The Parameters that are input and output from the GRTS module are listed in the module window. The list contains the following information:

- Object Type (MIMS Domain Object or Framework Object that generates the parameter)
- Parameter (lists the name of the parameter)
- Internal name (parameter name within the module)
- Value (parameter value or file name where information is stored)
- Additional Type Info (units, file locations, time zones, field delimiters, labels, etc.)
- Status (in, out, inout, required, used, local, *, and locked)

Since the GRTS module operates on a fixed process, the parameters cannot be added, altered, or removed in the GRTS module window. You are only permitted to assign values to the first two parameters.

The first input parameter, "Current Time", is connected to a TemporalController object type, indicating that the values generally come from a temporal iterator in the scenario.

The second input parameter lets you specify the "Distribution" of the random values. You may currently select normal distributions or uniform distributions. When you click the **Set** button beside the Distribution parameter, you are prompted to name and specify the parameters for the parameter of interest.

"Random Values" is the output parameter from the GRTS module. It has the Floating Point Time Series (from Memory) parameter type and can be used by other domain and framework objects (e.g., time series plotting members).

The actions of the menu options and screen buttons are described in How to Use MIMS Module Windows.

Columnar File Time Series (CFTS) Module

A MIMS Module is a software program that the MIMS framework invokes. Each module can only implement one MIMS Process and performs according to the module instance's Parameter specifications. The Columnar File Time Series (CFTS) module enables you to generate files with a particular format.

A method to construct a sample CFTS Module is described below.

A CFTS lists dates and times in one column (or possibly listing month, date, year, etc. in multiple columns) and the associated observational data in other columns of the same file. The MIMS framework supports tab-, space-, and comma-delimited file formats. A sample comma-delimited CFTS might begin:

```
Time, Concentration, Temperature

01/01/2003 00:00:00, 163, 15

01/02/2003 00:00:00, 185, 16

01/03/2003 00:00:00, 137, 18

...
```

The CFTS Module Window (titled "Module"+module name) lists the MIMS Processes, Parameters, and the exterior program associated with the module. In addition to the parameters list, the CFTS Module window presents:

- Execution Command (the path and name for the software program as well as any command line arguments)
- Columnar File (directory path and file name for the CFTS)
- File format for the CFTS by clicking the **Edit** button (the *Edit CFTS Module* dialog box appears)
- Control files (viewer to the files read by an external module)
- Compatible Domain Object Type (the domain object type that may contain the CFTS module, not the name of the domain object)
- Name of Process This Implements (MIMS process associated with the CFTS module)

You may type directly into these fields or select **Browse** when necessary.

The ***Edit CFTS Module*** dialog box appears when you click the **Edit** button beside the name of the Columnar File. This dialog box is used to define the file format:

- Has Title Line - Check this box if the first row of the CFTS should contain column headings
- Delimiter - Choose whether tabs, spaces, or commas separate the data columns
- Use Insert Row to specify the column that contains the time series (Field Name), the column titles, the date-time format, and the indicator that data is missing for a particular value (e.g., -999) .

The Parameters that are input and output from the CFTS Module are listed in the module window. The list contains the following information:

- Object Type (MIMS Domain Object or Framework Object that generates the parameter)
- Parameter (lists the name of the parameter)
- Internal name (parameter name within the module)
- Value (parameter value or file name where information is stored)
- Additional Type Info (units, file locations, time zones, field delimiters, labels, etc.)
- Status (in, out, inout, required, used, local, *, and locked)

If opened from a Scenario Window, the parameters cannot be added, altered, or removed from the CFTS Module window. They may be adjusted if the CFTS Module window is opened from the Project Window.

The actions of the menu options and screen buttons are described in How to Use MIMS Module Windows.

Constructing a Sample CFTS Module

From the project window, click **New**. When the *Create Member* dialog box appears, choose Columnar File Time Series Module in the Modules folder. Type the name "CFTS" and click **OK**.

The module [CFTS] window appears. Beside Compatible Domain Object Type, choose Atmosphere. Beside Name of Process this Implements, type "Generate TS File."

The next step is to create the four parameters of interest::

1. Click the **Add New Parameter** (+) button to add a new parameter. Choose File Name as the parameter type, name the parameter FileNam, and set it as input and required.

2. Add a string parameter type with the **Add New Parameter** (+) button. Choose SITE_ID as the name, set it as input and required, and set the maximum string length as 20 characters.
3. Next add a floating point time series (generic) parameter type. Choose AIR_TEMP as the name, set it as input and required, and set the units to "deg C."
4. Finally add another floating point time series (generic) parameter type. Choose WATER_TEMP as the name, set it as input and required, and set the units to "deg C."

Following the creation of these parameters, double-click the mouse in the Value column and the FILENAM row. Then type "colFile" as the value and press return. The execution command and the columnar file name will use the FileName parameter for their location. Using the Python codes for calling parameters, type `notepad ${str("filenam")}` beside the execution command (if Notepad is not your default viewer, change the text accordingly). Also type `${str("filenam")}` beside Columnar File.

Double-click the mouse in the Value column and the SITE_ID row. Type "BVL" as the value and press return. Click the **Set** button in the AIR_TEMP row, and choose Time Series in Memory Value as the type. The Edit Time Series in Memory Value dialog box appears.

In this new dialog box, set the date format to MM/dd/yyyy and the time zone to EST. Check the *Apply Daylight Savings Time Adjustment* check box and then click on the **Generate Rows** button. A small dialog box appears to define the time series; set the values as follows:

Start Date: 06/01/2003
Time Step: 1
Units: days
Number of Steps: 20

Click the **OK** button to return to the *Edit Time Series in Memory Value* dialog box. Type values between 10 and 20 into the Values column. Click **OK** when this is complete.

Repeat this process for the WATER_TEMP by clicking the **Set** button in that row. However, set the number of steps to 19.

Click the **Edit** button in the Columnar File row of the Module [CFTS] window. The Edit Columnar File Time Series Module dialog box appears. Check the *Has Title Line* check box, and choose comma delimiters. Click the **Insert Row** button six times.

The field name column presents choices of Time Series Date and the parameters set in the module. You type values directly into the Title, Format, and Missing Value columns.

Fill in the six rows with the following information:

SITE_ID, station, *(blank)*, *(blank)*
Time Series Date, Year, yyyy, *(blank)*
Time Series Date, Month, MM, *(blank)*
Time Series Date, Date, dd, *(blank)*
AIR_TEMP, Air Temp (C), *(blank)*, -999
WATER_TEMP, Water Temp (C), *(blank)*, -999

Click **OK** to save this information and close the dialog box.

Create a domain object with compatible domain object type Atmosphere and a process called Generate TS File. Then add this domain object to a new scenario and set the module instance to CFTS. Then execute the scenario. The file you open should look similar to this (note that the last entry ends in the missing value of -999):

```
station,Year,Month,Date,Air Temp (C),Water Temp (C)
BVL,2003,06,01,10.0,10.0
BVL,2003,06,02,11.0,11.0
BVL,2003,06,03,12.0,12.0
BVL,2003,06,04,13.0,13.0
BVL,2003,06,05,14.0,14.0
BVL,2003,06,06,15.0,15.0
BVL,2003,06,07,16.0,16.0
BVL,2003,06,08,17.0,17.0
BVL,2003,06,09,18.0,18.0
BVL,2003,06,10,19.0,19.0
BVL,2003,06,11,20.0,20.0
BVL,2003,06,12,19.0,19.0
BVL,2003,06,13,18.0,18.0
BVL,2003,06,14,17.0,17.0
BVL,2003,06,15,16.0,16.0
BVL,2003,06,16,15.0,15.0
BVL,2003,06,17,14.0,14.0
BVL,2003,06,18,13.0,13.0
BVL,2003,06,19,12.0,12.0
BVL,2003,06,20,11.0,-999
```

Parameters

Parameter

A MIMS Parameter is any data, metadata, or placeholder that defines a system and determines (or limits) its performance. Within the MIMS Framework, parameters provide the input, output, and operational configuration of a module, domain object, scenario, or other member. Example parameters include:

- Integers and floating point numbers describing physical phenomena
- Files with input data
- Time steps
- File names for output data
- Geospatial data set
- Geographic projection
- Choice of chemical mechanism

Every MIMS parameter has a type associated with it, and most will have specified values. The following list shows some examples of the parameter types included in MIMS:

Parameter Type	Description
Boolean flag (logical)	True or false.
Chemical mechanism	Table of chemical species and reaction equations.
Date	Date-time value that includes year, month, day, hour, minute, and seconds.
Directory path name	File directory on the system drive.
Duration	Simulation time period or other length of time.
File	Name for a file that already exists when it is called. The file name directs where the simulation should get input information or place output.
FileName	Contains the name of the file and possibly path. This can be used to write an output file that will not exist until used.

Parameter Type	Description
Floating point number	Any real number.
Integer	Number whose absolute value can be expressed as a whole number.
M3IO File Variable	Reference to a variable found within a M3IO file. The m3io and ioapi formats are discussed further at http://www.cep.unc.edu/empd/EDSS/ioapi/.
Parameter Group	Set of parameters identified by a single group name
String	Text
Typed File	File name that will only allow choices that fall under a particular defined file tag

The parameter type is chosen when the parameter is created and cannot be altered. Parameters are associated with a particular MIMS member and may be viewed and set from the member's window. Within a MIMS member (e.g., scenario, domain object, or module), each parameter must have a unique name. The output parameter from one MIMS module instance or other member may serve as the input for another member through parameter connections.

Pages have been organized to assist you in choosing the appropriate parameter types and values.

Parameter Connection

Within the MIMS framework, the output Parameter from one module instance may serve as the input for another module instance, but only if the output is set at the object level. Two module instances within the same object will share parameters that are listed as domain object parameters. If a connection is set between two different objects, the parameters from the object's module instances will be passed. You can match these inputs and outputs through Parameter Connections.

The Scenario Window, Domain Object Window, Framework Object Window, and the Module Instance Window allow you to make Parameter Connections. In the Scenario window, right-click on the object requiring input, choose **Add Parameter Source** from the menu, then click on the source object. In the other windows (e.g., module instance windows), select the Parameter, click on the **Connect to Parameter** (-■■-) icon, then select the domain or framework object containing the parameter.

Within the MIMS framework, you also need to know how scenario, object, and module parameters are associated. Input parameters pass from the scenario level to the module level, and the output parameters from a module are passed up to the scenario level.

Parameter Group

A MIMS Parameter Group provides a convenient method for organizing all of the parameters necessary to execute a MIMS Scenario. The parameter group stores parameters together and can be viewed on a single screen (Parameter Group Window). By grouping parameters, you can avoid scrolling through long lists of variables that may be associated with any given module. Parameter groups also make it easier to introduce a large number of parameters to multiple members within a MIMS Project.

Since parameters within a group are considered parameters of the member themselves, you must make sure that the parameter group's elements have unique names within the member.

Any MIMS Module will draw the parameters it needs from a Parameter Group and ignore the rest. This grouping approach also allows you to see which parameters are passed between the MIMS Domain Objects and MIMS Framework Objects on a single screen, but the modules do not treat individual parameters or parameters within a group differently.

Chemical Mechanism

A Chemical Mechanism describes all of the chemical and physical processes and rates for all chemical species. The chemical processes include chemical reactions for both the species of interest and tracer species.

The Chemical Mechanism currently in MIMS focuses on atmospheric processes. Therefore, the chemical reactions (and reaction rates) may be described for the aerosol (solid), aqueous (liquid), and gaseous phases.

Physical processes are also described within the Chemical Mechanism because each chemical species may have different settling velocities, rates of conversion to other phases (e.g., from gas to aerosol), wet deposition rates, or surrogate species to be used for tracking.

Chemical Mechanisms are viewed and altered through the Edit Dialog for Mechanism.

Chemical Mechanism Family

To quantify the sensitivity that model results have to particular reactions or the reaction rate constants, research investigators may choose to vary the Chemical Mechanism. This approach is also useful to test the effect that simplifying assumptions have on the model results. For a particular MIMS Scenario, the MIMS Chemical Mechanism Family may be specified as a Parameter within a MIMS Module.

The Chemical Mechanism Family is a MIMS classification that allows investigators to group together similar chemical mechanisms, perhaps with just a single variable being altered at a time. The Chemical Mechanism Family Editor is the control window that allows you to examine and edit the family members.

IsRequired Formula

Some modules will always require that a parameter be specified externally, and other modules will only need to have the parameter when certain options are operating. In the cases where the parameter is only required under certain operating schemes, the MIMS framework uses the IsRequired field.

When the IsRequired field evaluates to True, then the parameter is necessary for the module instance to execute. The evaluation expression in the IsRequired field may contain other parameters but should always evaluate to a logical value of true or false. Use Ctrl-C and Ctrl-V to cut and paste from other formula expressions.

Here are some formula examples:

Parameter Condition	IsRequired Formula Expression
Boolean = true	${val("bool1")}
Integer > 10	${val("int1") > 10}
Boolean and integer	${val("bool1") and val("int1")>10}
String = "abcdef"	${ str("str1")=="abcdef" }

To understand more about how to construct the formulas, see the section on Calling Parameters.

Status Field

The Status field is shown for all Parameters in the Module window, Module Instance window, Domain Object window, Framework Object window, and Scenario Parameters window. It indicates how the parameter should be used during a module instance. Most of the parameter status fields (in, out, inout, locked, local, and required) are set when the parameter is created using the Add Parameter function. After this, the input/output/local status and locked status of a variable cannot be altered.

The table below gives a brief description of the status conditions and how to set them:

Set by...	Function	Appears as...
Is Input	Used as input during the module instance.	in
Is Local	When a scenario is used as a module, the local scenario parameters are not visible or accessible outside this module.	local
Is Output	Created as output during the execution of the module instance.	out
Is Required	Necessary for successful completion of the module instance. If a file parameter has not yet been set, the text should appear red.	required
Is Input and Output	Both used as input and created as output during the execution of a module instance.	inout
Is ReadOnly	May be altered by modules or a connection to another parameter but not directly by the user.	locked
Parameter connection	Indicates that another module instance or any domain object uses the parameter's value.	used
Design of modules	Ignores the parameter names and domain object types when matching parameters. This setting is commonly used for plotting tools that operate on many parameters.	*

How to Use MIMS

This chapter describes the members available within MIMS, and the How To Use MIMS chapter details how to make specifications for these members. Within this chapter, pages are about the following topics:

Projects and Scenarios

- Project Selection Window
- Project Window
- Scenario Window
- Edit Administration Information Window
- Importing and Exporting

Objects

- Domain Object Window
- Framework Object Window

Modules

- Module Window
- Module Instance Window
- Synchronized Input List Editor Window

Parameters

- Calling Parameters
- Scenario Parameters Window
- Parameter Group Window
- Chemical Mechanism Family Editor
- Edit Dialog for Mechanism
- Grid Family Window

Projects and Scenarios

Project Selection Window

The *MIMS Project Selection Window* displays the list of MIMS Projects that can be accessed by the MIMS framework . Double-click on a project name to call up a Project window.

The **File** menu on this window provides ways to accomplish common file management tasks:

- **New Project** – creates a new project from scratch
- **Open Project** – after selecting a project, this option calls up the project window
- **Rename Project...** – gives a new name to the selected project
- **Delete Project** – permanently removes a project
- **Import Project** – brings new files into the MIMS framework as projects
- **Export Project...** – creates a file that describes the project in XML or binary format
- **View System Logs...** – opens a viewer that lists the Standard or Error Log for this MIMS session
- **Close** – closes the opening MIMS window
- **Edit Administration Information** – opens a window allowing system administrators to assign processors and viewers
- **Exit MIMS Framework** – closes all windows and exits the MIMS framework program

Project Window

The Project window (titled with the Project name) lets you view all of the members available in your MIMS Project. Just select the type of member (e.g., Chemical Mechanism Family, Domain Object, Framework Object, Grid Family, Module, or Scenario) you want to see in more detail from the **Category** pull-down menu.

Double-click on the project member you want to see in more detail, and a new window will be called up to describe that member.

The actions of the menu options and the screen buttons (▭)are described below:

Identifier	Function	Window Access	Menu Access
Close Project	Closes this window but not the MIMS framework (unless this was the last open MIMS window)		File
Delete	Erases the selected member(s)	▭	
Duplicate...	Copies the selected member and gives it a new name	▭	
Edit Administration Information	Opens a window allowing system administrators to assign processors and viewers		File
Edit Project Description	Allows you to describe the current study and its members. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Edit
Edit Predefined Domain Object Types	Allows you to create a preferred names list for object types		Edit
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework		File

Identifier	Function	Window Access	Menu Access
Export Members...	Creates a file that other MIMS Projects can access. The file describes members such as scenarios, domain objects, framework objects, modules, or processes		File
Import Members...	Prompts you to select existing members to be included in the current project		File
Load User Defined Members	Allows you to import new types of members (classes) by specifying their file locations		File
New	Creates a new member within the current project	▭	
New Project...	Creates a new project and the associated project window		File
Open	Opens a window describing the selected member	▭	
Open Project...	Opens an existing project and the associated project window	Ctrl-O	File
Project Selection Window	Brings the list of projects into the foreground		File
Redo	Repeats previous action	Ctrl-Y	Edit
Remove User Defined Class	Removes the selected user-defined member from the types of members		File
Rename...	Assigns a new name to the selected member	▭	
Undo	Cancels previous action	Ctrl-Z	Edit
View System Logs...	Open a viewer that lists the Standard or Error Log for this MIMS session		File

Scenario Window

The Scenario window (titled "Scenario"+scenario name) lets you view all of the MIMS Domain Objects, MIMS Framework Objects, and MIMS Processes in your MIMS Scenario in two different formats. The left portion of the window provides a hierarchical tree view of the scenario; the right portion shows interconnected boxes in the graph view. The **View** menu controls which views are shown, and their sizes may be adjusted by dragging the divider bar. Both views show the same information.

A red lock appears on the right side of the window. If the lock is closed, then the modules, objects, and processes may not be altered (although the positions may be changed). However, the parameters and parameter connections may still be changed. When the lock is opened (click the mouse on the lock), then you are able to alter all aspects of the scenario.

If you select the **View/Parameter Table** menu option, a list of scenario or member parameters appears within a separate view at the bottom of the Scenario window. The Parameter Table view displays the parameters for the currently selected member.

To view the MIMS Processes for a Domain Object or Framework Object, click on the "+" symbol beside the object name; to hide the MIMS Processes, click on the "-" symbol.

Each MIMS Process name is followed by the name of the associated MIMS Module Instance. The Process name is preceded by a circle to indicate the execution status:

No symbol -- no module instance has been connected to the process yet
◌ **Missing inputs** -- some required input parameters have not been set
○ **Ready to process** -- all required input parameters have values or are connected to the output of another execution
● **Active** -- the module instance is currently executing
✓ **Successful** -- the event completed successfully
⊗ **Unsuccessful** -- the event failed to complete

The actions of the menu options and right clicks on the mouse are described below:

Identifier	Function	Window Access	Menu Access
Add Domain Object	Adds one of the project's MIMS Domain Objects to the Scenario.	Right click	Edit
Add Framework Object	Adds one of the project's MIMS Framework Objects to the Scenario.	Right click	Edit
Add Parameter Source	Builds a parameter connection (graph view only). Right click on the receiving domain object, choose **Add Parameter Source**, then left-click on the source domain object. See how parameters are associated.	Right click	
Add Process	Inserts a new process within the selected object.	Right click	Object
Allow Use as Module	If you fill in the object type and process name fields within this window, the entire scenario operates as a module within the current project.		Scenario
Clear Module Instance	Disconnects the selected MIMS module instance from the process.	Right click	Object
Close	Closes this window but not necessarily the MIMS framework.		File
Collapse All	Hides the MIMS processes associated with all of the domain and framework objects.		Object
Copy Object to Project	Creates a duplicate of the selected object in the project under a new name.	Right click	Object
Delete Object	Erases the selected domain or framework object from the scenario.	Delete key or right click	Edit
Edit Administration Information	Opens a window allowing system administrators to assign processors and viewers.		File
Edit Object	Opens a domain object window for the highlighted domain or framework object.	Right click	Edit

Identifier	Function	Window Access	Menu Access
Edit Object Description	Allows you to describe the highlighted domain or framework object. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Object
Edit Parameter Values	Opens the Scenario Parameters window and allows you to specify variables such as default paths and module instance protocols. This option cannot be selected when the Parameter Table view is shown (adjust from the **View** menu).		Scenario
Execute Ready Modules	Will execute only the module instances that are currently ready to process. **CAUTION:** Executing a MIMS scenario invokes a variety of software that has full access to your computer, including the ability to modify and delete files and invoke additional programs. Only accept and use MIMS members (modules, projects, parameters, etc.) from a source you trust.		Scenario
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework.		File
Expand All	Shows the MIMS processes associated with all of the domain and framework objects.		Object
Lock/Unlock Scenario	Locking the scenario prevents you from changing the objects, module instances, and processes but still allows parameters and parameter connections to be changed.		Scenario
Move Process up in list	Raises the highlighted process up one entry in the list of processes within a MIMS object.	Right click	Object
New Scenario...	Creates a new scenario and the associated scenario window.		File
Open Scenario...	Opens an existing scenario and the associated scenario window.		File
Print...	Sends the tree view, graph view, or parameters table to the printer.		File
Project Selection Window	Brings the list of projects into the foreground.		File

Identifier	Function	Window Access	Menu Access
Redo	Repeats the previous action.	Ctrl-Y	Edit
Remove Parameter Value Source	Removes a parameter connection.	Right click	
Remove Process	Disconnects the selected MIMS process from the domain or framework object.	Right click	Object
Rename Object	Assigns a new name to the highlighted domain or framework object.	Right click	Object
Reset All	Returns all processes that have been executed to a ready-to-execute state.	Ctrl-R	Scenario
Reset Selected	Returns the highlighted MIMS processes to a ready-to-execute state.		Scenario
Save	Saves the scenario within the project. Even if the **Save** button is not clicked, open scenarios are saved when you exit MIMS.	Ctrl-S	File
Edit Scenario Description	Allows you to describe the scenario. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Scenario
Set Module Instance	Connects a MIMS module instance to the selected process.	Right click	Object
Set Node View Size	Changes the font size in both the tree and graph views.	Right click	View
Stop All	Stops the execution of all MIMS processes that began with the **Execute All** or **Execute Selected** commands.	Ctrl-K	Scenario
Stop Selected	Stops the execution of the MIMS processes that began with the **Execute Selected** command.		Scenario
Undo	Cancels the previous action.	Ctrl-Z	Edit
Update Module Instance	Retrieves the updated version of the highlighted MIMS module from the project.	Right click	Object
Update Object	Retrieves the updated version of the highlighted MIMS object from the project.	Right click	Object

Identifier	Function	Window Access	Menu Access
View Error Log	Opens a viewer that lists the error log for the highlighted module instance.	Right click	
View Error Message	Opens a viewer that lists the error message for the highlighted module instance.	Right click	
View Execution Log	Opens the file that recorded starts and completions of the module instances.		Scenario
View Output Log	Opens a viewer that lists the standard log for the highlighted module instance.	Right click	
View System Logs...	Opens a viewer that lists the Standard or Error Log for this MIMS session.		File
View Run Script	Opens a text editor showing the batch script files for the MIMS module instance.	Right click	

Edit Administration Information Window

The *Edit Administration Information* window is composed of two tab listings: **Computers** and **Viewers**. The **Computers** tab allows you to specify the names, operating systems, and number of processors available to MIMS. Before the remote systems can be used in a secure shell, secure shell remote access should be established.

Use the **Edit/Insert Row** or **Edit/Delete Row** menu item to insert or delete rows under the **Computers** tab. Under the Fully Qualified Name column, you may enter the internet protocol (IP) address. The operating system (Windows, UNIX, or LINUX) of the remote machine and the number of available processors on the remote machine should also be entered.

The **Viewers** tab allows you to set the program with which to browse file contents. The viewer that MIMS chooses depends on the file extensions or file tags (for tagged file parameters) as specified on the **Viewers** tab. The **View File** command is generally available when you right click the mouse on a file parameter in Parameters tables.

To enable some basic viewers, select **Edit/Insert Row** and type "txt; log; xml" in the File Extensions column and "default" in the Viewer Command column. This instructs the **View File** command to open files with extensions txt, log, and xml to open under "notepad %" for Microsoft Windows and a Java viewer method for UNIX systems.

To enable viewing additional files, insert a second row. Type "nc; m3io; ioapi" under the File Extensions column and "java ioapifileviewer.session.Viewer %s" under the Viewer Command column. The m3io and ioapi formats are discussed further at http://www.cep.unc.edu/empd/EDSS/ioapi/.

The **Viewers** tab also includes a check box on each line under the heading "Use Windows Association." If this check box is checked, if MIMS is operating in a Windows environment, and if no entry is included under the Viewer Command column, then MIMS uses the default Windows viewer for that file extension.

Importing and Exporting

The MIMS framework is constructed to allow you to freely exchange files. The **File** menu in the Project Selection Window allows you to import and export projects by specifying only the format and file name.

You may import and export files in binary format (files with bin extensions) or in Extensible Markup Language (XML). The binary format results in compact files, and the xml format allows for flexible identification of information within the files. For a basic understanding of XML, you may go to the XML Frequently Asked Questions Web site.

The **File** menu in the Project Window also contains options to import and export MIMS members. When choosing to export members, you may select the entire current project, all members within one category, or the currently highlighted member. When you click on the **Export to XML** or **Export to Binary** buttons, you are prompted for the file name. To import members into the current project, you are prompted to choose the desired file. An imported project is added to the Project Selection window, and other imported members are added to the current project in the Project window.

CAUTION: Executing a MIMS scenario invokes a variety of software that has full access to your computer, including the ability to modify and delete files and invoke additional programs. Only accept and use MIMS members (modules, projects, parameters, etc.) from a source you trust. Be sure to review all modules, formulas for parameters, formulas in control file templates, and module preprocessors from outside sources before executing a scenario.

Objects

Domain Object Window

The Domain Object window (titled "Domain Object"+domain object name) lets you view the MIMS Processes and Parameters associated with the MIMS Domain Object. When domain object windows are opened by double-clicking an object within a scenario, the window is titled "Domain Object"+domain object name+"in Scenario"+scenario name; changes to the object only apply within that scenario.

The *Type* field is used to determine which modules and parameters may be connected with this domain object. For example, a domain object with a "Surface Water" type could be used to describe domain objects for both Lake Erie and the Mississippi River. Type names are case sensitive.

Click the tabs to view a list of either the parameter values or processes. The **Parameter Values** tab lists the Parameters that are associated with the domain object and their origins. The list contains the following information:

- Parameter name
- Value (output value or file name where information is stored)
- Source (MIMS Module or other member that creates the parameter)
- Status (in, out, inout, required, used, local, *, and locked)

Hold the mouse over a parameter's value box to identify the parameter type.

The specifications about domain object type, the domain object parameters, and the values can be set from the domain object window opened from the project window, but parameter connections cannot be established to modules or scenarios. The parameter values and connections to modules and scenarios can be set when you open the domain object window from within the scenario, but the other specifications may not be altered.

The **Processes** tab lists the MIMS processes (and associated modules) for the domain object. After a module instance has been set for a process, double-click on that process to call up a Module Instance window for the process.

The actions of the menu options, screen buttons (▭), and icons (e.g., +) are described below:

Identifier	Function	Window Access	Menu Access
Add New Parameter	Adds a new parameter to the list, one that the user defines. The feature only works for domain object windows opened from the project window.	+	Parameters
Add Parameter Values	Copies parameters from the contained module into the domain object	▭	
Add Process	Adds a new MIMS Process to the domain object	▭	
Add to Scenario	Adds the highlighted parameter to the Scenario Parameters Window. See how parameters are associated.	⊞	Parameters
Check Connectivity to Parameter	Compares the interchangeability of the selected parameters. A reason is displayed when two parameters cannot be connected.	?	Parameters
Clear Module Instance	Removes all parameter values and modules from the highlighted MIMS process	▭	
Clear Parameter	Erases the value assigned to the highlighted parameter	✎	Parameters
Close Domain Object	Closes the domain object window but not necessarily the MIMS framework		File
Condense/ Expand View	Displays the expanded view (showing all parameters) or the condensed view (showing just parameter groups and individual parameters)	↓↑ ↑↓	Parameters
Connect to Parameter	Establishes a call to retrieve a parameter value from another MIMS domain object or MIMS framework object. See how parameters are associated.	-■■-	Parameters
Copy	Duplicates a parameter found in the domain object window (and allows it to be placed in another domain object window or framework object window by the **Paste** command)	📋	Edit

Identifier	Function	Window Access	Menu Access
Copy from Module	Brings up a list of modules and imports the parameters of the selected modules into the parameter list. See how parameters are associated.	⊛	Parameters
Copy Top Value to Selected Parameter Values	Copies the first selected parameter value to all other selected parameters of the same parameter type	⇩	Parameters
Cut	Erases a parameter from the domain object window (and allows it to be placed in another domain object window or framework object window by the **Paste** command)	✂	Edit
Delete Parameter	Deletes the selected parameters	✕	Parameters
Edit Description	Allows the you to describe the domain object. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Edit
Edit Description	Allows you to describe the highlighted process. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	▭	
Edit IsRequired Formula	If the parameter is only to be used under certain conditions, the IsRequired formula is set to specify those conditions logically.	*Req	Parameters
Edit Module Instance	Opens a Module Instance Window for the highlighted process	▭	
Edit Parameter Value Description	Allows you to describe the highlighted parameter and the source of its assigned value. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	▤	Parameters
Edit Parameter Type	Edits the description of the highlighted parameter. In some cases, additional features of the parameter type can be altered.	✎	Parameters
Evaluate Parameter	Displays the current value for the highlighted parameter	Right click	

Identifier	Function	Window Access	Menu Access
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework		File
Import Parameter Values	Prompts you to select a file containing the parameter values		File
Open Project...	Opens an existing project and the associated project window	Ctrl-O	File
Paste Parameter	Inserts a parameter that has been cut or copied into the module instance window	📋	Edit
Paste Value	Inserts a parameter value that has been cut or copied into the highlighted parameter	📋	Edit
Print	Sends the parameter table to the printer	🖨	Parameters
Project Selection Window	Brings the list of projects into the foreground		File
Redo	Repeats the previous action	↻	Edit
Remove Processes	Disconnects the selected processes from the domain object	□	
Reorder parameter	Moves the selected parameter one place higher in the list	⇅	Parameters
Reorder Process	Shifts the highlighted process up one place in the list	□	
Save Domain Object	Saves the domain object		File
Set From File	Generates a set of M3IO file variable type parameters by reading the specified file. The m3io and ioapi formats are discussed further at http://www.cep.unc.edu/empd/EDSS/ioapi/.	📂	Parameters
Set Icon	Assign an image to the particular domain object		Edit
Set Module Instance	Attaches a module to the highlighted process	□	

Identifier	Function	Window Access	Menu Access
Set Parameter From Project	Assigns the value from this parameter at the project level to the value of this parameter	↑	Parameters
Undo	Cancels the previous action	↶	Edit
View File	Opens a browser showing the file contents for the selected parameter	Right click	

Framework Object Window

The Framework Object window (titled with the framework object name) lets you view the MIMS Processes and Parameters associated with the MIMS Framework Object. Framework objects include:

- Iterator Framework Object
- Simple Monte Carlo Iterator
- Simple Monte Carlo Iterator (max)
- Synchronized List Iterator

Use the *Type* field to describe the modules and parameters associated with this framework object. Click the tabs to view a list of either the parameter values or processes.

The **Parameters Values** tab lists the parameters of the framework object and their origins. The list contains the following information:

- Parameter name
- Value (output value or file name where information is stored)
- Source (MIMS Module that creates the parameter)
- Status (in, out, inout, required, used, local, *, and locked)

The specifications about framework object type, the framework object parameters, and the values can be set from the framework object window opened from the project window, but parameter connections cannot be established to modules or scenarios. The parameter values and connections to modules and scenarios can be set when you open the framework object window from within the scenario, but the other specifications may not be altered.

The **Processes** tab lists the MIMS processes (and associated modules) for the framework object. Double-click on a process to call up a Module Instance window for the process.

The actions of the menu options, screen buttons (▭), and icons (e.g., +)are described below:

Identifier	Function	Window Access	Menu Access

Identifier	Function	Window Access	Menu Access
Add New Parameter	Adds a new parameter to the list, one that you define. The feature only works for framework object windows opened from the project window.	+	Parameters
Add Parameter Values	Copies parameters from the contained module into the framework object.	▭	
Add Process	Adds a new MIMS Process to the framework object.	▭	
Add to Scenario	Adds the highlighted parameter to the Scenario Parameters Window. See how parameters are associated.	⊞	Parameters
Check Connectivity to Parameter	Compares the interchangeability of the selected parameters. A reason is displayed when two parameters cannot be connected.	?	Parameters
Clear Module Instance	Removes all parameter values and modules from the highlighted MIMS process.	▭	
Clear Parameter	Erases the value assigned to the highlighted parameter.	✎	Parameters
Close [framework object type]	Closes the framework object window but not necessarily the MIMS framework.		File
Condense/ Expand View	Displays the expanded view (showing all parameters) or the condensed view (showing just parameter groups and individual parameters).	⤓ ⇕	Parameters
Connect to Parameter	Establishes a call to retrieve a parameter value from another MIMS domain object or MIMS framework object. See how parameters are associated.	●●	Parameters
Copy	Duplicates a parameter found in the framework object window (and allows it to be placed in another domain object window or framework object window by the **Paste** command)	📋	Edit

Identifier	Function	Window Access	Menu Access
Copy from Module	Brings up a list of modules and imports the parameters of the selected modules into the parameter list. See how parameters are associated.	⊕	Parameters
Copy Top Value to Selected Parameter Values	Copies the first selected parameter value to all other selected parameters of the same parameter type	⇩	Parameters
Cut	Erases a parameter from the framework object window (and allows it to be placed in another domain object window or framework object window by the **Paste** command)	✂	Edit
Delete Parameter	Deletes the selected parameters	✗	Parameters
Edit Description	Allows you to describe the framework object. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Edit
Edit Description	Allows you to describe the highlighted process. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	▭	
Edit IsRequired Formula	If the parameter is only to be used under certain conditions, the IsRequired formula is set to specify those conditions logically.	Req	Parameters
Edit Module Instance	Opens a Module Instance window for the highlighted process.	▭	
Edit Parameter Value Description	Allows you to describe the highlighted parameter and the source of its assigned value. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	▤	Parameters
Edit Parameter Type	Edits the description of the highlighted parameter. In some cases, additional features of the parameter type can be altered.	✎	Parameters
Evaluate Parameter	Displays the current value for the highlighted parameter.	Right click	

Identifier	Function	Window Access	Menu Access
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework.		File
Import Parameter Values	Prompts you to select a file containing the parameter values.		File
Open Project...	Opens an existing project and the associated project window.	Ctrl-O	File
Paste Parameter	Inserts a parameter that has been cut or copied into the module instance window.	📋	Edit
Paste Value	Inserts a parameter value that has been cut or copied into the highlighted parameter.	📋	Edit
Print	Sends the parameter table to the printer.	🖨	Parameters
Project Selection Window	Brings the list of projects into the foreground.		File
Redo	Repeats the previous action.	↻	Edit
Remove Processes	Disconnects the selected processes from the framework object.	▭	
Reorder parameter	Moves the selected parameter one place higher in the list.	⇅	Parameters
Reorder Process	Shifts the highlighted process up one place in the list.	▭	
Save [framework object type]	Saves the framework object.		File
Set From File	Generates a set of M3IO file variable type parameters by reading the specified file. The m3io and ioapi formats are discussed further at http://www.cep.unc.edu/empd/EDSS/ioapi/.	📂	Parameters
Set Icon	Assign an image to the particular framework object.		Edit

Identifier	Function	Window Access	Menu Access
Set Module Instance	Attaches a module to the highlighted process.	▭	
Set Parameter From Project	Assigns the value from this parameter at the project level to the value of this parameter.	↑	Parameters
Undo	Cancels the previous action.	↶	Edit
View File	Opens a browser showing the file contents for the selected parameter.	Right click	

Modules

Module Window

The Module window (titled "Module"+module name) lists the MIMS Processes, Parameters, and the exterior program associated with the MIMS Module. In addition to the parameters list, the module window presents:

- Compatible Domain Object Type (the domain object type that may contain the module, not the name of the domain object)
- Name of Process This Implements (MIMS Process associated with the module)

In addition, many modules also include preprocessing instructions, execution commands, control files, custom GUI objects, and/or columnar files. The execution command represents the path and name for the software program as well as any command line arguments. Control files send messages to the external programs and may be edited from a pop-up dialog box that the module window can generate. You may type directly into these fields or select **Browse** when necessary. Check the *If no output, do not wait for completion* check box to control processing. The *Perform Preprocess* option allows preliminary execution of arbitrary script code (e.g., to perform consistency checks among parameters). Not all of these options are available for Iterate modules.

The Parameters that are input and output from the MIMS module are also listed in this window. The list contains the following information:

- Object Type (MIMS Domain Object or Framework Object that generates the parameter)
- Parameter (lists the name of the parameter)
- Internal name (parameter name within the module)
- Value (parameter value or file name where information is stored)
- Additional Type Info (units, file locations, time zones, field delimiters, labels, etc.)
- Status (in, out, inout, required, used, local, *, and locked)

If opened from a Scenario Window, the parameters cannot be added, altered, or removed from the module window. They may be adjusted for some modules (e.g., default external modules) only when the module window is opened from the Project Window.

The actions of the menu options and icons (e.g., +) are described below:

Identifier	Function	Window Access	Menu Access
Add New Parameter	Adds a new parameter to the list, one that you define.	+	Parameters
Clear Parameter	Erases the value assigned to the highlighted parameter.	✎	Parameters
Close [module type]	Closes the module window but not necessarily the MIMS framework.		File
Condense/ Expand View	Displays the expanded view (showing all parameters) or the condensed view (showing just parameter groups and individual parameters).	⇊ ⇕	Parameters
Copy	Duplicates a parameter found in the domain object window (and allows it to be placed in another domain object window or framework object window by the **Paste** command).	📋	Edit
Copy Top Value to Selected Parameter Values	Copies the first selected parameter value to all other selected parameters of the same parameter type.	⤓	Parameters
Cut	Erases a parameter from the module window (and allows it to be placed in another member by the **Paste** command).	✂	Edit
Delete Parameter	Deletes the selected parameters.	✕	Parameters
Edit Description	Allows you to describe the module. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Edit
Edit IsRequired Formula	If the parameter is only to be used under certain conditions, the IsRequired formula is set to specify those conditions logically.	Req	Parameters
Edit Parameter Value Description	Allows you to describe the highlighted parameter and the source of its assigned value. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	▤	Parameters

Identifier	Function	Window Access	Menu Access
Edit Parameter Type	Edits the units and description of the highlighted parameter. In some cases, additional features of the parameter type can be altered. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	✎	Parameters
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework.		File
Open Project...	Opens an existing project and the associated project window.	Ctrl-O	File
Paste Parameter	Inserts a parameter that has been cut or copied into the module window.	📋	Edit
Paste Value	Inserts a parameter value that has been cut or copied from another parameter into the value for the highlighted parameter.	📋	Edit
Print	Sends the parameter table to the printer.	🖨	Parameters
Project Selection Window	Brings the list of projects into the foreground.		File
Redo	Repeats the previous action.	↻	Edit
Reorder parameter	Moves the selected parameter one place higher in the list.	⇅	Parameters
Set from File	Generates a set of M3IO file variable type parameters by reading the specified file. The m3io and ioapi formats are discussed further at http://www.cep.unc.edu/empd/EDSS/ioapi/.	📂	Parameters
Set Parameter from Project	Assigns the value from this parameter at the project level to the value of this parameter.	↑	Parameters
Undo	Cancels the previous action.	↺	Edit

Module Instance Window

The Module Instance window (titled "Module Instance"+module name) lists the parameters associated with a MIMS Module. The **Module** tab shows the same module description lines seen on the Module window (e.g., execution Command, Control Files, Compatible Domain Object Type, Process Name, Description, and instructions to wait for completion). The parameters that are input and output during the execution of the Module Instance are listed on the **Parameters** tab of this window. The list contains the following information:

- Object Type (MIMS Domain Object that generates the parameter)
- Parameter (lists the name of the parameter)
- Value (parameter value or file name where information is stored)
- Additional Type Info (units, file locations, time zones, field delimiters, labels, etc.)
- Source (if the parameter is read from a domain or framework object output, this field indicates the object name)
- Status (in, out, inout, required, used, local, *, and locked)

If the module instance window is opened from the Scenario window, parameters cannot be added, altered, or removed from the module instance window. However, you can change parameters in module windows that were accessed through the Project window.

The actions of the menu options and icons (e.g., +) are described below (if some are grayed out and not available in the module instance window, use the module window instead):

Identifier	Function	Window Access	Menu Access
Add New Parameter	Adds a new parameter to the list, one that the user defines	+	Parameters
Add to Scenario	Adds the highlighted parameter to the Scenario Parameters window. See how parameters are associated.	⊞	Parameters
Check Connectivity to Parameter	Compares the interchangeability of the selected parameters. A reason is displayed when two parameters cannot be connected.	?	Parameters

Identifier	Function	Window Access	Menu Access
Clear Parameter	Erases the value assigned to the highlighted parameter	✎	Parameters
Close [module type]	Closes the module instance window but not necessarily the MIMS framework		File
Condense/ Expand View	Displays the expanded view (showing all parameters) or the condensed view (showing just parameter groups and individual parameters)	⤒ ↕	Parameters
Connect to Parameter	Establishes a call to retrieve a parameter value from a MIMS domain object or framework object. See how parameters are associated.	⊶	Parameters
Copy	Duplicates a parameter found in the domain object window (and allows it to be placed in another domain object window or framework object window by the **Paste** command)	📋	Edit
Copy Parameter from Object	Adds a new parameter to the list, one created in a domain object window or framework object window	＋	Parameters
Copy Top Value to Selected Parameter Values	Copies the first selected parameter value to all other selected parameters of the same parameter type	⤓	Parameters
Cut	Erases a parameter from the module instance window (and allows it to be placed in another member by the **Paste** command)	✂	Edit
Delete Parameter	Deletes the selected parameters	✕	Parameters
Edit Description	Allows you to describe the module instance. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Edit
Edit IsRequired Formula	If the parameter is only to be used under certain conditions, the IsRequired formula is set to specify those conditions logically.	Req*	Parameters

Identifier	Function	Window Access	Menu Access
Edit Parameter Value Description	Allows you to describe the highlighted parameter and the source of its assigned value. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	🗎	Parameters
Edit Parameter Type	Edits the units and description of the highlighted parameter. In some cases, additional features of the parameter type can be altered. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	✎	Parameters
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework		File
Open Project...	Opens an existing project and the associated project window	Ctrl-O	File
Paste Parameter	Inserts a parameter that has been cut or copied into the module instance window	📋	Edit
Paste Value	Inserts a parameter value that has been cut or copied from another parameter into the value for the highlighted parameter	📋	Edit
Print	Sends the parameter table to the printer	🖨	Parameters
Project Selection Window	Brings the list of projects into the foreground		File
Redo	Repeats the previous action	↻	Edit
Reorder Parameter	Moves the selected parameter one place higher in the list	⇅	Parameters
Set from File	Generates a set of M3IO file variable type parameters by reading the specified file. The m3io and ioapi formats are discussed further at http://www.cep.unc.edu/empd/EDSS/ioapi/.	📂	Parameters
Set Parameter from Project	Assigns the value from this parameter at the project level to the value of this parameter	⇡	Parameters

Identifier	Function	Window Access	Menu Access
Show Module	Opens a Module Window for the module associated with this module instance. Parameters cannot be added or removed in the displayed module window.	□	Parameters
Undo	Cancels the previous action	↶	Edit
View Parameter Value Description	Allows you to describe the value assigned to the highlighted parameter	▤	Parameters

Synchronized Input Lists Editor

The purpose of the Synchronized Input Lists Editor window is to allow you to match and order the parameter values that will be used in the iterations of an iterator framework object. The left panel of the window shows the input parameter values associated with one iteration, and the right panel shows the input parameter values for all of the iterations. The values may only be edited in the left panel.

The left panel features three columns for the parameter names, for the values, and for **Set** or **Edit** buttons to describe complex parameter divisions (e.g., execution method). The right panel displays the synchronized list, showing one column with all of the parameter values for each iteration.

Initially no items are contained in the iteration list. Click on the **Add New Iteration** (⊕) icon to create the first entry. A dialog box prompt s you for a name to the entry (e.g., "baseline case"). Click **OK** to create the first iteration, then use the blanks in the left panel to assign values to the parameters.

The actions of the screen buttons are described below:

Identifier	**Function**	**Button**
Add new iteration	Creates the next iteration in the synchronized input list	⊕
Copy to clipboard	Copy all parameter values from the highlighted iteration to the clipboard	📋
Import table	Retrieves a previously created synchronized input list	📂
Move left	Change the left panel display to show the values of the previous iteration	←
Move right	Change the left panel display to show the values of the next iteration	→
Paste from clipboard	Copy all parameter values from the clipboard to the highlighted iteration	📋
Print table	Sends synchronized input list to printer	🖨

Identifier	Function	Button
Remove iteration	Deletes the highlighted iteration from the synchronized input list	⊖
Reorder iteration to left	Switch the positions of the highlighted iteration and the one to the left	←
Reorder iteration to right	Switch the positions of the highlighted iteration and the one to the right	→
Save table to file	Saves the synchronized input list to a file	💾

Parameters

Calling Parameters

The MIMS Framework uses the Python code format for specifying parameters that should be passed between objects, modules, and scenarios. Under this format, specify that a parameter should be called by including the parameter name in single or double quotation marks. For example, if the **numeric** parameter *nlayers* has the number 12 associated with it, you would type "nlayers" or 'nlayers' as the parameter value in the module window's parameter table. When the module needs to know the number of layers (nlayers), the formula "nlayers" will be evaluated as 12.

Similarly, for a **Boolean** parameter called *test*, the expression "test" would evaluate to be either true or false. By right-clicking on the expression in the parameters table, you can evaluate the value currently assigned to a parameter.

Note that the Python code format is case-sensitive, so "nlayers" and "Nlayers" would be treated as two distinct parameters.

For **string**, **date/time**, and **environment** variables, you must specify formulas to read the parameters. A formula is denoted by a "*${* " on the left and "*}*" on the right. If *pstring* is a variable with value "abcdef", then you would call this parameter with the expression `${str("pstring")}`. MIMS also uses the following formulas to evaluate strings:

String Parameter	Evaluation
${str("pstring")}	abcdef
${str("pstring") [2:3]}	cde (2 specifies the character to begin the substring after and 3 shows the number of characters in the substring)
${str("pstring").upper()}	ABCDEF (changes strings to all upper case characters)
${str("pstring").lower()}	abcdef (changes strings to all lower case characters)
${str("pstring")+str("pstring") [2:3]}	abcdefcde (the + sign concatenates strings and substrings)

Similarly, a date/time variable and its components can be evaluated with formulas. The table below shows evaluations of the *pdate* variable (set to 07/14/2003 10:50:06):

Date/Time Parameter	Evaluation
${getDate("pdate","yyyymmdd")}	20030714
${getDate("pdate","mmddyyyy")}	07142003
${getDay("pdate")}	14
${getJDay("pdate")}	195 (Julian date in the calendar year)
${getYear("pdate")}	2003 (returns the 4-digit year)
${getYear2("pdate")}	03 (returns the 2-digit year)
${getHour("pdate")}	10
${getMinute("pdate")}	50
${getSeconds("pdate")}	06

Environment variables (such as the operating system *OS*) can be called with a simple formula like `${getEnv("OS")}`.

For variables associated with the List of Parameters parameter type, see the List of Parameters description.

MIMS also allows you to specify a parameter from two choices with ***ifElse*** formulas. The formula `${ifElse("test", getYear("pdate"), getYear2("pdate")}` would return 2003 if the *test* variable is true and 03 if the *test* variable is false. The ifElse conditional formulas are similar to those used for IsRequired formulas.

To understand more about constructing formulas to operate in the MIMS framework, visit the Python documentation web site.

Scenario Parameters Window

The Scenario Parameters window (titled "Parameter Values of Scenario"+scenario name) lists the Parameters associated with a MIMS Scenario. The parameters listed here, such as Default Directory, apply to all of the members of the scenario. By default, the scenario parameters window shows the default directory, execution path, and whether or not old output files should be deleted before execution of the scenario's processes. The list contains the following information:

- Parameter (lists the name of the parameter)
- Value (parameter value or file name where information is stored)
- Source (if the parameter is read from a domain or framework object, this field indicates the object name)
- Status (in, out, inout, required, used, local, *, and locked)

MIMS output files and error logs are directed to the default directory. If no default directory is specified, the output files and error logs are stored in the directory from where MIMS was launched.

The actions of the menu options and icons (e.g., +) are described below:

Identifier	Function	Window Access	Menu Access
Add New Parameter	Adds a new parameter to the list, one that you define.	+	Parameters
Add Parameter from Project	Adds a new parameter to the list, one created in the Project Window.	+	Parameters
Clear Parameter	Erases the value assigned to the highlighted parameter.	✎	Parameters
Close Parameter Values Table	Closes this window but not necessarily the MIMS framework.		File

Identifier	Function	Window Access	Menu Access
Condense/ Expand View	Displays the expanded view (showing all parameters) or the condensed view (showing just parameter groups and individual parameters).	⇕ ⇕	Parameters
Copy	Duplicates a parameter found in the scenario parameters window (and allows it to be placed in parameters tables by the **Paste** command).	📋	Edit
Copy Top Value to Selected Parameter Values	Copies the first selected parameter value to all other selected parameters of the same parameter type.	↓	Parameters
Cut	Erases a parameter found in the scenario parameters window (and allows it to be placed in parameters tables by the **Paste** command).	✂	Edit
Delete Parameter	Deletes the selected parameters.	✗	Edit
Edit IsRequired Formula	If the parameter is only to be used under certain conditions, the IsRequired formula is set to specify those conditions logically.	Req	Parameters
Edit Parameter Value Description	Edits the description for the highlighted parameter. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	▤	Parameters
Edit Parameter Type	Edits the description for the highlighted parameter. In some cases, additional features of the parameter type can be altered. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	✎	Parameters
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework.		File
Open Project…	Opens an existing project and the associated project window.	Ctrl-O	File
Paste	Inserts a parameter that has been cut or copied into the parameters list.	📋	Edit
Print	Sends the parameters table to the printer.	🖨	Parameters

Identifier	Function	Window Access	Menu Access
Project Selection Window	Brings the list of projects into the foreground.		File
Redo	Repeats the previous action.	↻ Ctrl-Y	Edit
Reorder Parameter	Moves the selected parameter one place higher in the list.	↑	Parameters
Set from File	Generates a set of M3IO file variable type parameters by reading the specified file. The m3io and ioapi formats are discussed further at http://www.cep.unc.edu/empd/EDSS/ioapi/.	📂	Parameters
Set Parameter from Project	Copies a parameter value from a project.	↑	Parameters
Undo	Cancels the previous action.	↺ Ctrl-Z	Edit

Parameter Group Window

If you choose to employ a Parameter Group to define a whole set of Parameters under a single name, then click the **Edit** button in the row beside the parameter name. This opens up a window titled "Parameter Group of "+<parameter group name> with a list of the parameters contained within the group.

The list contains the following information:

- Object Type (the MIMS domain object or framework object type that may contain the parameter)
- Parameter (lists the name of the parameter)
- Internal Name (parameter name within the module)
- Value (parameter value or file name where information is stored)
- Status (in, out, inout, required, used, local, *, and locked)

The actions of the menu options and screen buttons are described below:

Identifier	Function	Window Accesss	Menu Access
Add New Parameter	Adds a new parameter to the list, one that you define.	+	Parameters
Add Parameter from Project	Adds a new parameter to the list, one created in the Project Window.	⊞	Parameters
Add to Group	Pulls individual parameters for the module, object, or scenario that have already been defined into the parameter group.	⊕	Parameters
Clear Parameter	Erases the value assigned to the highlighted parameter.	✎	Parameters
Close Parameter Group	Closes this window but not necessarily the MIMS framework.		File

Identifier	Function	Window Accesss	Menu Access
Condense/ Expand View	Displays the expanded view (showing all parameters) or the condensed view (showing just parameter groups and individual parameters).	↧ ↥	Parameters
Copy Top Value to Selected Parameter Values	Copies the first selected parameter value to all other selected parameters of the same parameter type.	⇩	Parameters
Delete Parameter	Deletes the selected parameters.	✕	Edit
Edit: Description	Allows you to describe the Parameter Group. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.		Edit
Edit: IsRequired Formula	If the parameter is only to be used under certain conditions, the IsRequired formula is set to specify those conditions logically.	Req	Parameters
Edit: Parameter Value Description	Edits the description for the highlighted parameter. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	▤	Parameters
Edit: Parameter Type	Edits the description for the highlighted parameter. In some cases, additional features of the parameter type can be altered. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	✎	Parameters
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework.		File
Open Project…	Opens an existing project and the associated project window.	Ctrl-O	File
Print	Sends the parameters table to the printer.	🖨	Parameters
Project Selection Window	Brings the list of projects into the foreground.		File

Identifier	Function	Window Accesss	Menu Access
Redo	Repeats the previous action.	↻ Ctrl-Y	Edit
Remove from Group	Moves the selected group parameter out of the group and up to individual parameter status.	⊖	Parameters
Reorder Parameter	Moves the selected parameter one place higher in the list.		Parameters
Set from File	Generates a set of M3IO file variable type parameters by reading the specified file. The m3io and ioapi formats are discussed further at http://www.cep.unc.edu/empd/EDSS/ioapi/.	📂	Parameters
Set Parameter from Project	Copies a parameter value from a project.		Parameters
Undo	Cancels the previous action.	↺ Ctrl-Z	Edit

Chemical Mechanism Family Editor

This window describes the available Chemical Mechanisms available within a particular Chemical Mechanism Family. The chemical mechanism family is a parameter associated with MIMS Modules.

A list shows some key characteristics of the chemical mechanisms within the family: whether or not aerosol chemistry is included, whether or not aqueous phase chemistry is included, and whether or not the mechanisms can be altered (ReadOnly). The first two characteristics are important for atmospheric chemistry. If the Aerosol or Aqueous boxes are checked, it indicates that at least one chemical species in the mechanism is associated with the aerosol or aqueous phases. Turning off these check boxes will ignore the reactions/conversions associated with the Aerosol and Aqueous phases. If no associations exist with these phases within the chemical mechanism, the box cannot be checked.

One chemical mechanism name listed in the window will appear in black, while all others appear in gray. The black name indicates the Active Chemical Mechanism, the one that will currently be used in Module Instances. To select a different Active Chemical Mechanism, click on the new mechanism and then on the **Set Active** button.

This window also contains menu options and buttons (▭) to perform the following tasks:

Identifier	Function	Window Access	Menu Access
Add Mechanism	Adds a new chemical mechanism to the list, starting with a copy of the Active Chemical Mechanism.	▭	Edit
Close Chemical Mechanism	Closes this window but not necessarily the MIMS framework.		File
Delete Mechanism	Permanently erases the selected chemical mechanism.	▭	Edit
Edit Description	Allows you to describe the selected chemical mechanism. Use Ctrl-C and Ctrl-V to cut and paste from other descriptions.	▭	Edit

Identifier	Function	Window Access	Menu Access
Edit Mechanism	Opens the *Edit Dialog for Mechanism* window for the selected chemical mechanism, allowing you to view and alter the reactions, reaction rates, and species considered.	□	Edit
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework.		File
Export Mechanism	Saves the selected chemical mechanism and all of its associated parameters to a file.		Edit
Import Mechanism	Opens a file to add new chemical mechanisms to the current chemical mechanism family.	□	Edit
Open Project...	Opens an existing Project and the associated Project window.	Ctrl-O	File
Project Selection Window	Brings the list of Projects into the foreground.		File
Set Active	Marks the selected chemical mechanism in black and uses that one in subsequent module instances.	□	Edit

Edit Dialog for Mechanism

When you press the **Edit** button on the Chemical Mechanism Family Editor, the *Edit Dialog for Mechanism* window appears. Currently this window supports a standard format of menus aimed at air quality modules; you may click through a series of tabs to view five different lists:

- **Reactions** – list of all chemical reactions (and phase conversions) and their kinetic rate constants
- **Gas Species** – list of each chemical species (or grouped species) found in the gas phase, its known chemical and physical properties, and associated variables
- **Aerosol Species** – same as Gas Species list, for aerosol components (species must be listed here to have the Aerosol box checked on the Chemical Mechanism Family Editor
- **Non-Reactive Species** – same as Gas Species list, for non-reactive components
- **Tracer Species** – list of chemical species that will be tracked from the emissions source to the receptor location

To see what each of the fields stands for on these lists, see the List of Chemical Mechanism Parameters.

Most of this window is edited directly by double-clicking on the cell that you want to change. A few additional operations are available through the menus:

Identifier	Function	Menu Access
Add Surrogate	Adds a surrogate species and its conversion factor to allow tracking of measurable quantities.	Edit
Add Switch	Adds a switch to allow you to leave a species out of calculations.	Edit
Close with saving	Closes this window but not necessarily the MIMS framework; saves any changes made to the chemical mechanism unless the mechanism was marked ReadOnly in the Chemical Mechanism Family Editor.	File
Close without saving	Closes this window but not necessarily the MIMS framework; does not save any changes made to the chemical mechanism.	File

Identifier	Function	Menu Access
Delete Row(s)	Permanently removes the selected rows from the list.	Edit
Delete Surrogate	Removes the surrogate species.	Edit
Delete Switch	Removes the switch to allow you to leave a species out of calculations.	Edit
Insert Row	Adds a new row to the front table for new reactions or chemical species.	Edit
Print	Sends the front table list to the printer.	File

Grid Family Window

When you choose to view a Grid Family or create a new grid family from the Project window (**New** button followed by a choice of Grid Family in the Others folder), the *Grid Family GUI Window* appears. This window can be used to create grids, visualize how multiple grids overlay land areas, and print the associated maps. If you use nested grids, this type of interface allows you to produce and display the grid boundaries quickly.

If you choose to create a new grid family, the window initially shows a map on the right with North American states and provinces (except Hawaii) with longitude/latitude coordinate system. The window is divided into three sections: the Grid Chooser panel, the Grid Info panel, and the Grid Display panel. The Grid Chooser panel (upper left) allows you to specify the coordinate system. The Grid Info panel (lower left) lets you know the exact coordinates of the grid boundaries, and the Grid Display panel (right side) overlays the grids on familiar land areas (e.g., states, provinces, counties, and cities).

To create a nested grid within this interface,

1. Click the **Edit** button in the Grid Chooser panel. This opens the *Map Projection Editor* dialog box.
2. Select the projection family, ellipsoid, and appropriate values for the parameters. Then press **OK**.
3. Click the **Add New** button in the Grid Chooser panel and provide the name for the coarse grid.
4. In the Grid Info panel (lower left), choose whether to use the units of the projection or latitude/longitude when reading the lower left and upper right coordinates. Type in the lower left and upper right coordinates for the coarse grid. Then, type the numbers of cells in the x and y directions.
5. Click on the **Calc** button beside the *DxDy* field to create the coarse grid. The coarse grid is displayed in blue in the Grid Display panel.
6. Click the **Add New** button in the Grid Chooser panel and provide the name for the fine grid.
7. In the Grid Info panel (lower left), click the **Nest Grid** button to display the *Set Nested Grid Info* dialog box.
8. Choose the coarse grid as the parent grid, the Offsets to specify the number of coarse cells distant from the lower left corner, the number of fine cells to cover with the fine grid, and the Delta Ratios to specify the number of fine cells per coarse cell.

9. Then press **OK** on the *Nested Grid Info* dialog box. The fine grid is displayed in green in the Grid Display panel.
10. Select the **File/Export GRIDDESC** menu item to save a description of this grid for use with MIMS modules or the MIMS Spatial Allocator.

The actions of the menu options, screen buttons (▭), and icons (e.g., 🔍) are described below:

Identifier	**Function**	**Window Access**
Add New	Places a new grid into the current grid family.	▭
Calc (beside Lower Left)	Calculates a grid layer based on the information in the upper right, number of cells, and delta X and Y fields.	▭
Calc (beside Upper Right)	Calculates a grid layer based on the information in the lower left, number of cells, and delta X and Y fields.	▭
Calc (beside Num Cells)	Calculates a grid layer based on the information in the lower left, upper right, and delta X and Y fields.	▭
Calc (beside Delta X,Y)	Calculates a grid layer based on the information in the lower left, upper right, and number of cells fields.	▭
Close Grid Family	Closes this window but not necessarily the MIMS framework.	File menu
Delete	Permanently removes a grid layer.	▭
Duplicate	Creates a copy of the existing grid layer.	▭
Edit	Allows you to change the name of the grid family or the coordinate system.	▭
Edit Description	Adds a paragraph describing the selected grid layer.	▭
Edit Layers	Allows you to determine which ESRI shape files are shown with the grids, add new layers, and choose the order in which the layers are displayed.	Edit menu/ Layers ▭

Identifier	Function	Window Access
Exit MIMS Framework	Closes all windows and programs associated with the MIMS framework.	File menu
Export GRIDDESC	Saves an existing grid family to a file location.	File menu
Import GRIDDESC	Loads an existing grid family from a file.	File menu
Nest Grid	Subdivides a portion of one grid into smaller subunits.	▭
Open Overlay Files	Opens a file that can display polygons, lines, and text in the Grid Display panel.	File menu
Print	Prints the map displayed in the Grid Display panel.	▭
Redo	Repeats the previous action.	Edit menu/ Ctrl-Y
Rename	Changes the name of a grid layer.	▭
ROI	Lets you directly enter the minimum and maximum latitudes and longitudes for the region of interest (ROI) that will be shown in the display.	▭
Set LL&UR	Lets you drag the mouse to set the lower left and upper right coordinates of the Grid Info panel.	▭
Show Grid	Determines whether or not the grid layer is shown in the Grid Display panel.	Check box
Thin Lines By	The width of thicker grid lines can be thinned with this command or the Layers options.	Field Entry
Undo	Cancel the previous action.	Edit menu/ Ctrl-Z
XY/LL	Determines if the X/Y or latitude/longitude coordinates for the mouse are displayed in the Grid Display panel.	Check boxes
Zoom Area	Allows you to drag the mouse across the desired zoom area.	▭

Identifier	Function	Window Access
Zoom In	Changes the zoom on the Grid Display panel to double the magnification.	🔍+
Zoom Out	Changes the zoom on the Grid Display panel to halve the magnification.	🔍−

Parameter Types and Values

The MIMS Framework allows you to specify the parameter types listed below. When you create a scenario, domain object, or module, new parameters may be added. To help you choose the appropriate types, the Parameter Types and Values chapter presents the following parameter divisions:

- Single numbers (with parameter types Integer, Long Integer, Floating Point, Date, and Distribution)
- Numeric series (Floating Point Series and Floating Point Time Series)
- Strings (String and String Choice)
- Logical (True or False) - Boolean type
- File access (Directory, File Name, Typed File, and M3IO Variable)
- Execution method (local, Condor, SSH)
- Specialty (Duration, Parameter Group, Grid Family, Chemical Mechanism)

Within these parameter divisions, you may choose the data format by selecting the parameter type and assign the representations by specifying parameter values. After you specify the parameter type, the number of available parameter values for that parameter will be fixed. Predefined parameter values can be created directly within MIMS.

The parameter type is chosen when the parameter is created and cannot be altered. Parameters are associated with a particular MIMS member and may be viewed and set from the member's window. Within a MIMS member (e.g., scenario, domain object, or module), each parameter must have a unique name. The output parameter from one MIMS module instance or other member may serve as the input for another member through parameter connections.

Parameter Divisions

Parameter Division: Single Number

You may choose from several parameter types when interested in specifying a single number:

- **Integers** are whole numbers between -2,147,483,647 and 2,147,483,647 (2E31-1), inclusive.
- **Long Integers** are whole numbers between -2E63-1 and 2E63-1, inclusive, and may be specified by following the number with "L". The storage space requirements for long integers are twice that for integers.
- **Floating Point** parameters represent any rational numbers, and you may type these values directly into MIMS or use a parameter connection.
- **Floating Point (Generic)** parameters represent any rational numbers and may be filled by floating point constant values or floating point from file values.
- **Date** parameters represent the year, month, day, and optionally the time (in hours, minutes, and seconds).
- **Distribution** parameters allow the program to choose a single random value from a set distribution. The system currently supports Uniform Distribution Values and Normal Distribution Values.

Parameter Division: Numeric Series

You may choose from the following parameter types when interested in specifying a numeric series:

- **Floating Point Series (Generic)** parameter types represent an ordered list of numbers. This parameter type may be filled with the Double Series in Memory Values.
- **Floating Point Time Series (Generic)** parameter types represent a paired list of times and values. This parameter type may be filled with CASTNET Time Series Values, Time Series Constant Values, Time Series from File Values, Time Series from User Defined Function, or Time Series in Memory Values.

Parameter Division: Strings

You may choose from the following parameter types when interested in specifying strings:

- **String parameters** represent text that you may type directly. If you choose to create a string parameter, he/she must indicate the maximum length of the string.
- **String Choice parameters** create parameters that allow specific values to be chosen from a pulldown menu. When creating a string choice parameter, you are presented with the *Edit String Choice Parameter Type* dialog box. This dialog box prompts you to develop the pulldown menu with the following fields:

  Description - Type what the parameter should be used to describe.

  View Sorted List During Selection - Alphabetize the pulldown menu.

  Editable Combo Box - Type a string, in addition to choosing a value from the pulldown menu.

  Choices - Use the **Insert Row** and **Delete Row** buttons to develop a list of choices for the pulldown menu.

Parameter Division: File Access

You may choose from the following parameter types when interested in specifying variables that deal with file access:

- **Directory** parameter type returns a computer directory path, including the drive designation.
- **File Name** parameter type returns a file name, along with the computer directory path, including the drive designation.
- **Typed File** parameter type treats the specified file as it would any file with the specified extension. For example, a typed file parameter with a type of "txt" is treated as a text file. When you try to view a file (right click on the parameter value) with the "txt" extension, the default viewer is used. If a typed file parameter has the "txt" type, then it is also viewed with the default viewer.
- **M3IO File Variable** parameter type returns the specified variable from the selected file. The selected file is the value associated with this parameter, and the M3IO file variable is typed when the you create this parameter type on the **Edit** dialog box. The m3io and ioapi formats are discussed further at http://www.cep.unc.edu/empd/EDSS/ioapi/.

An alternate way to access data from files is to use the **Floating Point from File** parameter value.

Parameter Division: Execution Method

You may only choose the **Execution Method** parameter type within this parameter division. Parameters of this type specify where and how executable programs should be run.

MIMS currently allows you to define the following values for Execution Method parameters:

- Local Execution - run programs on the local machine
- Condor Execution - run programs on a remote machine that the Condor machine selects based on the queue
- Secure Shell Execution - run programs on a remote machine that you select

Parameter Division: Specialty

The Specialty parameter division is reserved for non-standard parameters that you, as a MIMS user may find useful when constructing projects:

- **Chemical Mechanism** - represents a group of reactions, reaction rates, and other descriptors (e.g., reaction phase) that can be selected through the Chemical Mechanism Family Editor and edited through the Edit Dialog for Mechanism dialog box.
- **Duration** - represents a time period and is expressed as a floating point number and a unit of time. Choosing new units changes the numbers appropriately (e.g., changing the hours to minutes multiplies the value by 60).
- **Grid Family** - represents a group of evenly gridded model domains and the projection created by the Grid Family GUI window.
- **Regular Grid** - represents a single gridded model domain within a grid family. You must specify both the grid family and then the grid name. This parameter cannot be set until at least one grid family is created within or imported into the project.
- **Group Parameter** - represents a way to collect several similar parameters (e.g., File Management Parameters) into a single group to make it easier to track many parameters. More details are available in the description of the Parameter Group Window.

List of Parameters - represents a list of values that may be substituted in as a vector of information, all values being of the same parameter type. If you are creating a List of Parameters parameter type, then you must decide on the type of parameters within the list and whether or not each row in the list should have a name.

Some modules will use the entire list of parameters, but others will only need particular information. To access information within a list of parameters, the following Python formulae may be used:

- *${val2(NameOfList, index)}* will retrieve a single value from NameOfList where the index corresponds to the row number.
- *${expand(NameOfList, "title", onOneLine)}* will retrieve all values in the list as "title = x1, x2, x3, x4, ..." where x1, x2, x3, and x4 refer to the ordered values in the list.

Parameter Values

CASTNET Time Series Value

The Clean Air Status and Trends Network (CASTNET) monitors ambient air quality, meteorology, and acid deposition rates. Based on the measurements, the CASTNET program builds files reporting hourly, daily, and weekly values from the monitoring efforts. MIMS has created a CASTNET Time Series Value to describe the meteorological conditions and ground-level ozone concentrations found in the hourly meteorological CASTNET files.

To create a CASTNET Time Series Value within the MIMS framework:

1. Click the **New** button in the Project Window. Choose CASTNET Time Series Value within the Parameter Values folder and type a name for the variable.
2. Click **OK**. The *Edit CASTNET Time Series Value* dialog box appears.
3. In the *Edit CASTNET Time Series Value* dialog box, type the directory and file name for the CASTNET file (meteorological files only) that should be read.
4. Under the **Value** menu, choose the variable contained in the CASTNET file that should be used. The following variables are contained in the CASTNET files:

   - temperature (C)
   - change in temperature between 1 and 10 m probes (C)
   - relative humidity (%),
   - solar radiation (watt/m^2)
   - ozone concentrations (ppb)
   - precipitation (mm/hr)
   - wind speed (m/s)
   - wind direction (degrees)
   - standard deviation of wind direction (sigma theta)
   - sample flow rate (l/min)
   - scalar wind speed (m/s)
   - wetness (%)
   - date and time of last record update

   Type a description of the file below that.

5. Click **OK** to save the information and create the parameter value.

Condor Execution

According to http://www.cs.wisc.edu/condor/description.html, "Condor is a specialized workload management system for compute-intensive jobs. Like other full-featured batch systems, Condor provides a job queueing mechanism, scheduling policy, priority scheme, resource monitoring, and resource management. Users submit their serial or parallel jobs to Condor, Condor places them into a queue, chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion."

Within MIMS, you may create a Condor Execution parameter value through the following steps:

1. Click the **New** button in the Project Window. Choose Condor Execution within the Parameter Values folder and type a name for the variable.
2. Click **OK**. The *Condor Execution Editor* dialog box appears.
3. In the *Condor Execution Editor* dialog box, enter the Condor Universe (Vanilla, Standard, or MPI), the Target Operating System, whether or not to move data files to/from the remote computer, whether or not to copy the execution to the remote computer, and any additional condor commands.
4. Click **OK** to save the information, dismiss the dialog box, and create the parameter value.

The open source Condor software and documentation are available for free download at http://www.cs.wisc.edu/condor/.

Double Series in Memory Value

If you wish to create and store a series of values, the Double Series in Memory Value is the MIMS member to create. This parameter value returns a sequential list of values, each with double precision.

To create a Double Series in Memory Value within the MIMS framework:

1. Click the **New** button in the Project Window. Choose Double Series in Memory Value within the Parameter Values folder and type a name for the variable.
2. Click **OK**. The *Edit Double Series in Memory Value* dialog box appears.
3. In the *Edit Double Series in Memory Value* dialog box, enter the Units and a Description of the value.
4. To create the list of values, click the **Insert Row** button once for each entry. Then click the value row, and type the number.
5. Click **OK** to save the information, dismiss the dialog box, and create the Double Series in Memory Value.

Floating Point Constant Value

If you wishes to create a single value, the Floating Point Constant Value is the MIMS member to create. This parameter value returns a single constant value.

To create a Floating Point Constant Value within the MIMS framework:

1. Click the **New** button in the Project Window. Choose Floating Point Constant Value within the Parameter Values folder and type a name for the variable.
2. Click **OK**. The *Edit Floating Point Constant Value* dialog box appears.
3. In the *Edit Floating Point Constant Value* dialog box, enter the Constant Value and a Description of the value.
4. Click **OK** to save the information, dismiss the dialog box, and create the parameter value.

Floating Point from File

If you wish to access a value in a file where the value corresponds to a particular key word (e.g., in control files), the Floating Point from File Value is the MIMS member to create. This parameter value returns a floating point number from the line in the file that contains the key word.

To create a Floating Point from File Value within the MIMS framework:

1. Click the **New** button in the Project Window. Choose Floating Point from File Value within the Parameter Values folder and type a name for the variable.
2. Click **OK**. The *Edit Floating Point from File* dialog box appears.
3. In the *Edit Floating Point from File* dialog box, enter the Key Word file or click on **Browse** to search for the file.
4. Type the Key Word, whether or not to verify the units, and a Description of the value.
5. Click **OK** to save the information, dismiss the dialog box, and create the parameter value.

Local Execution Value

The Local Execution Value designates that operations should be conducted on the local machine. The local execution value (and remote execution values) are assigned to parameters of the Execution Method type, generally found in the Scenario Parameters window.

To create a Local Execution value for use with an Execution Method Type within the MIMS framework:

1. Click the **New** button in the Project window. Choose Local Execution within the Parameter Values folder and type a name for the variable.
2. Click **OK**. The *This parameter value has no editable components or cannot be edited* dialog box appears.
3. Click **OK** to save the information, dismiss the dialog box, and create the parameter value.

Normal Distribution

If you wish to create a normal distribution or Gauss function (commonly referred to as a bell curve), the Normal Distribution Value is the MIMS member to create. This parameter value returns a single constant value chosen randomly from the distribution each time the value is called. The same random values will be returned each time a scenario is run if a random number seed is assigned a specific value in the Scenario Parameters window.

To create a Normal Distribution Value within the MIMS framework:

1. Click the **New** button in the Project Window. Choose Normal Distribution within the Parameter Values folder and type a name for the variable.
2. Click **OK**. The *Distribution parameters for :+[variable name]* dialog box appears.
3. In the Distribution parameters dialog box, enter the Mean (central value for the bell) and the Standard Deviation (width parameter). For a normal distribution, 68% of the values will be within one standard deviation of the mean, and 95% will be within two standard deviations of the mean.
4. Click **OK** to save the information, dismiss the dialog box, and create the parameter value.

MIMS also allows you to assign random numbers to values from a uniform distribution. The Generate Random Time Series module is an example of a member using distribution parameter values.

Secure Shell (SSH) Execution

If you wish to specify instructions for a secure shell (SSH) execution, the SSH Execution Value is the MIMS member to create. This parameter value returns the information necessary to operate programs on a remote server. Before using this value, you must establish secure shell remote access.

To create an SSH Execution Value within the MIMS framework:

1. Click the **New** button in the Project Window. Choose SSH Execution within the Parameter Values folder and type a name for the variable.
2. Click **OK**. The *SSH Execution Editor* dialog box appears.
3. Enter the User ID, default directory on the remote server, and Private Key File directory, name, and extension. To search for the Private Key File, use the **Browse** button.
4. Click on the **View** button beside Assumptions and Limitations in order to see the restrictions on an SSH execution. These include:

    - Only external modules (executable programs) can be distributed
    - All programs and initial input files must exist on the remote machine before the scenario is executed
    - Outputs will remain on the remote machine(s)
    - All selected remote machine(s) share a common file space
    - Secure shell software is available on local and remote machines.

5. Click on an available computer and then click on **Add Computer** to include it in the SSH execution.
6. Click **OK** to save the information, dismiss the dialog box, and create the parameter value.

More information on using secure shell servers is available at:
http://www.ssh.com/support/documentation/all/secureshellwinserver/.

Time Series Constant Value

If you wish to create a single parameter value to be used repeatedly in a time series, the Time Series Constant Value is the MIMS member to create. This parameter value returns a single constant value for any time. A pair of Time Series Constant Values may be useful if you wish to introduce a step change function into a model at a particular time using the IfElse formula.

To create a Time Series Constant Value within the MIMS framework:

1. Click the **New** button in the Project Window. Choose Time Series Constant Value within the Parameter Values folder and type a name for the variable.
2. Click **OK**. The *Edit Time Series Constant Value* dialog box appears.
3. In the *Edit Time Series Constant Value* dialog box, enter the Units, Constant Value, and a Description of the value.
4. Click **OK** to save the information, dismiss the dialog box, and create the parameter value.

Time Series from File Value

If you wish to read a time series from a file, the Time Series from File Value is the MIMS member to create. This parameter value reads a file that contains dates/times and an associated value for each date/time (e.g., the pairs in the file would appear as 1/1/03 and 2.1, 1/2/03 and 4.2, etc.).

To create a Time Series from File Value within the MIMS framework:

1. Click the **New** button in the Project Window. Choose Time Series from File Value within the Parameter Values folder and type a name for the variable.
2. Click **OK**. The *Edit Time Series from File Value* dialog box appears.
3. In the *Edit Time Series from File Value* dialog box, first type the file directory, name, and extension in the *File Name* field.
4. Indicate how times are referenced in the second line of this dialog box. Choose the time zone and whether or not Daylight Savings Time is applied to the times in the file.
5. Choose the appropriate delimiter and whether or not multiple delimiters should be treated as one.
6. If the first line of the file contains labels, type the date label and value label in the appropriate blanks. Also indicate the date/time format found in the file and the units for the parameter value.
7. If a particular code is used to indicate missing values (e.g., -9999), then check the *Recognize Missing Value* check box and type in the code in the *Missing Value* field.
8. Type a description of the parameter, its references, and when it would be useful in the *Description* field. Click **OK**.

Time Series from User Defined Function

If you wish to create a time series based on a mathematical construct, the Time Series from User Defined Function is the MIMS member to create. This parameter value uses a start date, end date, time step, and mathematical function to describe a time series.

To create a Time Series from User Defined Function within the MIMS framework:

1. Click the **New** button in the Project Window. Choose Time Series from User Defined Function within the Parameter Values folder and type a name for the variable.
2. Click **OK**. The *Edit Time Series from user defined Function* dialog box appears.
3. In the *Edit Time Series from user defined Function* dialog box, first choose the date format (note that DDD denotes Julian dates).
4. Type in the Start and End Dates (using the chosen date format). Also choose the Time Step and its Units.
5. Using the Python code format, type in the function where the variable read is called *time*. If using the time, calendar, or math functions, be sure to import those libraries before calling the functions. An example of a Python User Function is given below.
6. Enter a description of the parameter value in the *Description* field, and then click **OK**.

To construct a Python user function for a time series, you must include the following elements:

1. Use the variable *time*, where time is treated as the number of milliseconds since 01/01/1970.
2. Be sure to import any necessary Python libraries for functions you may wish to reference. Common libraries used for a time series would be:
   - time (http://www.python.org/dev/doc/devel//lib/module-time.html)
   - math (http://www.python.org/dev/doc/devel//lib/module-math.html)
   - calendar (http://www.python.org/dev/doc/devel//lib/module-calendar.html)
3. Use the *return* statement to specify the user-defined function.

The sample Python User Function shown below calculates the solar declination angle (in radians) based on the date and time. The solar declination angle can be used to find the solar zenith angle, solar azimuth, sunrise, and sunset.

```
## Time is in milliseconds
## The described function represents the solar declination
angle.

t=time

## Import the libraries with two different styles of the
commands.

import math
from time import *

## Convert time to seconds and compute the time of year
## and time of day.

t=t/1000.0
timestruct=gmtime(t)
day_of_year=float(timestruct[7])
hour=float(timestruct[3])

## Calculate the declination and return this value.

gamma=2.0*math.pi/365*(day_of_year -1+(hour-12)/24)
decl=0.006918 -0.399912*math.cos(gamma)+0.070257*math.sin(gamma)
   -0.006758*math.cos(2*gamma)+0.000907*math.sin(2*gamma)-
   0.002697*math.cos(3*gamma) +0.00148*math.sin(3*gamma)
return decl
```

Time Series in Memory Value

If you wish to create a time series one value at a time, the Time Series in Memory Value is the MIMS member to create. This parameter value holds a list of dates/times and an associated value for each date/time (e.g., the pairs in the list would appear as 1/1/03 and 2.1, 1/2/03 and 4.2, etc.).

To create a Time Series in Memory Value within the MIMS framework:

1. Click the **New** button in the Project Window. Choose Time Series in Memory Value within the Parameter Values folder and type a name for the variable.
2. Click **OK**. The *Edit Time Series in Memory Value* dialog box appears.
3. In the *Edit Time Series in Memory Value* dialog box, choose the date format, time zone, and whether or not to turn on the daylight savings adjustment. Also, type a description of the parameter, its references, and when it would be useful in the *Description* field. The units entry should describe the values, not the time units.
4. If you wish to enter the dates/times individually, click on the **Insert Row** button to create one new entry at a time where the dates/times must be entered manually. Then type the dates and values in the spaces that are created.
5. To create an ordered list of dates/times, click on the **Generate Rows** button, and the *Define Time Series* dialog box appears. Within the *Define Time Series* dialog box, select the Start Date, increment between Time Steps, Units for the time step, and the Number of Steps. Then click **OK** on the *Define Time Series* dialog box.
6. The **Generate Rows** button will create the dates on the *Edit Time Series in Memory Value* dialog box. Enter the values for each of the times, then click **OK**.

Uniform Distribution

If you wish to create a uniform distribution (where one value is as likely as any other across a range), the Uniform Distribution Value is the MIMS member to create. This parameter value returns a single constant value chosen randomly from the distribution each time the value is called. The same random values will be returned each time a scenario is run if a random number seed is assigned a specific value in the Scenario Parameters window.

To create a Uniform Distribution Value within the MIMS framework:

1. Click the **New** button in the Project Window. Choose Uniform Distribution within the Parameter Values folder and type a name for the variable.
2. Click **OK**. The *Distribution parameters for :"+[variable name]* dialog box appears.
3. In the distribution parameters dialog box, enter the Inclusive Minimum and Exclusive Maximum values. Any value generated by this distribution will be greater than or equal to the Inclusive Minimum and less than the Exclusive Maximum.
4. Click **OK** to save the information, dismiss the dialog box, and create the parameter value.

MIMS also allows you to assign random numbers to values from a normal distribution. The Generate Random Time Series module is an example of a member using distribution parameter values.

Appendix

Establishing Secure Shell Remote Access

To access the resources of a remote system, you need the following information:

- Internet Protocol (IP) address for the remote machine
- An account name and password on the remote machine
- The remote machine's number of available processors

The remote system must first be configured to allow secure shell access. The following description details how to establish access to a remote machine. Note that the instructions below frequently use the lower-case letter L, and this should not be confused with the letter I or the number 1. The instructions below describe ways to establish access the first time for the following operating systems:

From a Microsoft Windows system

From a UNIX system

After the first time you connect to a particular remote system, these steps do not have to be repeated. You can then direct the SSH executions through MIMS. After gaining access, you need to set up MIMS to recognize the remote operating system (Edit Administration Information Window) and then include a SSH execution parameter value as the Execution Method in the desired scenario.

More information on using secure shell servers is available at http://www.ssh.com/support/documentation/all/secureshellwinserver/.

From a Microsoft Windows system

1. In the MIMS directory, find the \bin subdirectory. Start the program "puttygen.exe" in this directory to generate the public and private keys. The public and private keys identify the communicating machines. If those files have not been included, they may be downloaded at http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html for Windows operating systems.
2. Click the **Generate** button in the *PuTTY Key Generator* dialog box. This will prompt you to move the mouse over the blank area of the dialog. Continue moving the mouse until the key is generated.

3. Type an entry into the *Key passphrase* field and the same entry again into the *Confirm passphrase* field. Remember the passphrase.
4. Click the **Save Public Key** button and name the saved file. Remember the subdirectory where the file is stored. Click **OK** to complete the action. Repeat the process by clicking the **Save Private Key** button.
5. Click on the X in the upper right corner of the *PuTTY Key Generator* dialog box to end this program.
6. Open a *Command Prompt* window from the **Start** menu on the Windows task bar. (Select the **Start/Programs/Accessories/Command Prompt** menu item.) In the *Command Prompt* window, type "cd MIMS\bin" and press **Enter** to change to the MIMS\bin directory (or substitute the appropriate path to the MIMS\bin directory).
7. Type "pageant [followed by the path and name of the private key]" at the command prompt and press **Enter**. The computer will prompt you for the passphrase. Type the passphrase and press **Enter**. This action will begin the secure shell agent service, and the Pageant icon should appear near the clock in the Windows task bar.
8. At the command prompt, type "plink [IP address of remote machine] -l [user name on remote machine] ls -l" and press **Enter**. You will be prompted for the password on the remote machine and may be asked to specify the public key files that should be added. After the prompts, the *Command Prompt* screen will list the folders and files in your directory on the remote machine ("ls" command). If the "plink" command does not operate successfully, reboot the local machine and try again.
9. In order to test the secure shell copy command from the command prompt, type "pscp -q plink.exe [user name]@[remote IP address]:plink.exe" and press **Enter**. You will be prompted for a password, and then this command should copy the plink.exe file to the remote machine.
10. The keys are now in place. Use the Edit Administration Information Window in MIMS to add the remote computer to the list of available operating systems.

From a UNIX system*

1. Setup key authentication: Run the command "ssh-keygen" to create public and private key files. When prompted, enter a name for the key file and a passphrase. Be sure to remember the passphrase. The public key file is the name you choose with the extension ".pub". The private file key is just the name you choose.
2. Appending keys: Add the contents of the public key file you created to the file /.ssh/authorized_keys in your home directory of the remote computer.
3. Establish a c-shell using the ssh-agent: Type the command "ssh-agent csh".
4. Connecting the key to the c-shell: Add the key to the ssh-agent using the command "ssh-add [name of private_key_file]" and enter the passphrase when prompted.

5. Test a secure shell command: Type "ssh -l [remote computer name] -l [user name on remote computer] ls -l" and the file directory on the remote computer should be shown.
6. Test the secure shell copy command: Type "scp -q -B -oProtocol=l [local file name] [user name]@[remote computer]:[remote file name]" to copy a local file to the remote machine.
7. Exit the shell by typing "exit" or with the command "ssh-agent -k".
8. The keys are now in place, and you should use the Edit Administration Information Window in MIMS to add the remote computer to the list of available operating systems.

* The UNIX section was drawn from Appendix A of the MIMS for Models-3 Applications Tutorial (February 2003).

Requesting Support

If you have questions concerning the use of the MIMS framework or spatial allocator with SMOKE or CMAQ, the Community Modeling and Analysis System Center provides support for those issues. Please submit your question to the CMAS Help Desk.

Other questions may be submitted or viewed on the MIMS support web page. The MIMS team will answer those questions as time permits.

Included Databases

List of included databases to be provided by the EPA

Licensing Information

When you install MIMS, a folder named Licenses is created. The software licenses appear within that folder.

Known Bugs

To view the current framework bugs, go to

http://sourceforge.net/tracker/?group_id=27492&atid=390709

Index