PNNL-16758

Pacific Northwest National Laboratory
Operated by Battelle for the U.S. Department of Energy

FRAMES-2.0 Software System: Linking to the Groundwater Modeling System (GMS) RT3D and MT3DMS Models

G. Whelan
K.J. Castleton
M.A. Pelton

August 2007

Prepared for
The U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Division of System Analysis & Regulatory Effectiveness
Rockville, Maryland 20852
under Contract DE-AC05-76RL01830

DISCLAIMER

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

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576 5728
email: reports@adonis.osti.gov

Available to the public from the National Technical Information Service,
U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161
ph: (800) 553-6847
fax: (703) 605-6900
email: orders@nits.fedworld.gov
online ordering: http://www.ntis.gov/ordering.htm

PNNL-16758

FRAMES-2.0 Software System: Linking to the Groundwater Modeling System (GMS) RT3D and MT3DMS Models

G. Whelan
K.J. Castleton
M.A. Pelton

August 2007

Prepared for
The U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Division of System Analysis & Regulatory Effectiveness
Rockville, Maryland 20852
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Summary

Linkages to the Groundwater Modeling System (GMS) have been developed at Pacific Northwest National Laboratory to enable the Nuclear Regulatory Commission (NRC) to more realistically assess the risk to the public of radioactive contaminants at NRC-licensed sites. Common software tools presently in use are limited in that they cannot assess contaminant migration through complex natural environments. The purpose of this initiative is to provide NRC with a licensing safety-analysis tool with sufficient power, flexibility, and utility that it can serve as the primary software platform for analyzing the hazards associated with licensing actions at those “complex” sites at which the traditional tools are inappropriate. As a tool designed to realistically approximate prospective doses to the public, this initiative addresses NRC’s safety-performance goal by confirming that licensing actions do not result in undue risk to the public.

Groundwater Modeling System (GMS) Import and pre- and post-processing tools (collectively herein refereed to as the GMS Import tool) have been developed to facilitate the linkage of GMS groundwater models with existing models associated with the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES). The GMS Import tool allows a GMS user to easily execute a Reactive Multi-Species Transport in Three-Dimensional (RT3D) module or Modular Three-Dimensional Multi-Species Transport Model (MT3DMS) module in FRAMES from a GMS-based RT3D-calibrated or MT3DMS-calibrated groundwater simulation. Through pre- and post-processing tools, the “created” RT3D or MT3DMS module in FRAMES has the capability to access and modify inputs to RT3D or MT3DMS without having to exercise the GMS user interface and without invalidating the GMS calibration. Although the GMS Import tool allows the FRAMES user access to modify input information to RT3D or MT3DMS and consume output from RT3D or MT3DMS, the basic design is applicable to all models housed within GMS or other such systems.

The main requirements of the GMS Import tool are explained in Section 2. This tool represents the fundamental capability in FRAMES to access and execute calibrated cases from GMS that use certain aspects of the RT3D and MT3DMS models. Because the linkage software (e.g., GMS Import tool) can only exhibit functionality that it is aware of, the requirements are based on those characteristics exhibited by the test cases upon which the GMS-FRAMES linkage is based. As such, there may be conditions associated with the test cases that do not exercise all possible situations.

The design elements of the GMS Import tool, which must meet the requirements identified in Section 2, are discussed in Section 3. Key to meeting those requirements is the capability to modify input and read outputs from RT3D and MT3DMS. The file specification is independent of whether the file is being read or written, so a single definition of the file should be sufficient to define the file format. The primary mechanism for getting information from the user to the model and getting information from the model to the FRAMES system is the System Input/Output Dynamic Link Library (SystemIO.DLL). The SystemIO.DLL is a set of routines used by FRAMES to facilitate communication between components.

The specifications (defined boundaries between software systems) are discussed in Section 4. The GMS Import tool dictates the boundaries that a FRAMES 2.0 module must meet. GMS is on one side, and FRAMES is on the other side. The testing approach and results are summarized in Section 5.

Acronyms and Abbreviations

API	Application Programming Interface
ARAMS	Adaptive Risk Assessment Modeling System
ASCII	American Standard Code for Information Interchange
BPP	Beanshell Pre-Processor
CAS	Chemical Abstracts Service
CSM	Conceptual Site Model, simplified description of the environmental problem to be modeled
DCE	Data Client Editor
DCE	dichloroethylene
DoD	Department of Defence
DOE	Department of Energy
EPA	U.S. Environmental Protection Agency
FRAMES	Framework for Risk Analysis in Multimedia Environmental Systems
GIS	geographic information system
GMS	Groundwater Modeling System
HWIR	Hazardous-Waste Identification Rule
MT3DMS	Modular Three-Dimensional Multi-Species Transport Model
NRC	Nuclear Regulatory Commission
OCRWM	Office of Civilian Radioactive Waste Management
OSWER	Office of Solid Waste and Emergency Response (EPA)
PNNL	Pacific Northwest National Laboratory
RT3D	Reactive Multi-Species Transport in Three-Dimension (module)
TCE	trichloroethylene
VC	vinyl chloride
WCF	Water Concentration File, containing time-varying contaminant concentrations

Acknowledgments

The authors would like to extend their appreciation to Dr. Ralph Cady of the U.S. Nuclear Regulatory Commission for supporting this work. Thanks are also extended to Dr. Philip Meyer of the Pacific Northwest National Laboratory (PNNL) for his technical review and valuable comments, and Mr. Wayne Cosby of PNNL for the final edit of the document. This work was funded by NRC under Contract DE-AC05-76RL01830.

Contents

Figures

1.0 Introduction

To assess the potential consequence of licensing actions, the Nuclear Regulatory Commission (NRC) must evaluate the potential for human exposure to radioactive contaminants. This assessment may include the potential future human exposure as a result of contaminant migration through complex natural environments. Common tools applied to these assessments are limited in applicability by the model's inherent assumptions and simplifications. While a simple conceptualization of these natural environments and associated models may be appropriate for many NRC-licensed sites, some sites are sufficiently complex that additional realism is warranted for assessing some of the exposure pathways. The purpose of this initiative is to provide NRC with a licensing safety-analysis tool with sufficient power, flexibility, and utility that it can serve as the primary software platform for analyzing the hazards associated with licensing actions at those "complex" sites at which the traditional tools are inappropriate. As a tool designed to realistically approximate prospective doses to the public, this initiative addresses NRC's safety-performance goal by confirming that licensing actions do not result in undue risk to the public. Because this tool will enable more realism to be incorporated into safety analyses, it will increase public confidence in our approach to resolving the problem. By creating a single software platform, this initiative will increase efficiency and effectiveness by reducing the need to continually customize software for specific problems.

Critical to appropriate modeling of these complex sites is the capability to readily merge results from domain-specific modeling activities into the complete assessment of the problem. An existing consortium of federal agencies has enlisted the aid of Pacific Northwest National Laboratory (PNNL) in developing the Framework for Risk Assessment in Multimedia Environmental Systems (FRAMES). The consortium currently consists of the Nuclear Regulatory Commission (NRC), Environmental Protection Agency (EPA), the Department of Defence (DoD), and the Department of Energy (DOE). PNNL is presently completing the design of an update to FRAMES that will increase the flexibility of the framework and merge existing related systems with FRAMES. PNNL designed and implemented the Hazardous-Waste Identification Rule (HWIR) for EPA's Office of Solid Waste and Emergency Response (OSWER) for national assessments of a suite of site databases and site-specific models along much the same lines as FRAMES but without some of the flexibility incorporated into FRAMES (EPA 2007a,b; Section 6.1). PNNL is also helping to develop the DoD Adaptive Risk Assessment Modeling System (ARAMS) using many of the same elements found within FRAMES with the addition of a number of more sophisticated databases and models (ACOE 2007).

NRC has joined an existing consortium of federal agencies that developed the DoD's Groundwater Modeling System (GMS). GMS is an industry-standard groundwater modeling software platform that is used within and outside the consortium to simulate and manage complex groundwater flow and transport modeling activities. Currently, there are multiple models for both groundwater flow and contaminant transport in GMS.

1.1 Background

FRAMES is a software platform that allows users the ability to select and implement environmental software models for risk assessment and management problems. This program is a flexible and holistic approach to understanding how activities affect humans and the environment. It links models that integrate across scientific disciplines, allowing for tailored solutions to specific activities, and it provides meaningful information to business and technical managers. FRAMES is the key to identifying, analyzing, and managing potential environmental, safety, and health risks. The purpose of FRAMES is to assist users in developing environmental scenarios and to provide options for selecting the most appropriate computer codes to conduct human and environmental risk-management analyses.

GMS is one of the most sophisticated and comprehensive groundwater modeling packages that has been developed, containing numerous numerical models and support features for modeling the groundwater environment. GMS provides tools for every phase of a groundwater simulation, including site characterization, model development, calibration, post-processing, and visualization. GMS has the capability to have a single user interface supply input to a number of groundwater models. Of the models housed within GMS, the modular computer code for simulating Reactive Multi-Species Transport in Three-Dimensional (RT3D) groundwater systems and the Modular Three-Dimensional Multi-Species Transport Model (MT3DMS) are the first two models that are accessed and executed by FRAMES, following a GMS stand-alone calibration.

1.2 Purpose

The objective is to provide NRC, Office of Nuclear Regulatory Research, Division of Systems Analysis and Regulatory Effectiveness with a dose-assessment tool for licensing decisions with sufficient power, flexibility, and utility that it can serve as their primary platform for analyzing the hazards associated with licensing actions at those “complex” sites at which the traditional dose-assessment tools are inappropriate. To meet this objective, this work will further the design and development of the evolving FRAMES Version 2 software to enable the integration of or linkage to models of complex domain-specific processes to evaluate the potential for human exposure to radioactive contaminants in complex natural environments. DOE’s PNNL, in conjunction with NRC, has developed a design for linking FRAMES and GMS. The objective of the linkage is to facilitate the exchange of model data (inputs/outputs) between the two modeling systems. Although the term GMS is used frequently and rather loosely, in the context of this work, it specifically refers to the computer models housed within GMS.

1.3 Component Description

The GMS Import tool allows a GMS user to easily create a RT3D-module or MT3DMS-module in FRAMES from a GMS-based RT3D-calibrated or MT3DMS-calibrated, groundwater simulation. The “created” RT3D or MT3DMS module in FRAMES has the capability to access and modify inputs to RT3D or MT3DMS without having to exercise the GMS user interface and without invalidating the GMS calibration. Although the GMS Import tool allows the FRAMES user access to modify input information to RT3D or MT3DMS and consume output from RT3D or MT3DMS, the basic design is applicable to all models housed within GMS or other such systems. The fundamental assumption is that a legacy modeling system is composed of executables, represented by the models and documented input and output file formats. These input and output file formats usually are American Standard Code for Information Interchange (ASCII) or unformatted binary, but the specific order and meaning of the values in the file are model specific. In some limited cases, as with GMS, the files may follow or extend an existing file format. The ASCII-formatted files can be further defined by whether they are fixed-width formatted or delimited. The fixed-width formatted setup usually has columns of a fixed number of characters, where delimited files use a special character, such as a comma, to show the beginning and end of an entry. Both fixed-width and delimited setups are used often in modeling systems.

2.0 Requirements

Requirements are characteristics and behaviors that a piece of software must possess to function adequately for its intended purpose. The GMS Import tool represents the fundamental capability in FRAMES to access and execute calibrated cases from GMS that use certain aspects of the RT3D and MT3DMS models. Because the linkage software (e.g., GMS Import tool) can only exhibit functionality that it is aware of, the requirements are based on those characteristics exhibited by the test cases upon which the GMS-FRAMES linkage is based. As more complicated and comprehensive RT3D or MT3DMS test cases are integrated with FRAMES, additional requirements may be necessary to confirm that these cases execute properly. For example, the RT3D import tool can be used to create a FRAMES module that will 1) ask the user for appropriate input values, 2) modify appropriate input files needed for RT3D, 3) run RT3D, and 4) read the results into a FRAMES standard dictionary. A dictionary file is a comma-delimited text file that contains the metadata about the data that are contained in a particular dataset.

The following main requirements of the GMS Import tool are listed as follows:

1. Read a GMS Project File, and display it to the user, creating FRAMES 2.0 compatible dictionary (*.dic) and module (*.mod) files.
2. Develop a FRAMES 2.0 module that will create the needed input files for RT3D or MT3DMS from FRAMES 2.0 datasets, run the RT3D or MT3DMS model, and read the outputs from RT3D or MT3DMS into FRAMES 2.0 datasets.
3. Correlate the User Identifiable Module Name with the names of Windows Explorer folders where the GMS and FRAMES simulation files are stored.
4. Document the Windows Explorer folder that contains the GMS gpr file that is imported to FRAMES using the GMS Import tool.
5. Allow the user to choose the RT3D or MT3DMS output file type in the form of an OBS or a CON file.
6. Develop a module that allows the user to change RT3D or MT3DMS input parameters designated by the GMS developer.
7. Allow for an RT3D or MT3DMS module to consume concentrations from an upstream FRAMES 2.0 module.
8. Allow the GMS user to map the GMS constituent name to a FRAMES 2.0 recognized constituent.
9. Provide a location for the user to define unit conversions between concentrations in RT3D or MT3DMS and mg/L.
10. Provide the user the choice of either using concentrations from an upstream module or concentrations originally provided to the RT3D or MT3DMS module in the baseline case. The baseline case refers to the case that was initially set up on GMS and is the one imported to FRAMES.
11. Allow the user to change the output location where water concentrations are transferred from an RT3D or MT3DMS simulation to a downstream FRAMES 2.0 module.

12. Allow for a module that consumes a point or an area source.
13. Provide the parallelepiped—Row, Column, Layer—in the GMS Import tool, which defines the volumetric boundary that contains an irregularly shaped area source.
14. Provide the user an option of defining a time associated with the end of the simulation.
15. Provide the user with an option of defining the number of reported time periods associated with the output in the CON file.

3.0 Design

Design elements are strategies for meeting requirements. The GMS Import tool is designed to meet the requirements identified in Section 2.0. Key to meeting those requirements is the capability to modify input and read outputs from RT3D and MT3DMS. The file specification is independent of whether the file is being read or written, so a single definition of the file should be sufficient to define the file format. The primary mechanism for getting information from the user to the model and getting information from the model to the FRAMES system is the System Input/Output Dynamic Link Library (SystemIO.DLL). The SystemIO.DLL is a set of routines used by FRAMES to facilitate communication between components.

3.1 Installs and Unpackaging Simulations Containing Linkages to GMS Models

The FRAMES 2.0 install was modified to account for linkages to the GMS project file, GMS models (RT3D and MT3DMS), and the GMS Import tool. A new FRAMES 2.0 install consists of an executable, titled FramesV2Setup.exe. The FramesV2Setup.exe should be saved to the host computer. The user runs the executable, and the appropriate file structure and set of directories are automatically developed and populated with

- FRAMES 2.0 software
- all models, databases, and viewers (e.g., MEPAS, GENII, ERDC) that were applicable to FRAMES 2.0 prior to linkages to GMS models
- executables for RT3D and MT3DMS (i.e., RT3D.exe and MT3DMS4.exe, respectively) and the GMS RXNS.dll file
- test cases in the form of Packaged Simulation files (i.e., *.zip files) and associated results in the from of Excel spreadsheets for five tests:[a]
  - RT3D Tutorial: Steady-state release as input, point source, single constituent, and using the OBS file as output
  - RT3D Tutorial: Steady-state release as input, point source, single constituent, and using the CON file as output
  - MT3DMS Tutorial: Steady-state release as input, point source, single constituent, and using the OBS file as output.
  - RT3D (Long): Time-varying release (to 180 yr) as input, area source, multiple constituents, using the OBS file as output, and a 200-yr simulation (run time approximately 2 hrs)
  - RT3D (Short): Time-varying release (to 20 yr) as input, area source, multiple constituents, using the OBS file as output, and a 20-yr simulation (run time approximately 15 min)
- GMS Import tool
- Updated Simulation Unpackager.

(a) Pentium 4 CPU 2.80GHz, 2.79 GHz 1 GB of RAM.

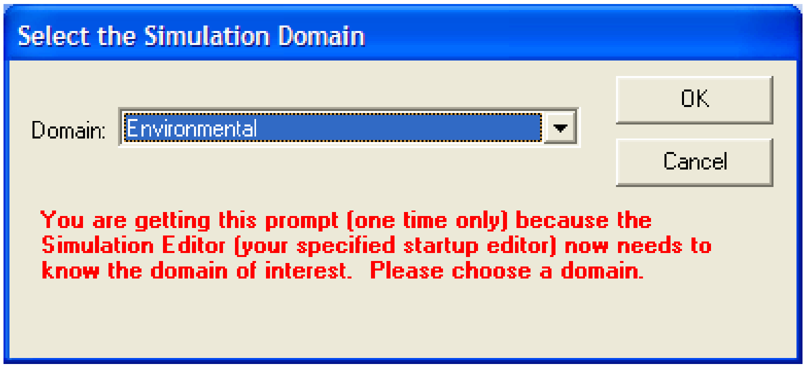
The new install automatically recreates the remote computer's FRAMES-relevant directory tree structure on the host machine. The FRAMES-relevant directory tree structure begins with the "FramesV2" folder, where ever that is located on the host machine. The directory tree structure is discussed in more detail under the section titled Managing Files. Once FRAMES 2.0 is installed on the host computer, the user runs the Framework Development Environment executable (FDE.exe). The first time the FDE is run, the user will be asked to select the default Domain, which defines the user interface associated with the Simulation Editor that greets the user when one first enters the system. Usually, the "Environmental" Domain is chosen, as illustrated by Figure 3.1.

In the previous version of the Simulation Unpackager, a packaged simulation, which is contained within a zip file, could only be successfully unpackaged if the host machine's directory tree structure exactly matched the remote computer's directory tree structure where the simulation was packaged. Unfortunately, the older version never told the user on the host machine where the simulation was unpackaged or the name of the simulation (*.sim) file contained in the zip file. These deficiencies have been corrected.

Prior to unpackaging a simulation, the user must save the zip file to the root directory that contains the FramesV2 folder; it can be placed in any folder, as long as it is in the FramesV2 root directory. These zip files should not be extracted using tools like WinZip but should be unpackaged using the FRAMES 2.0 Simulation Unpackager under the Tools menu. When the simulation is successfully unpackaged, the user is notified with a text box noting the folder into which the simulation was saved and the name of the simulation (*.sim) file, as illustrated in Figure 3.2. The Simulation Unpackager also automatically recreates the directory tree structure that was on the remote machine when the simulation was first packaged.

Once unpackaged, there is the possibility that the dic and mod files have not been automatically registered. In this case, the user needs to implement the Dictionary and Module Editors to register the dic and mod files. For setting up the GMS cases, this can also be done automatically by implementing the GMS Import tool. In the event the GMS Import tool is implemented before the Simulation Unpackager, the user will have to re-run the GMS Import tool again.

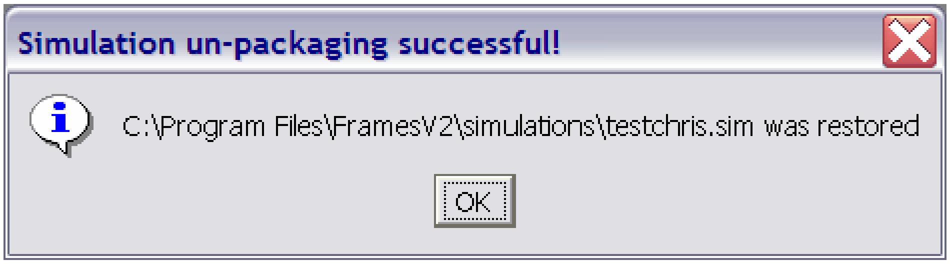
As noted above, five pre-packaged simulations are included in the install, all packaged as zip files. The zip files contain test cases, which can run on the host machine to check to confirm that the install performed as designed. They also demonstrate that the new software works as designed.



Domain: Environmental

You are getting this prompt (one time only) because the Simulation Editor (your specified startup editor) now needs to know the domain of interest. Please choose a domain.

Figure 3.1. Selecting the Default FRAMES 2.0 Domain


Simulation un-packaging successful!

C:\Program Files\FramesV2\simulations\testchris.sim was restored

Figure 3.2. Message Indicating that the Simulation Was Successfully Unpackaged

3.2 GMS Import Tool

The GMS Import tool generates batch files and associated text files from user-defined input (i.e., input choices) and then constructs a FRAMES 2.0 module that uses these batch and text files. The GMS Import tool currently only imports RT3D and MT3DMS runs. The GMS project and input and output files can be in any location. The model executable files will be placed in C:\Program Files\FramesV2\GMS by the FramesV2 install. For RT3D model, this includes RT3D.exe (Version 2.5) and RXNS.dll. For MT3D, this includes MT3DMS4.exe (Version 4.0). If GMS 5.0 is already installed, then no change is required. The GMS Import tool is installed by default by the FramesV2 install and will add the required files to a C:\Program Files\FramesV2\GMS directory. Once installed, the user will access the GMS Import tool through the FRAMES User Interface by going to Tools and selecting GMS Import.

3.2.1 GMS Import Tool User Interface

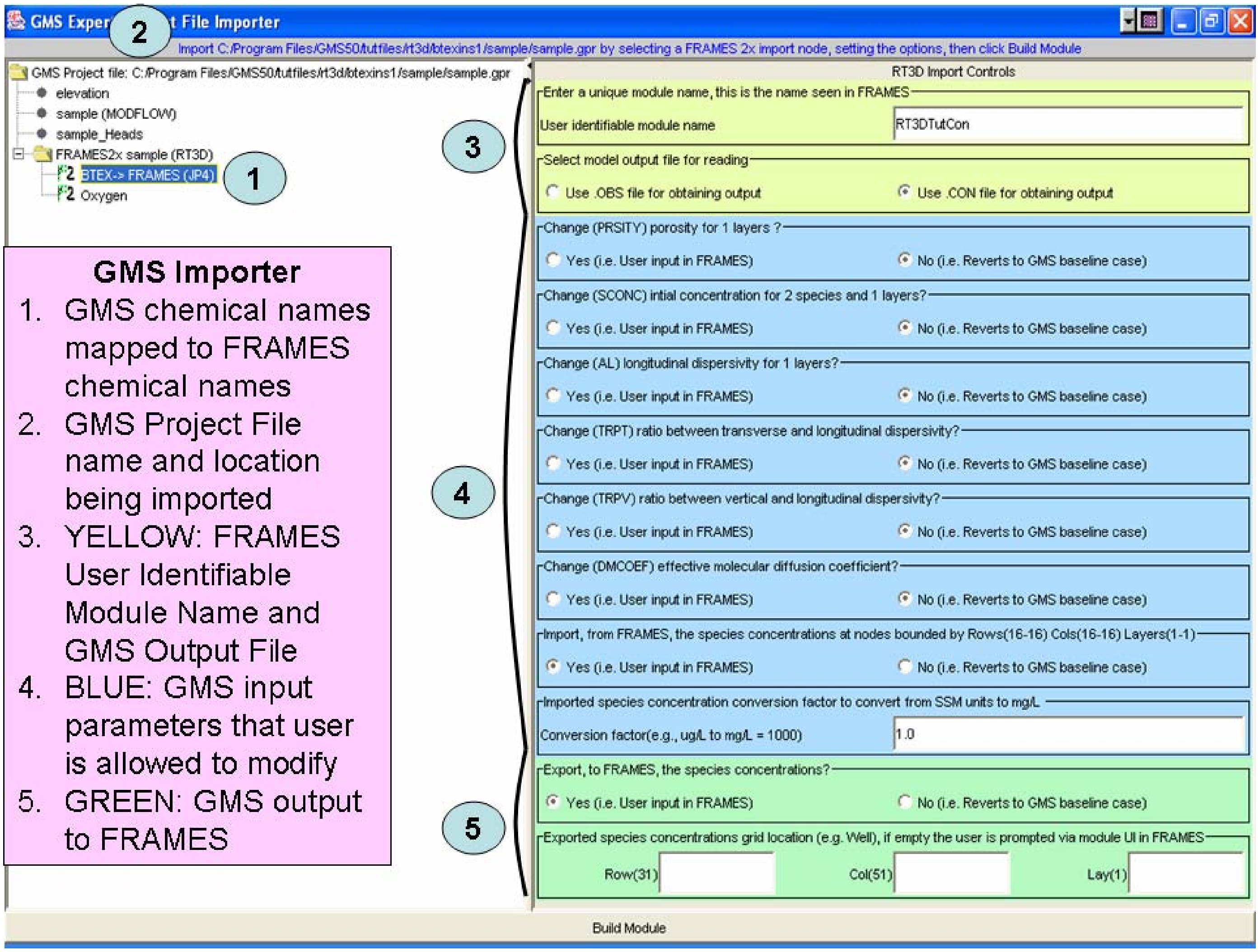
Once registered, the GMS Import screen will come up and allow the user to begin linking FRAMES and the selected GMS model. Figure 3.3 presents a screen illustrating the registration of a GMS model being imported to FRAMES using the GMS Import tool. The layout of the screen is meant to be a continuation of the GMS screen. The "tree" control on the left side (i.e., "1") shows the constituents that can be imported. If the constituents are not visible, click the plus sign to the right of FRAMESV2. The right side of the screen is the options area for import processing. When the button at the bottom of the display (i.e., "Build Module") is pressed, a new module is generated and is automatically registered with FRAMES 2.0. Key characteristics include the following:

- Divides the input screen into left- and right-hand sides.
  - The left-hand side provides a listing of the potential chemicals in GMS that need to be mapped to chemical names that are recognizable in FRAMES (see "1" in Figure 3.3). This may mean that GMS has identified a chemical or form of a chemical that is not in the FRAMES chemical list. To confirm compatibility between systems, the user is required to map the GMS chemical name to a FRAMES chemical name. If one does not exist, then the user is required to choose a surrogate chemical name from a list of FRAMES chemical names. The choice is made through a chemical selection wizard (not shown here), and the results of those choices are reported after the chemical name in the FRAMES-GMS Import User Interface. For example, in the chemical listing in Figure 3.3, where the green "F2" appears under the "FRAMES2X Sample (RT3D)" heading, GMS identifies the chemicals BTEX and Oxygen. Because only BTEX is a constituent of concern, only it has been mapped to the FRAMES database, which does

not contain BTEX but does contain JP4; so BTEX is mapped to JP4 [i.e., BTEX → FRAMES (JP4)].

- The right-hand side divides the screen into three color-coded sections: yellow (input and output files), blue (input parameters), and green (output location), corresponding to "3, 4, and 5" in Figure 3.3, respectively).

- Identifies the name of the GMS project file (i.e., sample.gpr) that is being imported to FRAMES (see "2" in Figure 3.3 and Figure 3.4) and location from where the GMS project file was imported. As Figure 3.4 illustrates, the program file sample.gpr is being imported from C:\Program Files\GMS\tutfiles\rt3d\btexins1\sample.



Import C:/Program Files/GMS50/tutfiles/rt3d/btexins1/sample/sample.gpr by selecting a FRAMES 2x import node, setting the options, then click Build Module

GMS Project file: C:/Program Files/GMS50/tutfiles/rt3d/btexins1/sample/sample.gpr
- elevation
- sample (MODFLOW)
- sample_Heads
- FRAMES2x sample (RT3D)
  - BTEX-> FRAMES (JP4)
  - Oxygen

Enter a unique module name, this is the name seen in FRAMES
User identifiable module name: RT3DTutCon

Select model output file for reading
○ Use .OBS file for obtaining output ● Use .CON file for obtaining output

Change (PRSITY) porosity for 1 layers ?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (SCONC) intial concentration for 2 species and 1 layers?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (AL) longitudinal dispersivity for 1 layers?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (TRPT) ratio between transverse and longitudinal dispersivity?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (TRPV) ratio between vertical and longitudinal dispersivity?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (DMCOEF) effective molecular diffusion coefficient?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Import, from FRAMES, the species concentrations at nodes bounded by Rows(16-16) Cols(16-16) Layers(1-1)
● Yes (i.e. User input in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

Imported species concentration conversion factor to convert from SSM units to mg/L
Conversion factor(e.g., ug/L to mg/L = 1000): 1.0

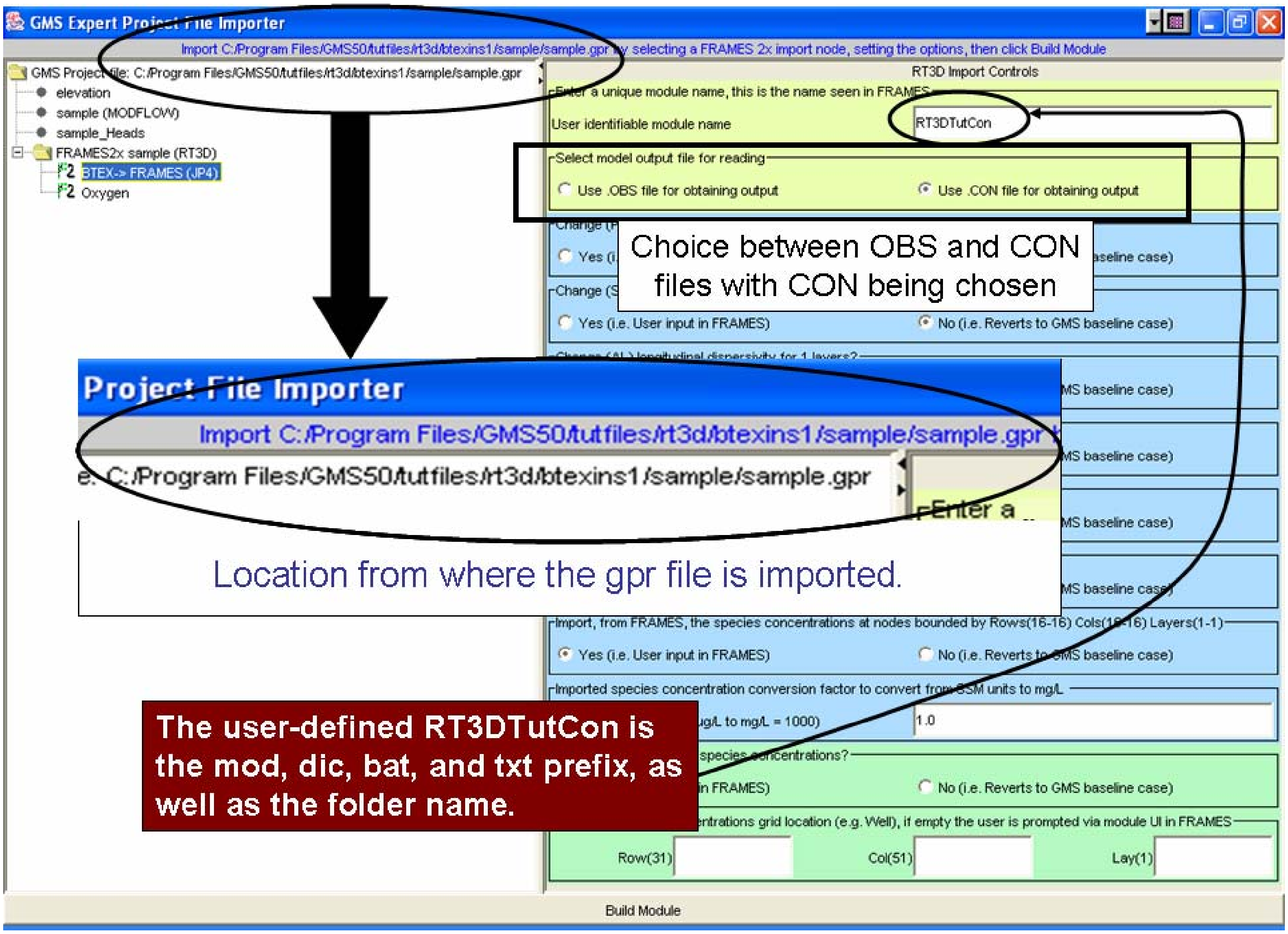
Export, to FRAMES, the species concentrations?
● Yes (i.e. User input in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

Exported species concentrations grid location (e.g. Well), if empty the user is prompted via module UI in FRAMES
Row(31) Col(51) Lay(1)

GMS Importer
1. GMS chemical names mapped to FRAMES chemical names
2. GMS Project File name and location being imported
3. YELLOW: FRAMES User Identifiable Module Name and GMS Output File
4. BLUE: GMS input parameters that user is allowed to modify
5. GREEN: GMS output to FRAMES

Figure 3.3. GMS Import Tool Interface

- Identifies the alias of the user-defined name in FRAMES (i.e., RT3DTutCon in Figures 3.1 and 3.2) that refers to the imported GMS module (see “3” in Figure 3.3, and Figure 3.4).
- If the user decides to reuse a User Identifiable Module Name, the GMS Import tool will notify the user that the name has already been previously used and will ask the user if he/she would like to proceed, as illustrated in Figure 3.5. If the user proceeds to use the same User Identifiable Module Name, then all older and previously developed files located in the directory of the same name under C:\Program Files\FramesV2\GMS will be overwritten, and the module will be successfully built accordingly, as illustrated in Figure 3.6.



Import C:/Program Files/GMS50/tutfiles/rt3d/btexins1/sample/sample.gpr by selecting a FRAMES 2x import node, setting the options, then click Build Module

GMS Project file: C:/Program Files/GMS50/tutfiles/rt3d/btexins1/sample/sample.gpr
- elevation
- sample (MODFLOW)
- sample_Heads
- FRAMES2x sample (RT3D)
  - BTEX-> FRAMES (JP4)
  - Oxygen

RT3D Import Controls

Enter a unique module name, this is the name seen in FRAMES
User identifiable module name: RT3DTutCon

Select model output file for reading
○ Use .OBS file for obtaining output ● Use .CON file for obtaining output

Change (S
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (AL) longitudinal dispersivity for 1 layers?

Import, from FRAMES, the species concentrations at nodes bounded by Rows(16-16) Cols(16-16) Layers(1-1)
● Yes (i.e. User input in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

Imported species concentration conversion factor to convert from GSM units to mg/L
ug/L to mg/L = 1000) 1.0

species concentrations?
in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

ntrations grid location (e.g. Well), if empty the user is prompted via module UI in FRAMES
Row(31) Col(51) Lay(1)

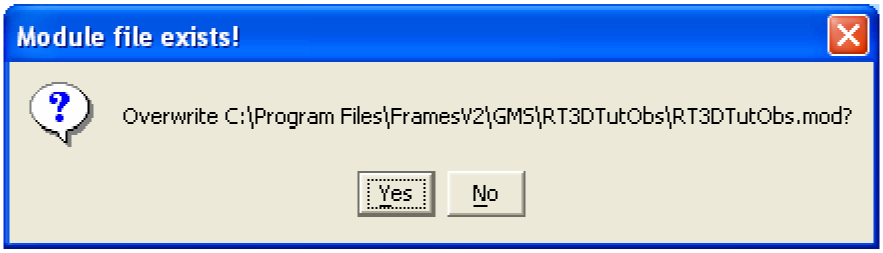
Choice between OBS and CON files with CON being chosen

Project File Importer
Import C:/Program Files/GMS50/tutfiles/rt3d/btexins1/sample/sample.gpr
e: C:/Program Files/GMS50/tutfiles/rt3d/btexins1/sample/sample.gpr

Location from where the gpr file is imported.

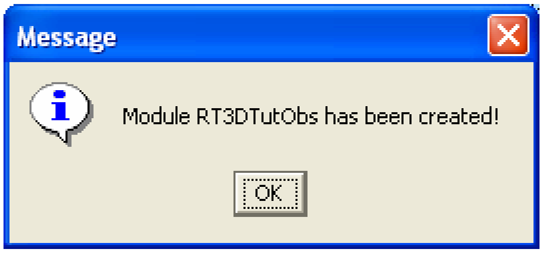
The user-defined RT3DTutCon is the mod, dic, bat, and txt prefix, as well as the folder name.

Figure 3.4. Explanations of the Components Comprising the GMS Import Tool User Interface


Module file exists!

Overwrite C:\Program Files\FramesV2\GMS\RT3DTutObs\RT3DTutObs.mod?

Figure 3.5. Text Box Indicating that the File Name Already Exists


Message

Module RT3DTutObs has been created!

Figure 3.6. Text Box Indicating that the Module Has Been Successfully Created

- Provides the user with the option of using RT3D/MT3DMS output data from the OBS file or the CON file. The OBS (i.e., OBServation) file contains data related to a specified location (e.g., a withdrawal well location). The CON file contains data for all locations. The OBS file is significantly smaller than the CON file.

- Presents the list of input variables that the GMS developer allows the FRAMES user to modify (blue region denoted as “4” in Figure 3.3). If the user chooses to modify any of these input variables, it is assumed that these parameters are spatially uniform within the GMS baseline case so that one value represents all locations. The baseline case refers to the case that was initially set up on GMS and is the one imported to FRAMES. If the value of the parameter is not uniformly distributed and constant in the system, then the text box for this parameter will not appear in the GMS Import user interface, and the baseline assumptions will prevail.

- Identifies the spatial extent of the volumetric source associated with GMS (e.g., RT3D). Each Row, Column, and Layer includes a qualifier (in parentheses) indicating an inclusive volume (i.e., parallelepiped) within which the contamination exists. For example, for Rows (12 to 16), Columns (2 to 6), and Layers (1 to 1), the source would be located within a parallelepiped with dimensions of 5 Rows, 5 Columns, and 1 Layer. The source may be irregular and may not completely fill the parallelepiped, but it would not have an areal extent outside of the designated parallelepiped. In Figure 3.7, contamination of the source exists within Rows (16 to 16), Columns (16 to 16), and Layers (1 to 1), inclusive, which is a point source at 16, 16, and 1. With this information, the user has a general idea as to the location of the source in relation to any user-defined output location. This information is captured by the following text box in Figure 3.7: “Import, from FRAMES, the species concentrations at nodes bounded by Row (16-16) Cols (16-16) Layers (1-1).” The user has the choice of accepting input from an upstream model (i.e., Yes), or using the current boundary conditions already simulated within GMS prior to importing the baseline case to FRAMES (i.e., No). In other words, if “No” is chosen, input to RT3D or

MT3DMS from an upstream model in FRAMES will be ignored, and the original boundary condition used in the GMS baseline case will be used.

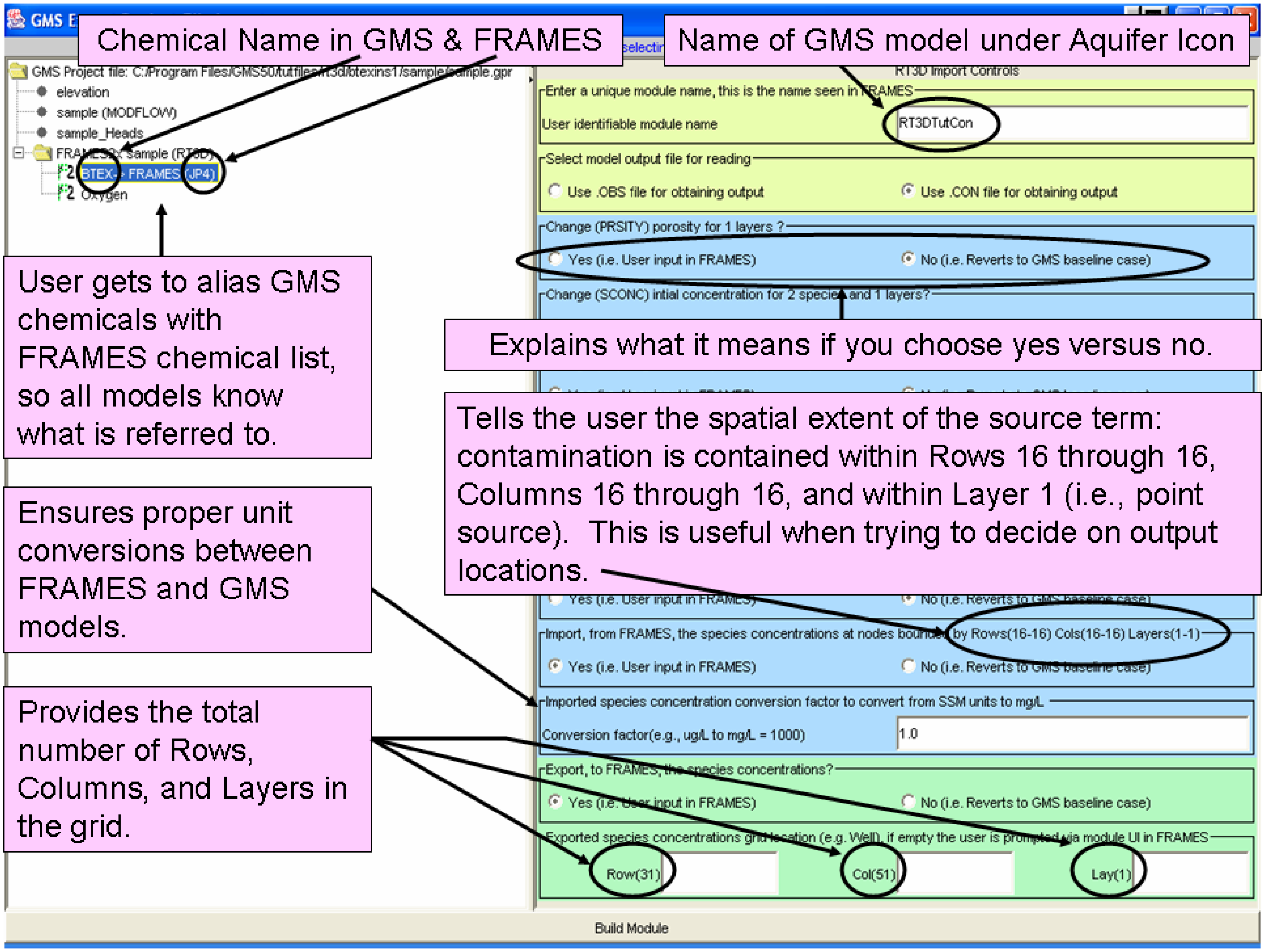
- Limits the size of the volumetric source within the parallelepiped to 200 cells.
- Provides the user a choice between using data generated by an upstream module in FRAMES to represent input to the GMS module or having the GMS module revert to the input boundary conditions already associated with the baseline case. If the source cells are not contiguous (i.e., touching) or do not contain a constant concentration across the source cells, then the system will not allow the user to generate data by an upstream module in FRAMES to represent input to the GMS module; the input will always be defined by the baseline case. The current input source models in FRAMES represent single source terms and produce uniform concentrations over the source.
- Makes sure of proper unit conversions between FRAMES and GMS models. The units in GMS are not necessarily the same as those in other models linked to FRAMES. Sometimes, units are not required by GMS, as some GMS models assume that the user will use consistent units throughout. When linking to other models, knowledge of units is of primary importance.
- Provides the user a choice between using the GMS module output generated from within FRAMES (e.g., OBS or CON file), or skipping the FRAMES-based output and using the baseline simulation results. The latter is chosen if the user wants to only supply downstream models with the baseline simulation results.
- Allows the user to define the output location (Row, Column, and Layer).
- Provides the total number of Rows, Columns, and Layers in the grid (see Figure 3.3, last text box in the green section). For example, Figure 3.7 illustrates that this GMS imported case has a grid size of 31 Rows, 51 Columns, and 1 Layer. When choosing an output location, it is useful to know the maximum size of the grid, so a location outside the grid is not identified. If the user defines an output location in the GMS Import user interface, then this location is fixed and cannot be modified at a later date. If the user leaves these text boxes blank, then the user will again be asked for the location (i.e., Row, Column, and Layer) in the module’s user interface, which will allow the user to change the location from run to run.

Import, from FRAMES, the species concentrations at nodes bounded by Row (16-16) Cols (16-16) Layer (1-1).

A summary of the attributes associated with the GMS Import user interface is provided in Figure 3.7.

3.2.2 Registering GMS Model Executables

The new dialogs in the GMS Import tool are implemented in Java using standard controls. The “File” dialog appears, as illustrated in Figure 3.8, after the user selects Build Module in the GMS Import tool main screen (see Figure 3.3). The first time the GMS Import tool encounters a gpr file containing a model name that has not been registered, following use of the Build Module button, it will ask the user to browse the host system and identify the directory that contains the executable for the model of interest (i.e., RT3D.exe or MT3DMS4.exe). In this particular example (Figure 3.8), the dialog is prompting the user to identify the directory containing the RT3D model executable. Note that the dialog title indicates which model executable is being sought. A similar title will appear for MT3DMS. Once identified, the path to the executable is saved to the GMSImport.ini file. Because two models are recognized by the system (RT3D and MT3DMS), this screen will appear only twice—the first time each model is chosen. With any additional linkage to gpr files that contain references to these two models, the screen will not appear again. In the event that no model is selected using the "File" dialog, an "Import cancelled, no model selected!" message will appear, as illustrated in Figure 3.9. The program will still continue to operate after the user presses "OK," and the user is returned back to the GMS Import screen.



Chemical Name in GMS & FRAMES

Name of GMS model under Aquifer Icon

GMS Project file: C:/Program Files/GMS50/tutfiles/rt3d/btexins1/sample/sample.gpr
- elevation
- sample (MODFLOW)
- sample_Heads
- FRAMES2x sample (RT3D)
  - BTEX-> FRAMES (JP4)
  - Oxygen

User gets to alias GMS chemicals with FRAMES chemical list, so all models know what is referred to.

Ensures proper unit conversions between FRAMES and GMS models.

Provides the total number of Rows, Columns, and Layers in the grid.

RT3D Import Controls

Enter a unique module name, this is the name seen in FRAMES
User identifiable module name: RT3DTutCon

Select model output file for reading
○ Use .OBS file for obtaining output ● Use .CON file for obtaining output

Change (PRSITY) porosity for 1 layers ?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (SCONC) intial concentration for 2 species and 1 layers?

Explains what it means if you choose yes versus no.

Tells the user the spatial extent of the source term: contamination is contained within Rows 16 through 16, Columns 16 through 16, and within Layer 1 (i.e., point source). This is useful when trying to decide on output locations.

○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Import, from FRAMES, the species concentrations at nodes bounded by Rows(16-16) Cols(16-16) Layers(1-1)
● Yes (i.e. User input in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

Imported species concentration conversion factor to convert from SSM units to mg/L
Conversion factor(e.g., ug/L to mg/L = 1000): 1.0

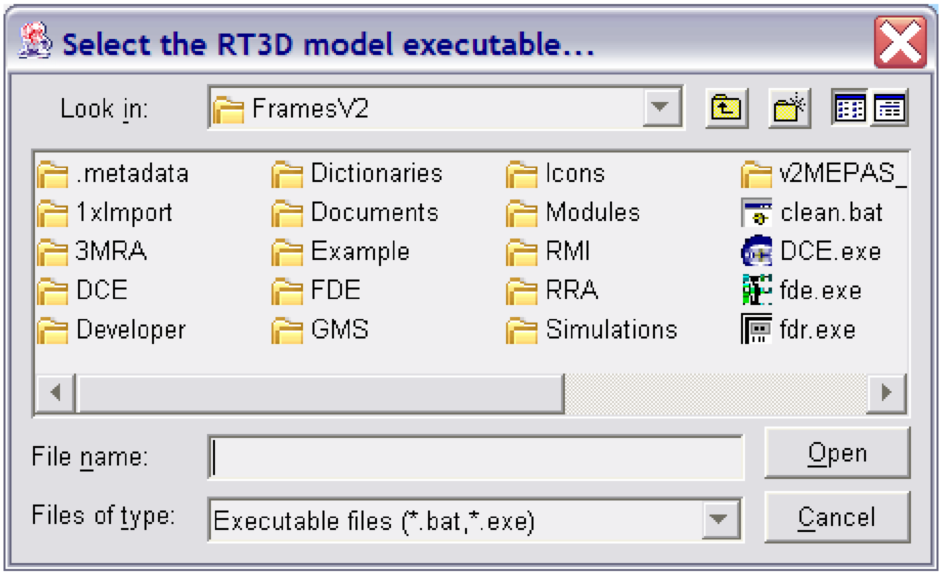
Export, to FRAMES, the species concentrations?
● Yes (i.e. User input in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

Exported species concentrations grid location (e.g. Well), if empty the user is prompted via module UI in FRAMES
Row(31) Col(51) Lay(1)

Figure 3.7. Summary of the Attributes Associated with the GMS Import Tool User Interface

3.3 Managing Files

When linking two modeling frameworks for the purpose of seamless communication, there are data sets that are applicable to each framework. Keeping them separate allows for efficient tracking and management of the files. GMS provides a completed baseline simulation with its assorted files, and the application of FRAMES produces its own output data sets that also need to be managed separately. In addition, since the files contain different names, consistent directory nomenclature and naming conventions are desirable, so the user can track and manage both the GMS imported files and the FRAMES simulation files within the host computers directory tree structure.



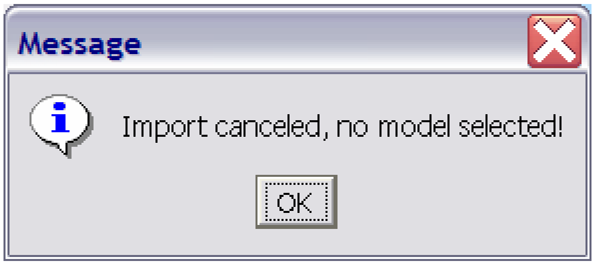
Select the RT3D model executable...

Look in: FramesV2

File name:

Files of type: Executable files (*.bat,*.exe)

Figure 3.8. User Interface for Registering the Location of the GMS Model Executables

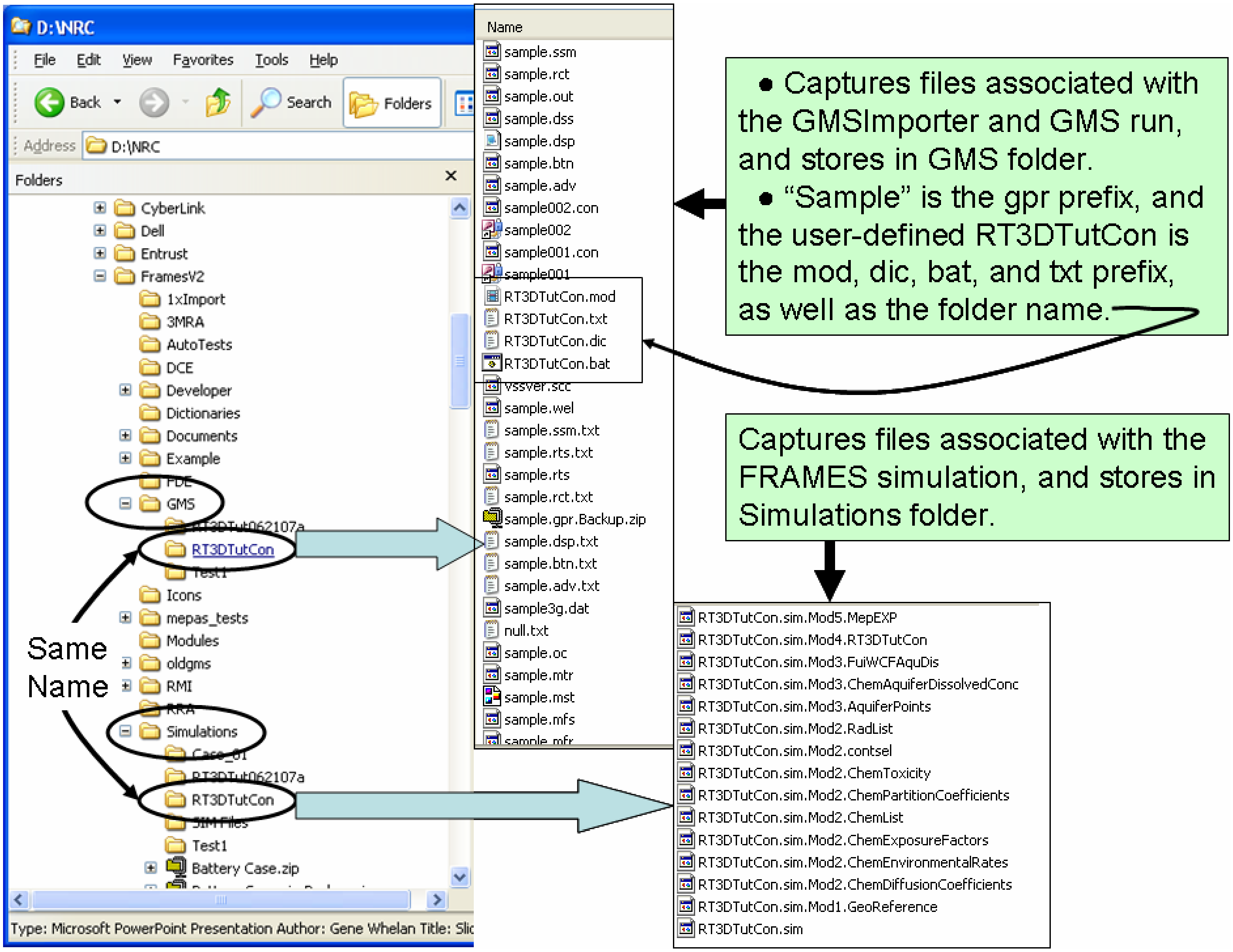


Message

Import canceled, no model selected!

Figure 3.9. Message if the User Does Not Select the Location of the GMS Model Executables

As noted earlier, the GMS Import tool records the location from where the project file (*.gpr) is archived (see Figures 3.3 and 3.4). The GMS Import tool allows the user to identify a unique name associated with the GMS module that is about to be created within FRAMES. This User Identifiable Module Name is the prefix on the FRAMES mod, dic, bat, and txt files, as well as the folder name where the GMS and FRAMES-Simulation files are stored on the host computer. Figure 3.10 provides screen captures of the file structure that was automatically set up by FRAMES using the User Identifiable Module Name "RT3DTutCon" defined in the GMS Import tool. The GMS files are located under C:\Program Files\FramesV2\**GMS**, and the FRAMES Simulation files are located under C:\Program Files\FramesV2\Simulations. If these folders do not exist, the system will create them. The **GMS** files are those that relate to the imported baseline GMS simulation, and the FRAMES **Simulations** files refer to those files developed as part of the FRAMES Conceptual Site Model (CSM) application developed under the Simulation Editor.



Name
sample.ssm
sample.rct
sample.out
sample.dss
sample.dsp
sample.btn
sample.adv
sample002.con
sample002
sample001.con
sample001
RT3DTutCon.mod
RT3DTutCon.txt
RT3DTutCon.dic
RT3DTutCon.bat
vssver.scc
sample.wel
sample.ssm.txt
sample.rts.txt
sample.rts
sample.rct.txt
sample.gpr.Backup.zip
sample.dsp.txt
sample.btn.txt
sample.adv.txt
sample3g.dat
null.txt
sample.oc
sample.mtr
sample.mst
sample.mfs
sample.mfr

- Captures files associated with the GMSImporter and GMS run, and stores in GMS folder.
- "Sample" is the gpr prefix, and the user-defined RT3DTutCon is the mod, dic, bat, and txt prefix, as well as the folder name.

Captures files associated with the FRAMES simulation, and stores in Simulations folder.

RT3DTutCon.sim.Mod5.MepEXP
RT3DTutCon.sim.Mod4.RT3DTutCon
RT3DTutCon.sim.Mod3.FuiWCFAquDis
RT3DTutCon.sim.Mod3.ChemAquiferDissolvedConc
RT3DTutCon.sim.Mod3.AquiferPoints
RT3DTutCon.sim.Mod2.RadList
RT3DTutCon.sim.Mod2.contsel
RT3DTutCon.sim.Mod2.ChemToxicity
RT3DTutCon.sim.Mod2.ChemPartitionCoefficients
RT3DTutCon.sim.Mod2.ChemList
RT3DTutCon.sim.Mod2.ChemExposureFactors
RT3DTutCon.sim.Mod2.ChemEnvironmentalRates
RT3DTutCon.sim.Mod2.ChemDiffusionCoefficients
RT3DTutCon.sim.Mod1.GeoReference
RT3DTutCon.sim

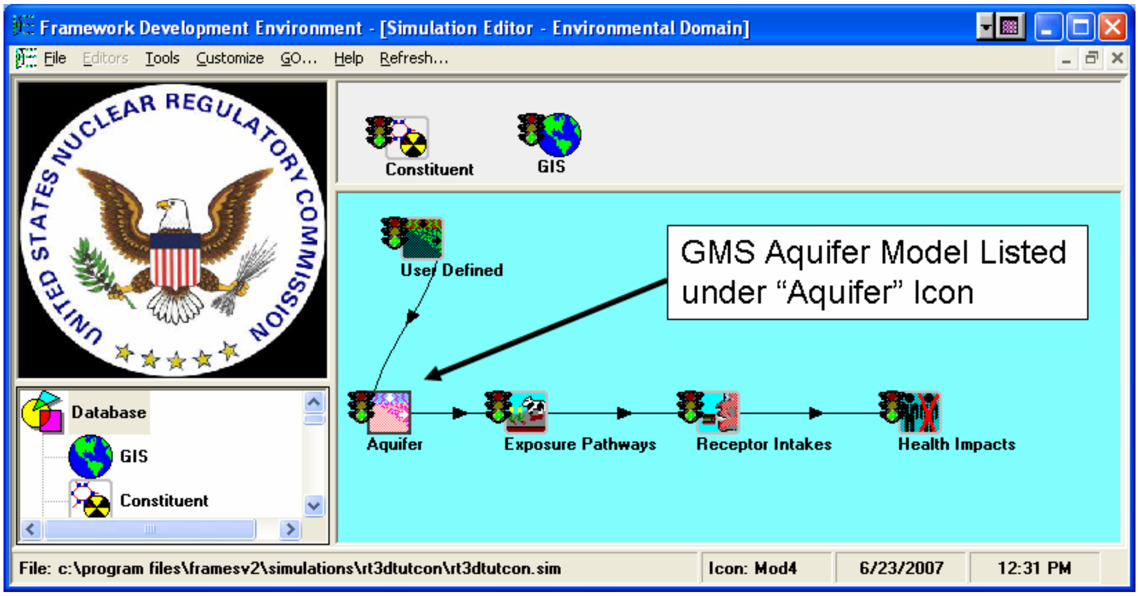
Figure 3.10. File Structure that is Automatically Created to Track GMS and FRAMES-Simulation Files Associated with the User Identifiable Module Name

If the user decides to reuse the same User Identifiable Module Name from within the GMS Import tool, it will notify the user that the name has already been previously used and will ask the user if he/she would like to proceed (see Figure 3.5). If the user proceeds to use the same User Identifiable Module Name, then all older and previously developed files located in the directory of the same name under C:\Program Files\FramesV2\GMS will be overwritten (see Figure 3.6).

The User Identifiable Module Name is also the name of the GMS module in the CSM application. Following execution of the GMS Import tool, the FRAMES 2.0 user can then create a CSM diagram that includes an aquifer, as illustrated in Figure 3.11 by choosing the Simulation Editor under the Editors menu. A CSM has been constructed in this figure, the GMS model name is chosen, and its associated Project File name and location are listed in the FRAMES module General Info user interface (see Figure 3.12). The FRAMES user can simply choose the module of the same name as that provided in the

GMS Import tool (Figure 3.12). Once the module has been chosen and the user exits the General Info user interface, the name and location of the file appear at the bottom of the screen (see Figure 3.13).

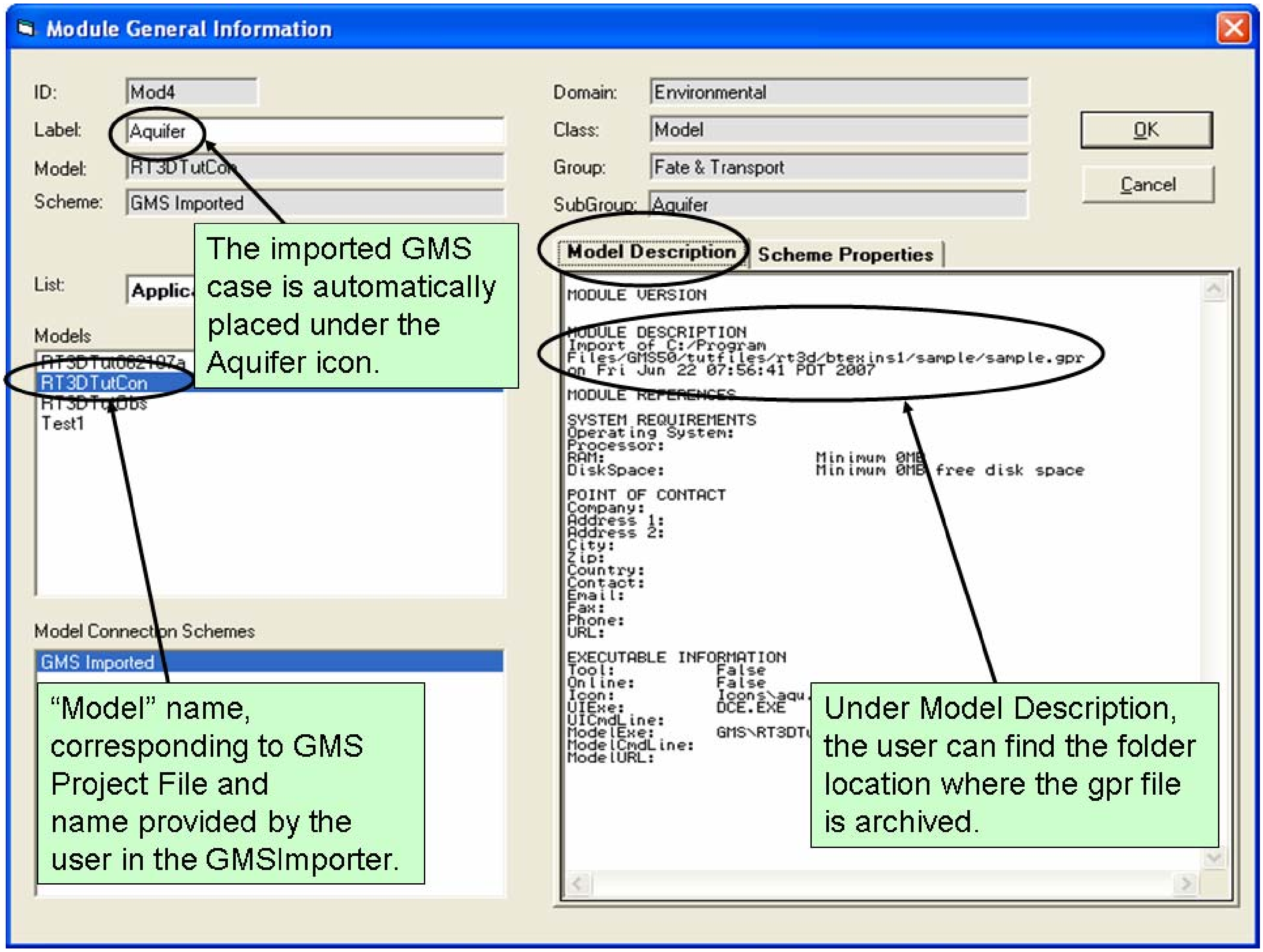
Consistency now exists between the User Identifiable Module Name, GMS file directory name, and FRAMES Simulations file directory name, all housed within C:\Program Files\FramesV2, if the user allows the system to establish this particular structure. If the user creates too many of these folders, the user has the option of just deleting folders, provided FRAMES is closed, without having to go through the process of deleting the mod and dic files through the Module and Dictionary editors, respectively, under the Editors menu.



GMS Aquifer Model Listed under “Aquifer” Icon

File: c:\program files\framesv2\simulations\rt3dtutcon\rt3dtutcon.sim | Icon: Mod4 | 6/23/2007 | 12:31 PM

Figure 3.11. Illustrative Example of the Aquifer Icon and Where the GMS Aquifer Module Would Be Located



ID: Mod4
Label: Aquifer
Model: RT3DTutCon
Scheme: GMS Imported
List: Applic

Domain: Environmental
Class: Model
Group: Fate & Transport
SubGroup: Aquifer

Models
RT3DTut062107a
RT3DTutCon
RT3DTutObs
Test1

Model Connection Schemes
GMS Imported

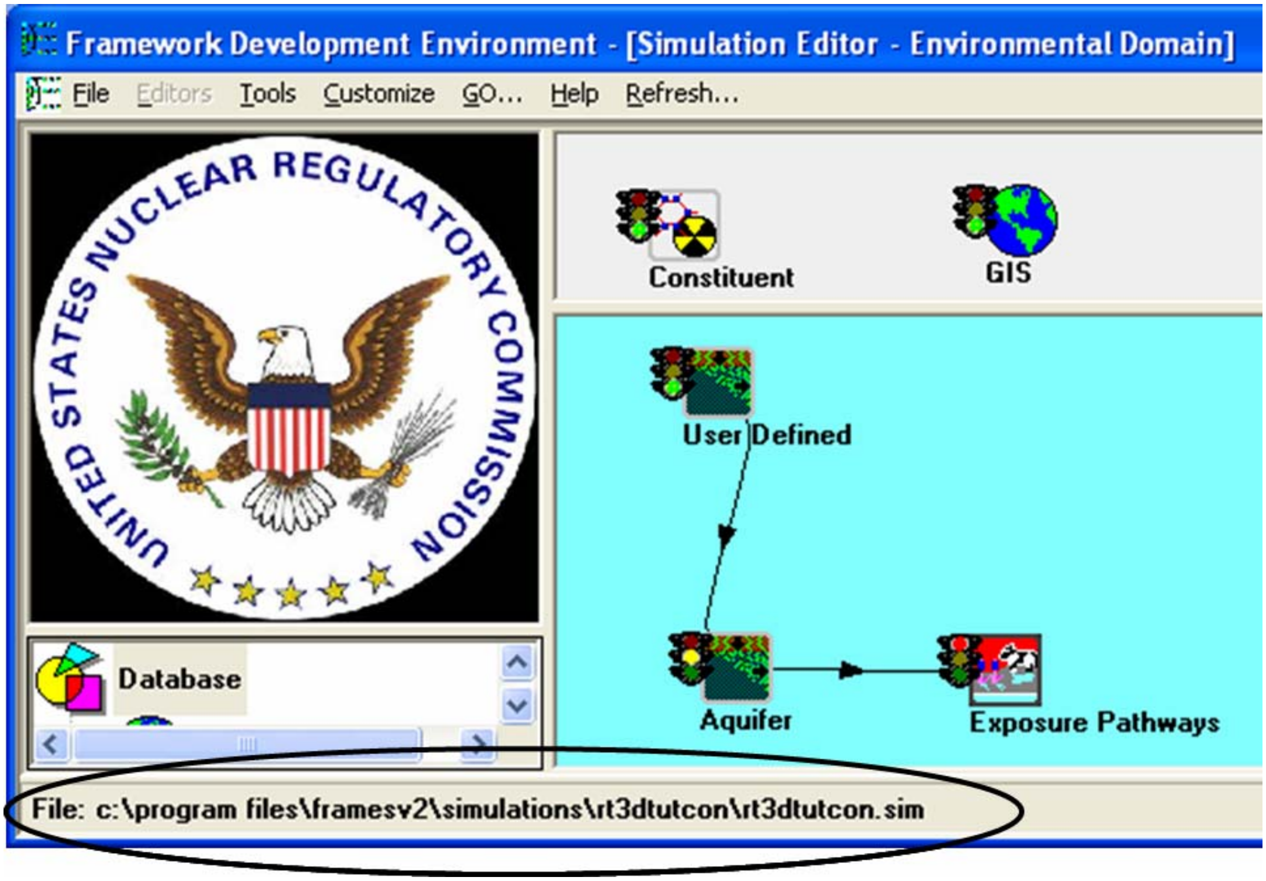
```
MODULE VERSION

MODULE DESCRIPTION
Import of C:/Program
Files/GMS50/tutfiles/rt3d/btexins1/sample/sample.gpr
on Fri Jun 22 07:56:41 PDT 2007

MODULE REFERENCES

SYSTEM REQUIREMENTS
Operating System:
Processor:
RAM:                         Minimum 0MB
DiskSpace:                   Minimum 0MB free disk space

POINT OF CONTACT
Company:
Address 1:
Address 2:
City:
Zip:
Country:
Contact:
Email:
Fax:
Phone:
URL:

EXECUTABLE INFORMATION
Tool:           False
Online:         False
Icon:           Icons\aqu.
UIExe:          DCE.EXE
UICmdLine:
ModelExe:       GMS\RT3DT
ModelCmdLine:
ModelURL:
```

The imported GMS case is automatically placed under the Aquifer icon.

"Model" name, corresponding to GMS Project File and name provided by the user in the GMSImporter.

Under Model Description, the user can find the folder location where the gpr file is archived.

Figure 3.12. FRAMES Module General Information User Interface, Where the Name of the GMS Module and its Associated Project File Are Listed


Framework Development Environment - [Simulation Editor - Environmental Domain]
File
Editors
Tools
Customize
GO...
Help
Refresh...
UNITED STATES NUCLEAR REGULATORY COMMISSION
Constituent
GIS
User Defined
Aquifer
Exposure Pathways
Database

File: c:\program files\framesv2\simulations\rt3dtutcon\rt3dtutcon.sim

Figure 3.13. Name and Location of the File After the User Chooses the Module in the General Information Screen

3.4 Pre-Processing Data for Input to Imported GMS Module

The output from an upstream-producing module is not expected to comply with the input requirements of the downstream-consuming module. Hence, FRAMES represents the middleware that provides seamless communication between these disparate components. Each module has it own unique input/output (I/O) requirements, and they do not have to be consistent with one another.

Each GMS module, such as RT3D, requires the input data to be formatted in a specific manner, which may involve multiple files. As such, any user-desired modifications to the input also require modifications to multiple GMS input files. When modifying input, the GMS Import tool needs to consider which parameters and files are being modified, how time and space are handled, and how multiple constituents are managed.

- Converting point values to step functions—Time-varying information within RT3D and MT3DMS are designed around stress periods. A stress period is a period of time where the properties throughout the simulation are constant. Hence, RT3D and MT3DMS assume time-varying input in the form of step functions. FRAMES makes no assumption with regard to the implied interpretation of time-varying data. FRAMES Environmental Domain dictionaries assume that a value at a point in time represents a point value. To confirm consistency between upstream models that produce data and downstream models, such as RT3D, which consume data and require time-varying data in the form of step functions, the GMS Import tool converts time-varying point values to step functions. Figure 3.14 provides illustrative examples on converting time-varying point values to step functions within FRAMES and GMS. This figure provides five input points that must be converted to four stress periods.

- The GMS Import tool averages the two adjoining points and assumes a constant value over the time increment. This method conserves mass over the entire curve.
- If the GMS user assumed an exponential release in the baseline case, RT3D does not average adjoining points but assumes that the initial point is constant until the beginning of the next stress period. As Figure 3.14 shows, this approximation creates mass. If the stress periods are small, then the error will be small.

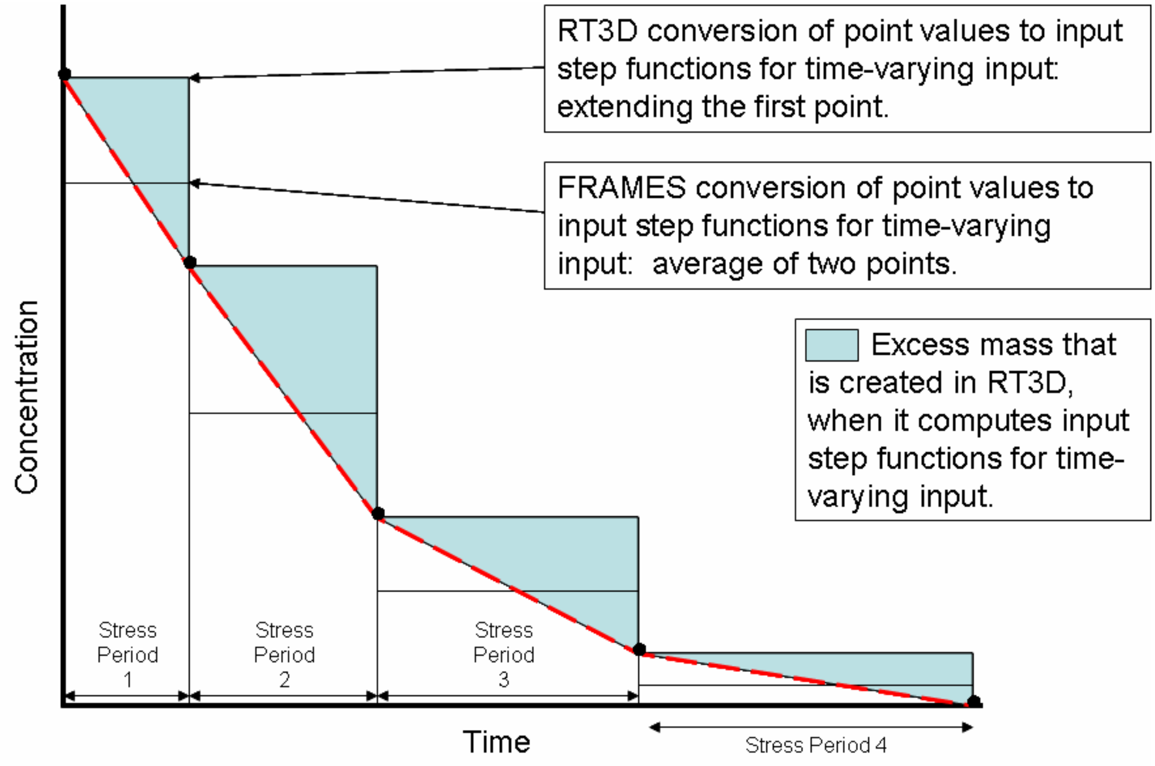
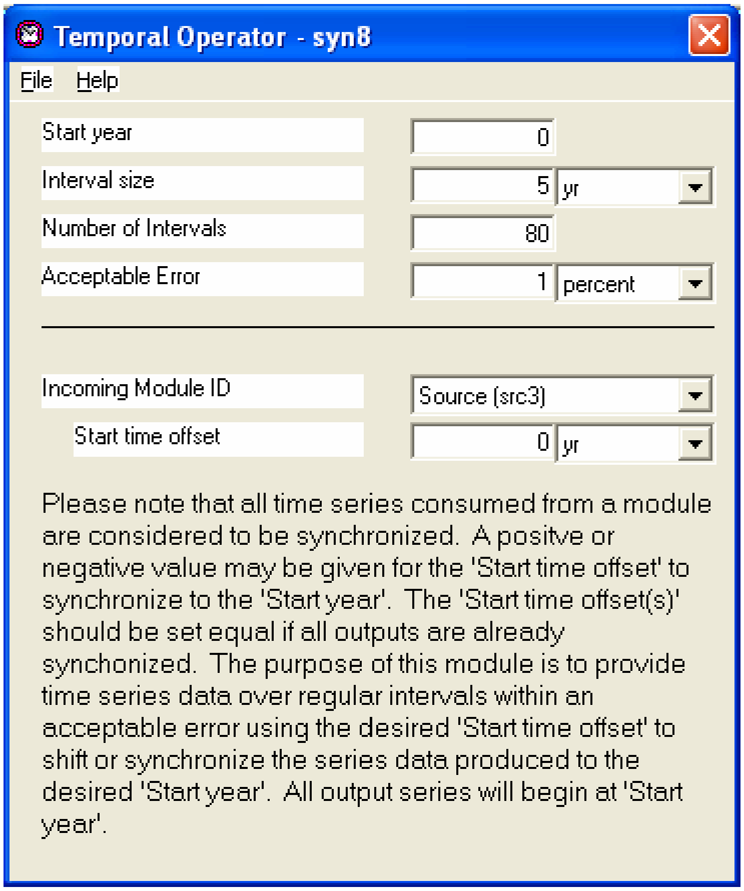

RT3D conversion of point values to input step functions for time-varying input: extending the first point.
FRAMES conversion of point values to input step functions for time-varying input: average of two points.
Excess mass that is created in RT3D, when it computes input step functions for time-varying input.
Concentration
Stress Period 1
Stress Period 2
Stress Period 3
Time
Stress Period 4

Figure 3.14. FRAMES and GMS Methods for Converting Time-Varying Point Values to Step Functions

- Cross correlating the number of stress periods between upstream and GMS modules—The stress periods are central to RT3D and MT3DMS and are holistically and systematically defined within these models. The hydrodynamic model, which is simulated independently (e.g., MODFLOW), defines its own stress periods, which also must match those in the transport model. The stress periods between the two models (e.g., source term and RT3D) have to be synchronized. RT3D and MT3DMS have strict rules on the number of stress periods per simulation. Once established, all components, including hydrodynamics (MODFLOW) and transport (RT3D and MT3DMS) must have the same number of stress periods. Problems arise when the number of input stress periods does not match the number of baseline stress periods. Either the output from the upstream model must be changed to comply with the baseline case, or the structure of the input to the baseline case must be modified to match the number of stress periods associated with the upstream model's output. Under the current linkage within FRAMES, it is assumed that the hydrodynamics are steady state and only changes from upstream boundary conditions will impact transport. With this assumption, the GMS Import tool reads the upstream models output, converts the point values to step functions (i.e., stress periods), determines the number of stress periods, and then modifies the appropriate files (e.g., SSM for concentrations and BTN for corresponding times) by substituting the upstream model's output for the baseline case input. The GMs Import tool must also make sure that the total time associated with the simulation is consistent within the system.

- Synchronizing time of multiple contaminants at different locations—The GMS Import tool converts point values to step functions and requires the synchronization of the number of stress periods between models. Many upstream models also produce time-varying output (e.g., concentrations) by contaminant. Because contaminants move through the environment at different speeds (e.g., chromatographic effect), different constituents can have these time-varying curves associated with them (e.g., different time steps and ending times). To be consistent with RT3D and MT3DMS, these curves must also be synchronized by stress period and ending time. A synchronization program has been written that synchronizes the time-varying output of multiple contaminants from an upstream model for input to RT3D and MT3DMS. This program is based on Whelan et al. (2006). The Synchronization Operator, implemented in VB (interface) and C++ (model), will confirm that the time-stepping protocol associated with each producing model's time-varying curves is consistently mapped, making sure that each time step is accounted for and assuming linear interpolation, when necessary. If the system API needs to combine the output from two producing model locations, and the time-step intervals associated with each output are different, the API will linearly interpolate to make sure that the time steps for both time-varying curves have values corresponding to, not only their own location's time-steps but also to the other location's time steps. For example, if location #1 provides output every 10 years, and location #2 provides output every 5 years, then values will be assigned to location #1, corresponding to every five years, using linear interpolation. The time series will then be integrated and checked for accuracy to see if the interval size is within the acceptable error specified. If not, an error message is written and execution stopped. Figure 3.15 presents a screen capture of the user interface for the Synchronization Operator corresponding to the CSM in Figure 3.16. The Synchronization Operator is utilized when
    - multiple modules require their outputs to be temporally synchronized.
    - there is too much output and a memory overflow occurs, requiring fewer time steps.

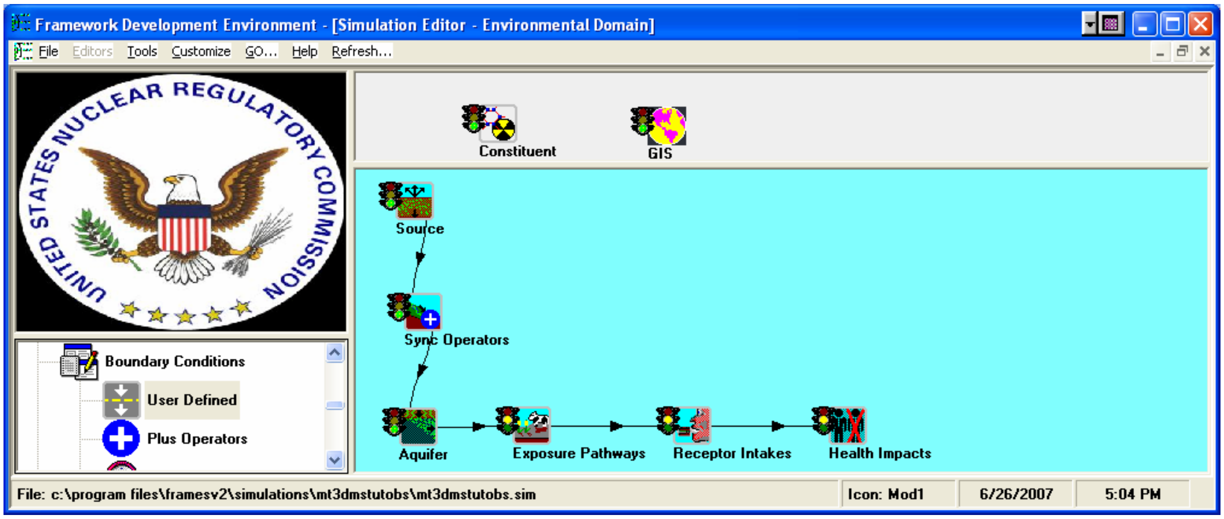
  The Synchronization Operator assumes that each set of output is temporally synchronized to the same starting time. For example, if output from model 1 begins at year 5 and the output from model 2 begins at year 50, then the Synchronization Operator assumes that the starting time for both models is year 0. The components that comprise the Synchronization Operator's user interface are described as follows (see Figure 3.15):
    - Start Year—Synchronized beginning associated with the data sets (e.g., 1985, year 0, etc.)
    - Interval Size—Time increment between values (e.g., stress period length).
    - Number of Intervals—Number of time increments, beginning with Start Year.



Start year	0	
Interval size	5	yr
Number of Intervals	80	
Acceptable Error	1	percent
Incoming Module ID	Source (src3)	
Start time offset	0	yr

Please note that all time series consumed from a module are considered to be synchronized. A positve or negative value may be given for the 'Start time offset' to synchronize to the 'Start year'. The 'Start time offset(s)' should be set equal if all outputs are already synchronized. The purpose of this module is to provide time series data over regular intervals within an acceptable error using the desired 'Start time offset' to shift or synchronize the series data produced to the desired 'Start year'. All output series will begin at 'Start year'.

Figure 3.15. User Interface for the Synchronization Operator



File: c:\program files\framesv2\simulations\mt3dmstutobs\mt3dmstutobs.sim | Icon: Mod1 | 6/26/2007 | 5:04 PM

Figure 3.16. Conceptual Site Model Corresponding to the Synchronization Operator Example

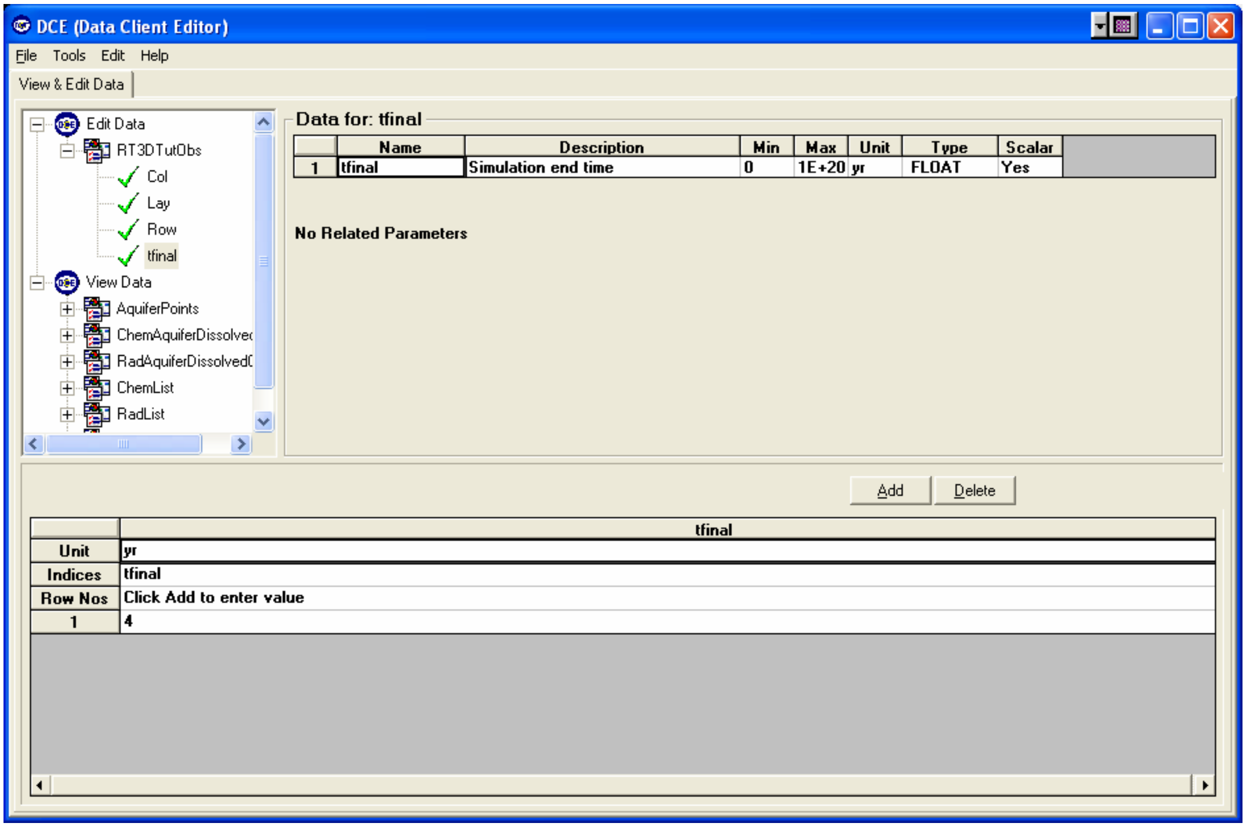
- Acceptable Error—Difference between original curve and the curve developed by the Synchronization Operator.
- Incoming Module ID—Module Identification, referring to the module linked to the Synchronization Operator.
- Start Time Offset—Time shift associated with a data set to synchronize multiple data sets to the same Start Year. For example, data set A begins its simulation in 1950, and its start time is represented by time zero, and data set B begins its simulation in 1960, and its start time is also represented by time zero. To synchronize the Start Year, the Start Time Offset for data sets A and B would be 0 and 10, respectively, indicating that year 1960 is 10 years a head of 1950 with a Start Year of 0 (or 1950) for both data sets.

3.5 Post-Processing Data as Output from GMS Module

Figures 3.3 and 3.4 illustrate how the GMS modules are integrated into the CSM and how they are defined within the user interface, respectively. Following the choice of the module, the user then supplies information related to the location and attributes of the output in the user interface. If the user did not supply the nodal locations for capturing the output from the GMS run in terms of Row, Column, and Layer in the GMS Import tool (see green section, last text box of Figure 3.9), the output location would have to be supplied by the user in the Data Client Editor (DCE), which is the module user interface, as illustrated by Figures 3.17 and 3.18. The only limitation on processing data and converting output from the GMS results is that only 1000 time points are allowed to be passed on to the downstream module from the GMS module.

Figure 3.17 illustrates the user interface when the user chooses to access GMS output data from the OBS file. Because the GMS model does not provide any indication of when interesting parts of the time-varying output might occur, “tfinal” allows the user to define the ending time of the GMS simulation, thus providing some control over the output. Under the OBS option, the user has no control over how many simulation output points are between zero and tfinal; this is determined by the RT3D or MT3DMS models. Because of the 1000-point limitation, the GMs Import tool post processes the data in two ways in the following order:

1. Many times the results consist mainly of zeros at the initial and tail ends of the time-varying output. The GMS Import tool post processes the results, determines where the interesting part of the curve is, and eliminates unnecessary zeros on both ends of the curve. This is done to make sure that most, if not all, of the 1000 points describe something interesting. Figure 3.19 presents an example OBS file containing extra zeros in the initial part of the curve.
2. If less than 1000 data points remain, the entire curve is processed and available for consumption by downstream models. If more than 1000 points still remain, then points are skipped. For example, if 1500 points remain, then every other point (i.e., skip one point: 1500/1000 = 1 in integer form) is processed. If 5300 points remain, then every sixth point (i.e., skip five points: 5300/1000 = 5 in integer form) is processed. The number of points passed to the next module is always between 1 and 1000, inclusive.



Data for: tfinal

	Name	Description	Min	Max	Unit	Type	Scalar
1	tfinal	Simulation end time	0	1E+20	yr	FLOAT	Yes

No Related Parameters

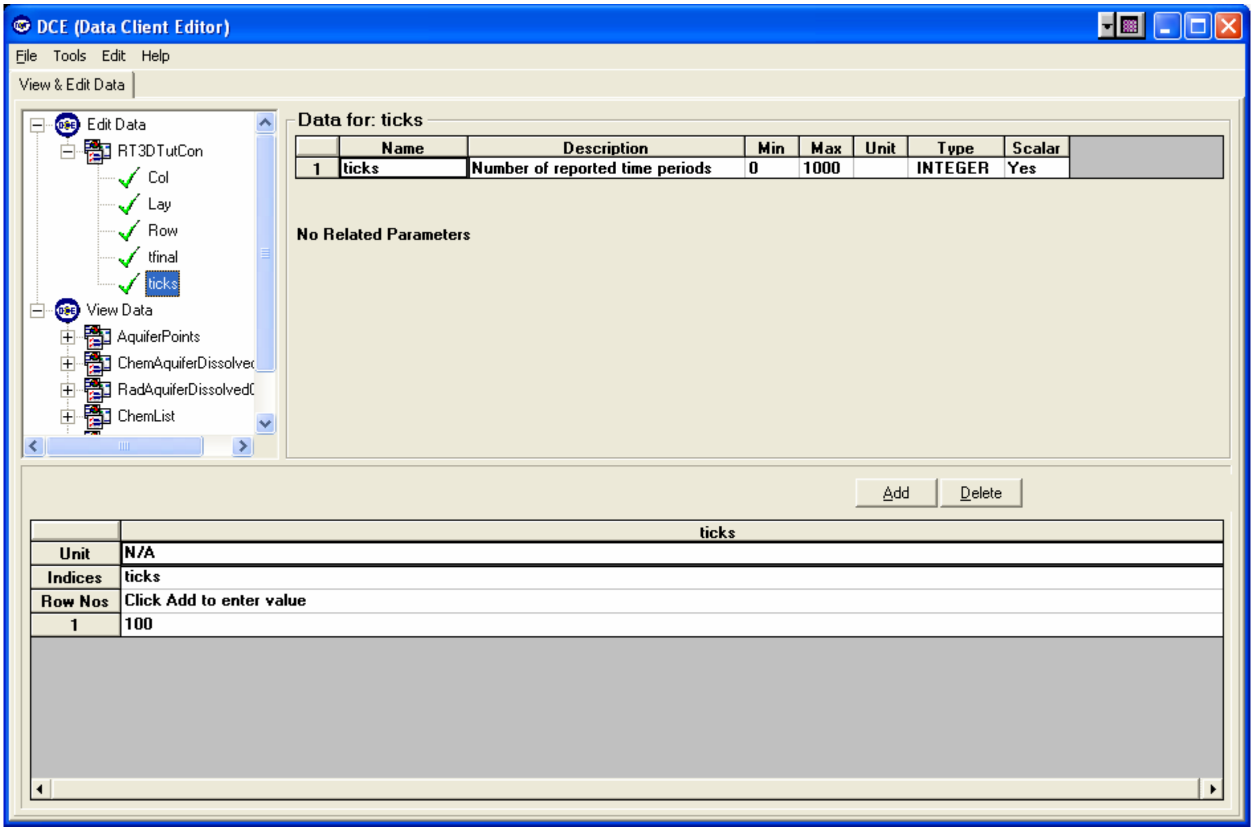
	tfinal
Unit	yr
Indices	tfinal
Row Nos	Click Add to enter value
1	4

Figure 3.17. The DCE Getting Output Data from a FRAMES 2.0 User when the OBS File Option Was Chosen in the GMS Import Tool

An example set of post-processed OBS file output is presented in Figure 3.20.

Figure 3.18 illustrates the user interface, when the user chooses to access GMS output data from the CON file. The CON file option will allow the user to

1. identify specific times associated with the output (e.g., years 73, 146, 219, … 730). The user can only select specific output times when they were set up as part of the baseline case in the original stand-alone GMS application, prior to linkage within FRAMES. These times are established *a priori* in the BTN file, which is illustrated in Figure 3.21.
2. identify an end time to the simulation (i.e., "tfinal" in Figure 3.18). This parameter operates in the same manner as tfinal under the OBS file option.
3. identify the number of reported time periods or intermediate points (i.e., "ticks" in Figure 3.18) between time zero and tfinal. Unlike the OBS file option, under the CON file option, the user can select the number of output time periods, but all extraneous zeros in the time-varying curve remain. This option can be onerous, as illustrated by Figure 3.22. Figures 3.22a and 3.22b present the same results but with different ending times. As this figure shows, the number of extraneous zeros can significantly influence the output results that are passed along to downstream models. Care should be used when using the CON option.



Data for: ticks

	Name	Description	Min	Max	Unit	Type	Scalar
1	ticks	Number of reported time periods	0	1000		INTEGER	Yes

No Related Parameters

	ticks
Unit	N/A
Indices	ticks
Row Nos	Click Add to enter value
1	100

Figure 3.18. The DCE Getting Output Data from a FRAMES 2.0 User when the CON File Option Was Chosen in the GMS Import Tool

3.6 Area Source Definitions: Consistency Between FRAMES and GMS

When high-fidelity models are involved in a simulation exercise, multiple source terms may be associated with multiple locations. These source terms could be defined in any number of ways (e.g., size, constituent makeup, geographic location, waste process that produced them [e.g., injection wells, French drains, trenches, spills]). Thousands of nodes or cells may be involved in each designated source area. It would be extremely cumbersome for the FRAMES user to have to input single entries for each node or cell, as it is not convenient for the user to individually identify and populate each node impacted by an area source. As an initial application of the linkage between FRAMES and GMS, these area sources will be assumed to be continuous (i.e., connected), and input concentrations are assumed to be uniformly distributed across the area source, so one concentration can be assigned to represent each node or cell in the area source. If they are next to each other and have the same concentration, the user will be asked to provide one concentration from FRAMES for that set of connected points. If users want to spatially redistribute different concentration levels at multiple nodes in the area source, then they should be using GMS, not linkages to FRAMES. In summary, if the source-term (or boundary-condition) area is spatially contiguous and a uniform distribution of the concentration represents source-term model output, then FRAMES can be used to link legacy, disparate, source-term models to RT3D or MT3DMS. If the source term is more complex (e.g., multiple, unconnected source areas or spatially varying contaminant levels), then FRAMES can still be linked to RT3D or MT3DMS, but it would be best if the source term is described from within GMS.

3.7 Reusing Legacy File Formats in FRAMES 2.0

Many legacy environmental models are written in the FORTRAN programming language and assume that all input is in fixed-width formatted ASCII input files or delimited ASCII (also known as free format). Sometimes legacy models read and write data into formatted binary files. The task of creating the marked-up file formats can be daunting, and for a given model in GMS, a tool that generates marked-up files is a much better choice than having the GMS user create text files that describe the input and output files. As such, the GMS Import tool has been developed to generate this information based on a few clear choices by the user.

To connect to any of these data storage and retrieval approaches, a need arises to read and write data in these legacy file formats. While there is not usually an Application Programming Interface (API) or library that facilitates reading and writing these file formats, the legacy file formats are almost universally described in a document. This document includes the conditionality of the data entries as well as the scientific meaning of the entries. Therefore, the data file itself communicates this meaning poorly. Listings 1 and 2 are a fixed-width and delimited version, respectively, of an example input/output file.

Listing 1: Fixed Width Example

```
Site: MySite
1.65    .10
Number of Constituents: 2
sr90      100     1 y90       100
71-43-2    50     0
40
```

Listing 2: Delimited Example

```
“Site:”,”OtherSite”
2.00,.12
“Number of Constituents:”,4
“sr90”,100,1,y90,100
“71-43-2”,100,0
“sr90”,25,2,y90,100,S89,10
“7443-76-2”, 12,0
100
```

These two example listings would have a “card deck” definition similar to the following. The name “Card Deck” comes from the fact that some of these models where created when each line of an input file was represented by a single IBM punch card.

Card Deck Definition for Example Files

Format	Range	Description
Card 1:		
Once for every file		
A5		“Site:”
A20		Name of the site
Card 2:		
Once for every file		
F8.3	1.65-2.65	Bulk density of the site in g/cm^3
F8.3	0.1-.24	Moisture content of the site in fraction of volume
Card 3:		
Once for every file		
A23		“Number of Constituents:”
I2	1-99	Number of constituents
Card 4:		
This card occurs for every constituent in Card 3.		
A10		Chemical Abstract Service (CAS) Number (with dashes) or Radionuclide ID
F8.3		Mass for constituent (kg)
I2		Number of progeny
A10		Progeny CAS or Radionuclide ID. Repeated for each progeny.
F8.3		Mass of progeny (kg). Repeated for each progeny.
Card 5:		
Once for every file		
I3		Number of times to report

This is a simplified example of what the input/output file format could look like. The solution will allow us to capture the information in the “Card Deck” definition into an algorithm that can read or write those data to a FRAMES 2.0 data set.

The simplest solution was to modify the behavior of the BPP. This tool allows you to mark up a file with Java code that is eventually interpreted by the BeanShell (http://www.beanshell.org). BPP 0.8.4 already had the capability of changing the behavior of any stage in the number of stages of interpretation. So, all that was written in a final stage that can either read a model’s specific file and write it to a FRAMES 2.0 data set, or read a FRAMES 2.0 data set and write the model-specific file.

While BPP is processing a file, when it encounters the “$” character, the text between the parentheses that follow is passed to a function in a class. The term the BPP author uses is that the “$” indicates that some “magic” should occur, but this simply means the text that should be replaced by some other text. Text that is not marked with “$” is processed by a separate method in the same class. Fundamentally, all that needs to happen is to have marked text interpreted as reading a value from the file into a FRAMES data set or vice versa.

The following example listing of a “marked-up” version, correlated to the example file described above, might make this process more clear. The # notation for code, mixed with an example file, needs to be introduced so that flow control (loops) can be added to the marked-up file. Even this reasonably simple case has the marked-up file more closely related to source code than a marked-up input file.

Listing 3: Example “marked-up” Version of File

```
Line
1 Site:$(sitename,20)
2 $(bulkd,8,”g/cm^3”)$(mc,8,”fraction”)
3 Number of Constituents:$(count,2)
4 #for (int i=0;i<count;i++) {
5 $(cas[i],10)$(mass[i],8,”kg”)$(numprog[i],2)$()
6 #  for(int j=0;j<numprog[i];j++) {
7 $(prog[i],10)$(progmass[i],8,”kg”)$()
8 #  }
9
10 #}
11 $(numtimes,3)
```

1. Line 1 is the first line of the original file, but the site-name location is marked with “magic.” The variable name and the width are given as additional information.
2. Line 2 is a marked-up version of the input file with meta-data about the bulk-density(bulkd) and moisture content(mc). Notice that units are part of this meta-data.
3. Line 3 is the definition of the line with number of constituents.
4. Line 4 is the first line that captures some of the algorithm that is in the “Card Deck” definition. This line contains the Java syntax of the loop that is implied by the card-deck definition. This would be equivalent to a “do I=0 to count-1.”
5. Line 5 is Card 4 marked-up until the progeny information is supposed to be defined. This line also contains a special use of magic. If there is not text in a magic statement (i.e., “$()”), this tells the interpreter to stay on the current line. This is nearly identical to the “\” use in FORTRAN.
6. Line 6 is the loop implied still on the definition of Card 4.
7. Line 7 is the progeny Chemical Abstracts Service (CAS) and mass information also on Card 4. Again, “stay on current line” is used.
8. Line 8 is the end of the loop for progeny. The FORTRAN “end do” is for Line 8.
9. Line 9 is the end of line that will occur at the end of the progeny. This empty line will basically force an “end of line” to occur.
10. Line 10 ends the loop for each constituent. So Card 4 implied the need for lines 4 to 10. A single card defined two loops.

11. Line 11 is a simple marked-up line to read/write numtimes.

Currently, the interpretation of the marked-up file is given below. By default, the BPP not only interprets, but also executes, the resulting code. This generation of the interpreted file was done to illustrate the process. The interpretation currently has one of four options: 1) Read Fixed Width file, 2) Read Delimited file, 3) Write Fixed Width file, and 4) Write Delimited file. Currently, these interpretations would make the variables available to the Java programming language. To make the Java programming language ready for the script, the variable types have to be defined before the file is read. To do this requires adding such a line as “# int count;”. This is not done in the example for clarity and also because when the FRAMES data sets (described later) are used, only one line will need to be added to the top to accomplish the declaration and space allocation.

Listing 4: Example Output

```
BIPScanner in=(BIPScanner)new BIPFixedWidth();
in.open(bsh.args[0]);
 {in.match(“Site:”);sitename=in.read(sitename,20);in.nextLine();}
 {bulkd=in.read(bulkd,8,”g/cm^3”);mc=in.read(mc,8,”fraction”);in.nextLine();}
 {in.match(“Number of Constituents:”);count=in.read(count,2);in.nextLine();}
for (int i=0;i<count;i++) {

{cas[i]=in.read(cas[i],10);mass[i]=in.read(mass[i],8,”kg”);numprog[i]=in.read(numprog[i],2);
}
  for(int j=0;j<numprog[i];j++) {
 {prog[i]=in.read(prog[i],10);progmass[i]=in.read(progmass[i],8,”kg”);}
  }
in.match(““); in.nextline();
}
 {numtimes=in.read(numtimes,3);in.nextLine();}
in.close();
```

From the example output, the text forms of the file are read and checked to see if a match between the parts within the expected string has occurred. If a match is not found, then an error is produced with a description of the error. The first two lines and the last are added automatically by the BPP interpretation. Each line with $ (known as magic by the bpp developer) is broken into pieces that are either matched or read into Java variables. The declaration of the Java variables is not shown here though.

The set of four interpreters that exist is the 1) Read Fixed Width and Write FRAMES data set, 2) Read Delimited and Write FRAMES data set, 3) Read FRAMES data set and Write Fixed Width, and 4) Read FRAMES data set and Write Delimited.

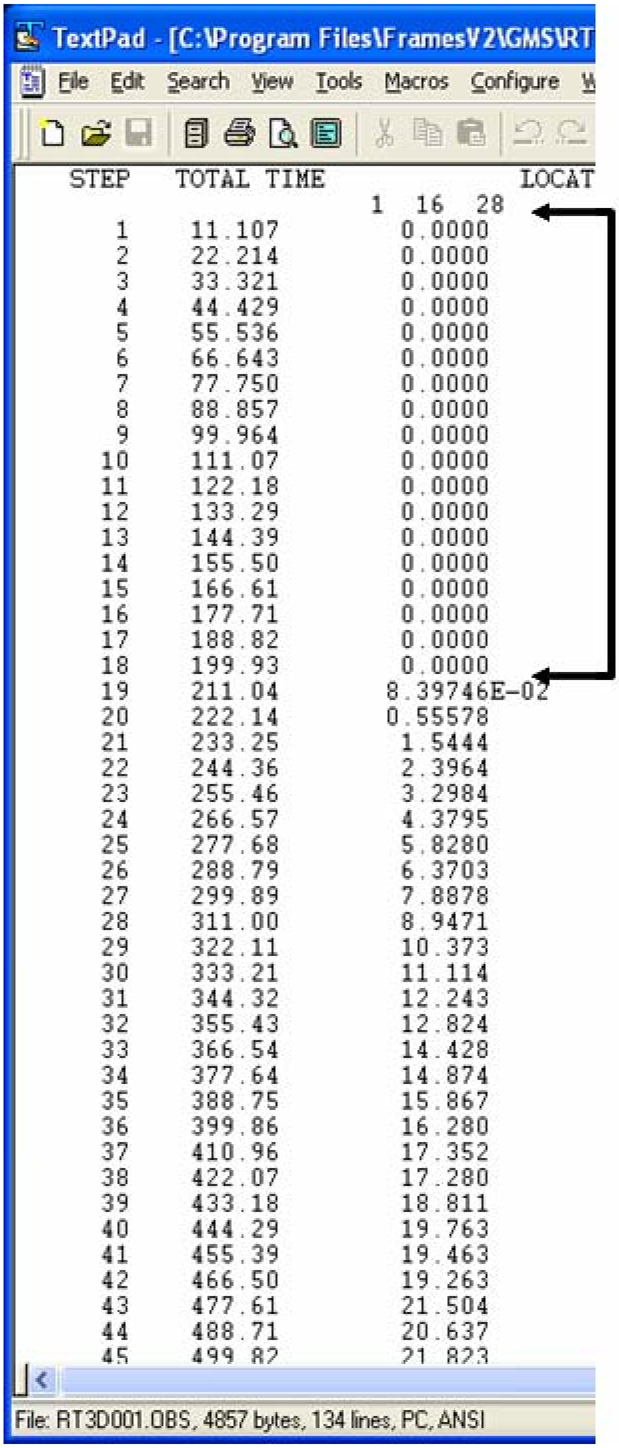
3.8 Source-Code-Only Approach

The marked-up approach, as described in Section 3.2, works very well for changing only a few values in the RT3D or MT3DMS file. As the file complexity increases, the marked-up file begins to be more source code than text, so it is cleaner to simply write source code. The source code is compiled and tested before its actual use, so syntax and logic errors can be found during construction of the pre- and post-processors instead of having to wait for error messages from the processing sequence.

The best example of a format from RT3D and MT3DMS that causes so much complexity in the mark up is the “RARRAY.” This data structure has a number of format options. Each of the format options changes the number of lines of data that follow. In format 100, a two-dimensional set of floating-point numbers is read. The RARRAY description is in the MT3DMS Manual (MT3DMS: A Modular Three-Dimensional Multispecies Transport Model for Simulation of Advection, Dispersion, and Chemical Reactions of Contaminants in Groundwater Systems; Documentation and User’s Guide) Section 6.0.

This RARRAY data structure is used for higher dimensional values as well. So, an RARRAY parameter that is determined by species and stress period would have logic embedded in the loops for species and stress period. At the very least, the import utility would have to contain the logic of the RARRAY format to be capable of accurately changing a value that followed any RARRAY. So, being able to read this structure is required even if the FRAMES GMS Import tool were going to be restricted to changing only single floating point numbers.

The solution is to emulate the model developer’s behavior but in a way that facilitates both reading and writing the file format. Also, reading the file in one format and writing another format should be possible. This would provide many of the advantages of the BPP approach but reduce some limitations. The key concept is to define a file format that is independent of reading or writing. A simple example is illustrated in Listing 5.



```
    STEP   TOTAL TIME          LOCAT
                        1  16  28
      1     11.107        0.0000
      2     22.214        0.0000
      3     33.321        0.0000
      4     44.429        0.0000
      5     55.536        0.0000
      6     66.643        0.0000
      7     77.750        0.0000
      8     88.857        0.0000
      9     99.964        0.0000
     10     111.07        0.0000
     11     122.18        0.0000
     12     133.29        0.0000
     13     144.39        0.0000
     14     155.50        0.0000
     15     166.61        0.0000
     16     177.71        0.0000
     17     188.82        0.0000
     18     199.93        0.0000
     19     211.04       8.39746E-02
     20     222.14       0.55578
     21     233.25        1.5444
     22     244.36        2.3964
     23     255.46        3.2984
     24     266.57        4.3795
     25     277.68        5.8280
     26     288.79        6.3703
     27     299.89        7.8878
     28     311.00        8.9471
     29     322.11        10.373
     30     333.21        11.114
     31     344.32        12.243
     32     355.43        12.824
     33     366.54        14.428
     34     377.64        14.874
     35     388.75        15.867
     36     399.86        16.280
     37     410.96        17.352
     38     422.07        17.280
     39     433.18        18.811
     40     444.29        19.763
     41     455.39        19.463
     42     466.50        19.263
     43     477.61        21.504
     44     488.71        20.637
     45     499.82        21.823
```

File: RT3D001.OBS, 4857 bytes, 134 lines, PC, ANSI

Extraneous zeros are removed in the initial and tail ends of the curve. This option only exists when reading the OBS file.

Figure 3.19. An Example OBS File Containing Extra Zeros in the Initial Part of the Curve

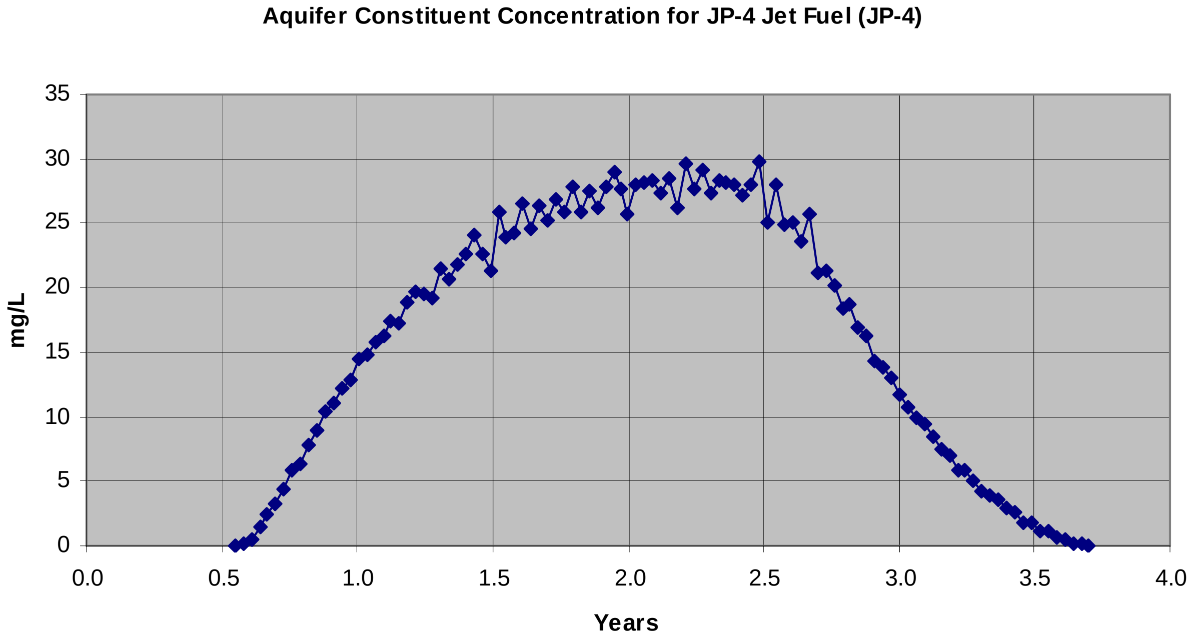
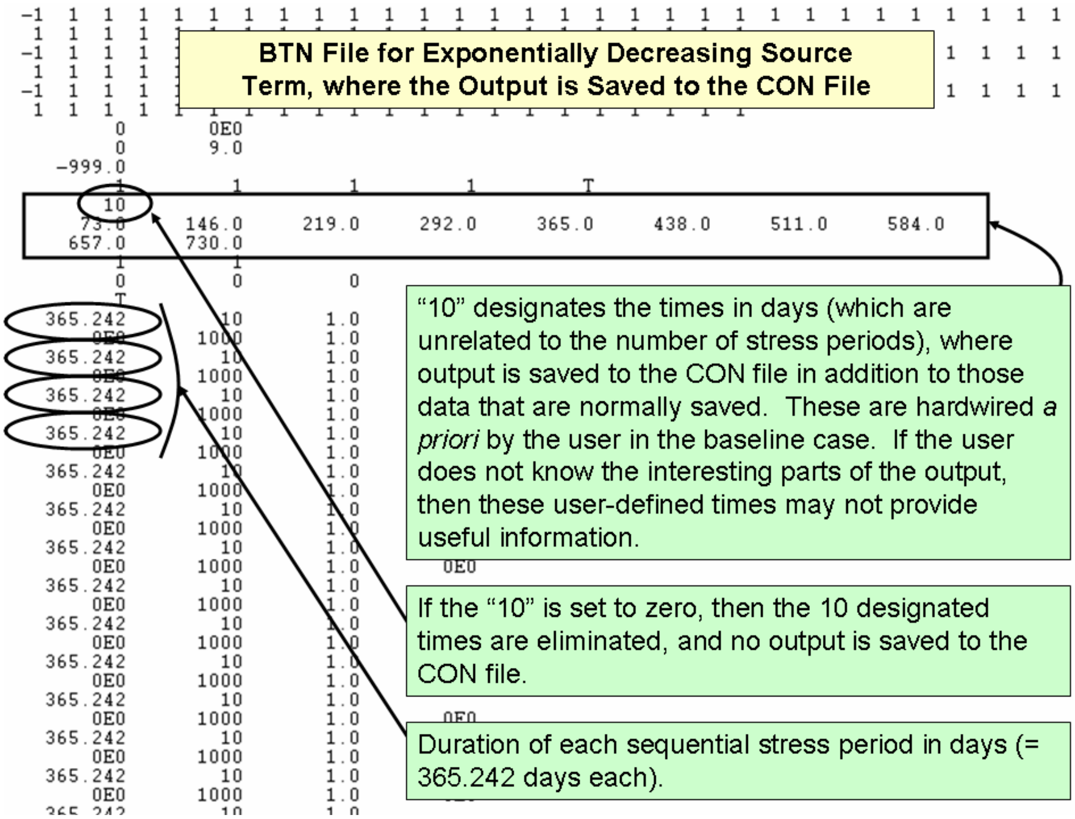


Figure 3.20. An Example Set of Post-Processed OBS File Output


```
-1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
 1   1   1   1                                                                                     
-1   1   1   1                                                                                      1   1   1   1
 1   1   1   1                                                                                     
-1   1   1   1                                                                                      1   1   1   1
 1   1   1   1
         0          0E0
         0          9.0
    -999.0
         1            1            1            1            T
        10
      73.0        146.0        219.0        292.0        365.0        438.0        511.0        584.0
     657.0        730.0
         1            1
         0            0            0
         T
   365.242           10          1.0
       0E0         1000          1.0
   365.242           10          1.0
       0E0         1000          1.0
   365.242           10          1.0
       0E0         1000          1.0
   365.242           10          1.0
       0E0         1000          1.0
   365.242           10          1.0
       0E0         1000          1.0
   365.242           10          1.0
       0E0         1000          1.0
   365.242           10          1.0
       0E0         1000          1.0
   365.242           10          1.0
       0E0         1000          1.0          0E0
   365.242           10          1.0
       0E0         1000          1.0
   365.242           10          1.0
       0E0         1000          1.0
   365.242           10          1.0
       0E0         1000          1.0          0E0
   365.242           10          1.0
       0E0         1000          1.0
   365.242           10          1.0
       0E0         1000          1.0
   365.242           10          1.0
```

BTN File for Exponentially Decreasing Source Term, where the Output is Saved to the CON File

"10" designates the times in days (which are unrelated to the number of stress periods), where output is saved to the CON file in addition to those data that are normally saved. These are hardwired a *priori* by the user in the baseline case. If the user does not know the interesting parts of the output, then these user-defined times may not provide useful information.

If the "10" is set to zero, then the 10 designated times are eliminated, and no output is saved to the CON file.

Duration of each sequential stress period in days (= 365.242 days each).

Figure 3.21. User-Selected Output Times, Which Were Pre-Determined Prior to Linking the RT3D or MT3DMS Baseline Case to FRAMES

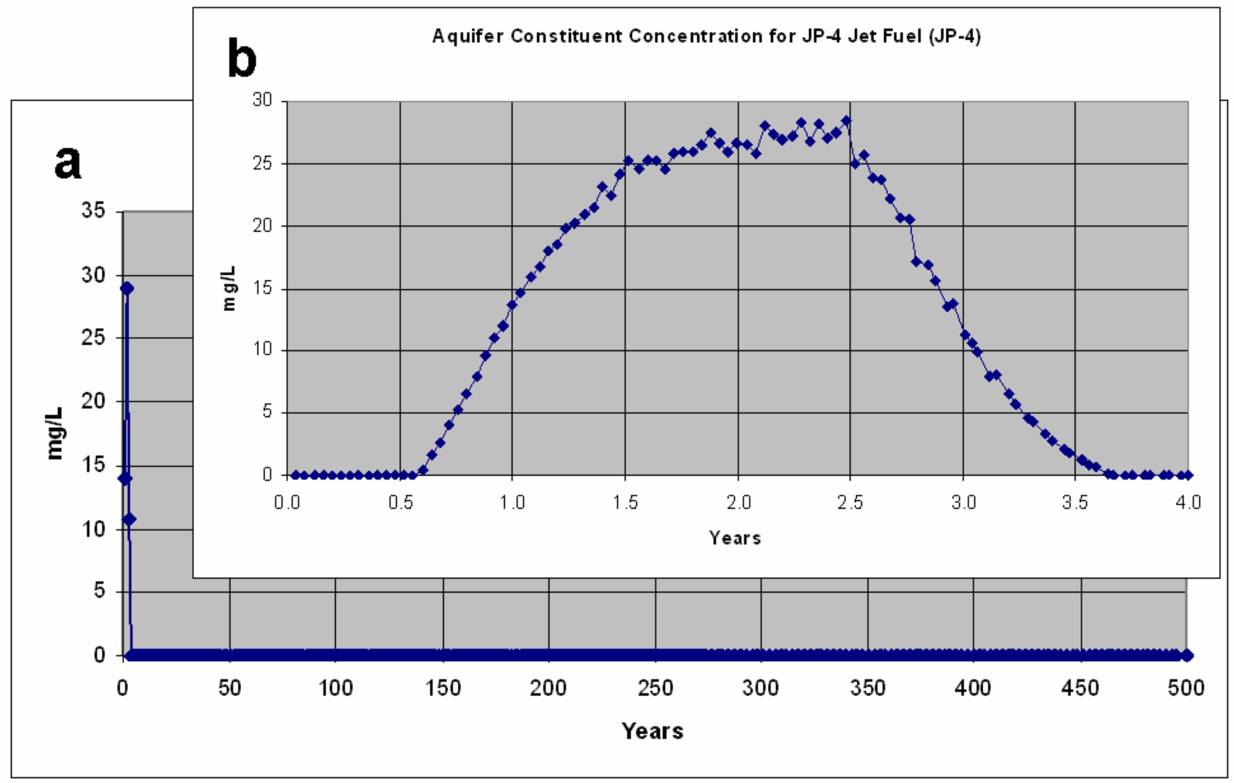


Figure 3.22. Example Illustrating how the same CON Results with a Different "tfinal" Can Impact the Output Results Passed to Downstream Models: a) tfinal = 500 years, b) tfinal = 4 years

In essence, the # notation is missing, and any $ has been replaced by a t.operate function call. When the reader/writer should move to the next line, it is explicit with t.nextline. Finally, values that had no markup are made into strings and passed to t.constant. The object "t" is a file reader or writer that implements the operate function for the integer, real, logical, and string data types, but it can be extended to include model-specific data types such as RARRAY. The "t" object also needs to implement nextline and constant. For readers, nextline should move the file position to the next line, while writers should add a new line to the output. The constant function checks that the value is present during reading and writes the constant value out during writing. In summary, a file operator (like object t) needs to implement the following functions as noted under Listing 5:

Listing 5. Pure Source Code Example

```
t.constant(“Site:”); t.operate(sitename,20); t.nextline();
t.operate(bulkd,8,”g/cm^3”); t.operate(mc,8,”fraction”); t.nextline();
t.constant(“Number of Constituents:”); t.operate(count,2); t.nextline();
for (int i=0;i<count;i++) {
t.operate(cas[i],10); t.operate(mass[i],8,”kg”); t.operate(numprog[i],2)
  for(int j=0;j<numprog[i];j++) {
    t.operate(prog[i],10); t.operate(progmass[i],8,”kg”);
   }
 t.nextline();
 }
t.operate(NTP,3);
```

Writing a subroutine that uses this set of functions can be switched from reading to writing by simply changing the object “t” from a reader to a writer. This is identical to the mechanism underlying the BPP process, but it is compiled. A simple test to check the file format is to use the subroutine to read the data from the original file and then use the writer to write a copy of the file to a different location. If the original file and the generated file match, then the syntax and logic of the reader is correct for that case at least. After running this process on a number of example cases, confidence can be built that the subroutine that contains the format algorithm is correct.

3.9 Allocation of Memory and Dataset Tags

The application of complex source terms with a more rigorous and robust problem is not atypical of what one encounters when using a high-fidelity model. Relatively simple CSMs do not challenge run times, memory allocation, or storage. For example, simple CSMs can read all data into memory, so one can quickly gain access to the data and use it in any analysis. The problem with reading all data into memory is that there may eventually be too much data. A mixture of two solutions was implemented to address the problem:

- Read only the data you need, so you do not need to allocate as much. This is mainly for the ssm files and may save 50% allocation. This is an optimizing of memory allocation. Results of this analysis indicated that this solution will not consistently provide the necessary robustness. This occurs when all the options are turned on, and then the memory allocation is the same as with the unoptimized allocation.
- An optimized allocation will not work on the contaminant (con) files. Because there may be tens of thousands of nodes (and maybe more), only storing the subset of locations was required. The exposure well location is read before the contamination files are read. Only the concentrations at the exposure locations are stored.

3.10 Conclusions on Design

Two distinct methods have been developed to make models operate within FRAMES 2.0, such as the models in GMS. The first method is quick and simple, taking the input and output files for a model and then adding notation that allows them to be read or written. This approach works well for specific file cases that need to be only slightly modified for use within FRAMES 2.0 or other modeling systems. As the modifications become more complex, the simplicity of this approach begins to make it harder than it needs to be. In these more complex cases, the full source-code approach always works well. The source code approach must abstract away the operation of reading and writing from the file format itself; if it does not, the effort required to confirm that both the reading and writing has correct logic is more than doubled. This is even more evident if different standards (delimited versus fixed-width formatting) are needed for each reader and writer.

3.11 Limitations and Issues

The GMS Import tool currently imports only RT3D and MT3DMS cases. The project and input and output files can be in any location. The resulting component requires that the RT3D or MT3DMS executable be placed on the computer. For RT3D, this includes RT3D.exe (Version 2.5) and RXNS.dll. For MT3D, this includes MT3DMS4.exe (Version 4.0). These files are included with the latest install of FRAMES.

4.0 Specifications

A specification is a defined boundary between software systems. Typical boundaries between systems include files and associated formats; in other cases, a specification may be a defined protocol that two machines will use to communicate. Often these protocols and file formats are not defined (or controlled) by the software being written but are required by the partnering software. The GMS Import tool is a good example of other systems dictating (GMS in this case) the boundaries that a FRAMES 2.0 module must meet. GMS is on one side, and FRAMES is on the other side. In between, the Beanshell and BPP tools are used to facilitate modification of input files. Given this situation, the specifications listed below are divided into three categories:

4.1 Specifications that Are Defined by GMS

Explicit documentation of these file formats is available with the RT3D (http://bioprocess.pnl.gov/rt3d_hist.htm) model for GMS. In addition to the RT3D documentation, it was found during the development of this component that the documentation for the MT3DMS was useful for a number of the given file formats (http://www.emrl.byu.edu/gms.htm). The file formats used by RT3D can be found in the documentation for that model. This is part of a standard GMS 5.0 installation. There are no new specifications developed for this tool.

- GMS Project file (.gpr) read by the GMS Import tool to provide the user appropriate choices of importing the case
- Basic Transport (.btn) read by created module to get values associated with the dimensions of the result
- Sink/Source (.ssm) modified by the created module to change the boundary concentrations at the injection site
- Dispersion (.dsp) modified by the created module to change the dispersion properties of the simulation
- Reaction (.rct) modified by the create module to change the reaction rates
- Concentration file (.con) read by the created module after the RT3D or MT3DMS module is run.

4.2 Specifications Defined by Beanshell/BPP

Phase 1 of the GMS Import tool required the use of the Beanshell/BPP, but this is no longer the case. The specifications of Beanshell (http://www.beanshell.org/) and BPP (http://bpp.sourceforge.net/) can be found at their respective web sites.

4.3 Specifications Defined by FRAMES 2.0

FRAMES 2.0 specifications are both an API and particular dataset types. The primary API used is the FRAMES 2.0 Module Developer API. Documentation for this API can be found in the developer subdirectory of any FRAMES 2.0 installation. The datasets specific to providing input to and consuming output from RT3D are created using the following dictionaries:

- WaterConcPoints—locations where water concentrations have been calculated both for input to and output from the RT3D module
- ChemWaterConc—calculated dissolved-water concentrations
- WaterFluxPolys—locations where water passes from one layer (modeling domain) into another
- ChemFluxTotal—total mass moving from one layer (modeling domain) into another.

4.4 Compiler Version

The source code that makes up the GMS Import tool is compiled with Java 1.4 or later.

4.5 Java Runtime Environment Version

For the FRAMES 2.0 module to run correctly, the following must be true of the system:

1. The gms.jar must be available to GMSImport.bat.
2. The models for GMS must be located on the hard drive or networked drive connected to the user's machine.

5.0 Testing Approach and Results

The test procedure summary includes as a minimum the date, name of tester, and the version. The target requirements should be clearly identified. Other information that is helpful and may be appropriate to add to the test procedure summary is start time, end time, and location of the test. If there are any regulations or change-request documentation related to the test, that information should also be noted in the summary of the test procedure.

Pass/Fail Test Criteria

Test criteria should be defined before testing. A standard set of nomenclature needs to be identified and documented. The standardization explanation needs to be archived with the test procedures so that future assessors can fully understand the measurement system used for the test. An explanation of what success means and what failure means is critical for the defensiveness of the test.

Test-Procedure Components:

A log of the test should be documented containing information on the steps needed to demonstrate the requirement, expected results, and requirement-specific criteria to be used to determine a pass or fail finding. Related comments and documentation should also be captured. It is very valuable to identify the documents used to support the understanding and expectation of the requirement (e.g., statement of work, user's guide, help file). This identification of related documentation helps to tie together the quality documentation package for future readers of the procedure.

When designing the flow of a testing procedure, there are several types of requirements that can be grouped to facilitate testing. Some standard categories are functional requirements, data flow, calculations, output checks, and user-scenario cases.

1. Functional Requirements—Based on the client requirements describing what functions exist in each component of the system.
2. Data Flow—Flow of information between users, database tables, modules, displays, and the output produced.
3. Calculations—Any processing of module/display values. This includes unit conversions. Any data transformation should be documented and tested for the system.
4. Output Check—Confirmation of the correct generation of the output products.
5. Additional Scenarios—Scenarios that are developed to provide some holistic assessment of cases that may be possible and relevant to the expected use of the product.

Other information, such as prerequisite requirements, inputs, and assumptions, can also be collected and should be documented if they exist.

Other supporting tools should be considered where appropriate. Tools such as change-tracking systems and automated testing tools are helpful in sustaining a quality testing process.

All tests assume that the FramesV2 and GMS5.0 has been installed and that the tutorial case files exist. The latest FramesV2 installation can be found on COLAB at https://colab.sc.egov.usda.gov/cb/displayDocument/FramesV2Setup.exe?doc_id=2744.

5.1 RT3D Test Case #1: Using the CON File

Description: RT3D Test Case #1 tests the GMS Import tool as it relates to all of the requirements for the RT3D model within GMS. It does this by importing a GMS case titled sample.gpr and naming it RT3DTutCon.

5.1.1 Input Data

The input data for this case is entered by the user in the GMS Import tool and in the DCE for the RT3D aquifer module. The specific entries associated with each step are provided in the section titled "Conducting the Test." It is assumed that the source-term model provides constant output at 1000 mg/L to the RT3D model for two years. Figure 5.1 presents an illustrative graphic of the plume and the grid system.

Expected Results: The expected result is a time-varying concentration plot. This is a result of using the FRAMES WCF viewer on the output from the imported module.

5.1.2 Conducting the Test

1) Install the GMS models of interest. For RT3D, this includes RT3D.exe (Version 2.5) and RXNS.dll. For MT3D, this includes MT3DMS4.exe (Version 4.0). These files should be added to the following directory, which is the default: C:\Program Files\GMS50\models. If GMS 5.0 is already installed, then this step is not required.

2) Extract the latest version of the GMS Import tool if available. The latest version can be found on COLAB: https://colab.sc.egov.usda.gov/cb/displayDocument/GMSUpdate.zip?doc_id=35261. Extract all contents to C:\Program Files\FramesV2\GMS directory, provided C:\Program files\FramesV2 was the FRAMES installation directory.

3) Start the FRAMES 2.0 Framework Development Environment (if it is not already up).

4) If the "Simulation Editor" is started, you will need to close that by choosing File—>Exit from the menu.

5) Using the Module Editor, register the GMS Import module using the Open button and browsing to C:\Program Files\FramesV2\GMS\GMSImport.mod file. This will register the GMS Import tool.

6) Close the Module Editor.

7) From the Tools menu, launch the GMS Import tool, labeled as GMS Import.

8) Choose the example C:\Program Files\GMS50\tutfiles\rt3d\btexins1\sample\sample.gpr file. See Figure 5.2.

9) Set the name to "RT3DTutCon." See Figure 5.3. This will be the name of the new FRAMES 2.0 module.

10) "Select model output file for reading" as "Use CON file for obtaining output."

11) Set to “Yes (User input to FRAMES)” to

a) “Import, from FRAMES, the species concentrations at nodes bounded by Rows (16-16) Cols (16-16) Layers (1 to 1).” When the user chooses Yes, input from an upstream model is used as the boundary condition to RT3D. If “No” had been chosen, input to RT3D from an upstream model in FRAMES would have been ignored, and the original boundary condition used in the GMS baseline case would have been used.

b) “Export, to FRAMES, the species concentrations?” When the user chooses Yes, data from the execution of RT3D from within FRAMES is passed along to downstream models or viewers. If the user had chosen No, then the output from the GMS baseline case would have been passed along to downstream models or viewers.

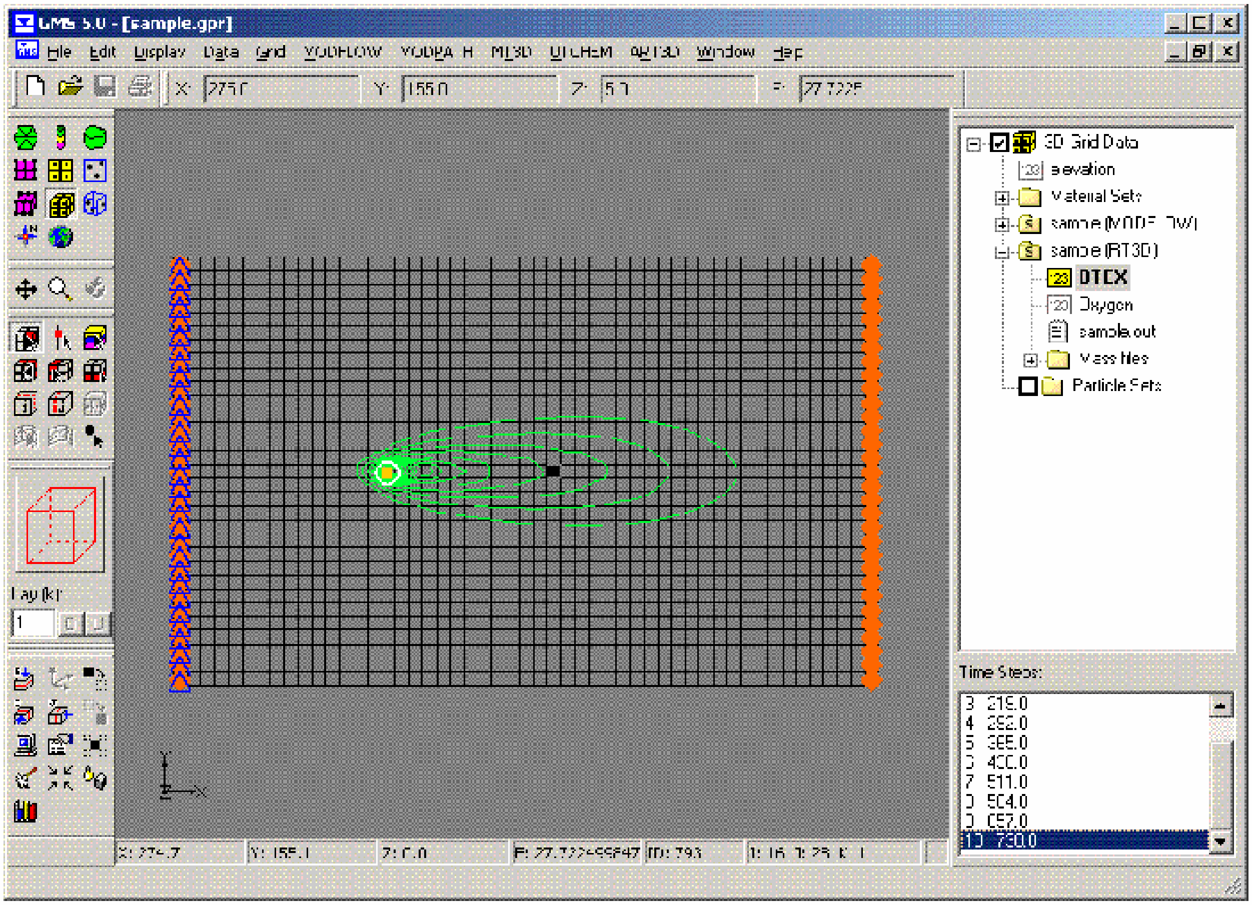


Figure 5.1. Illustrative Graphic of the Plume and the Grid System for RT3D Test Cases #1 and #2

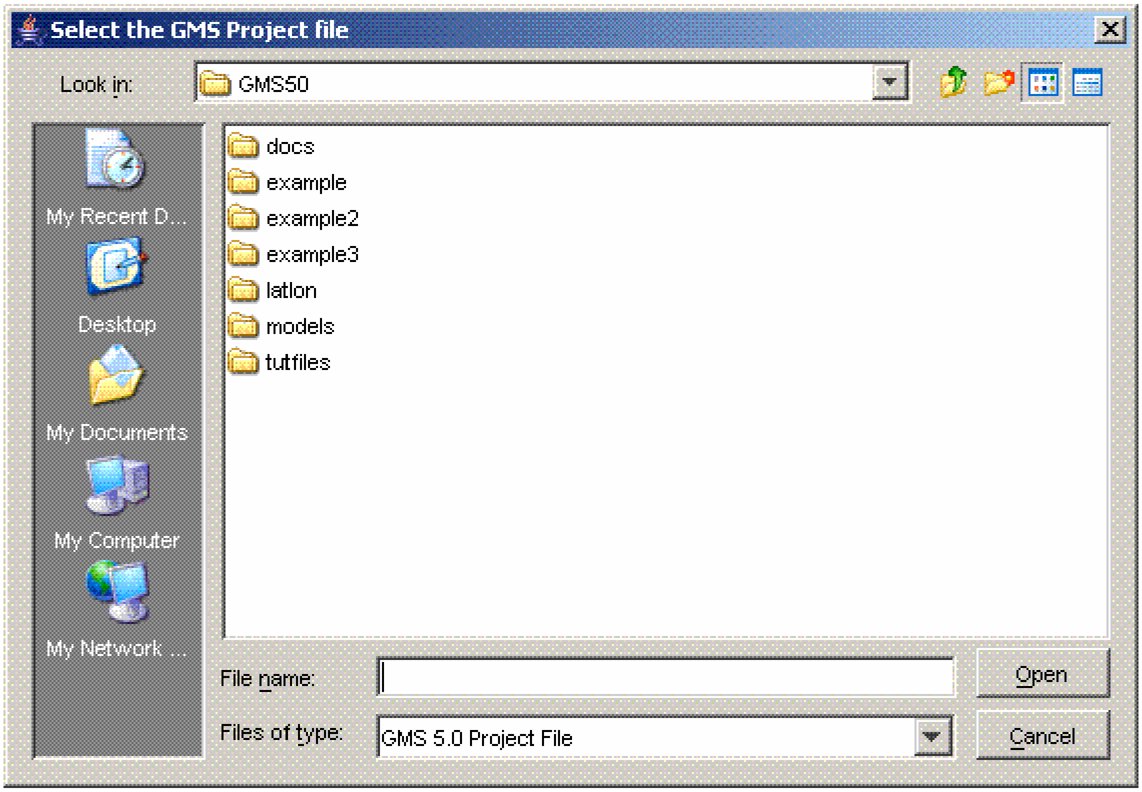
12) Map BTEX to JP4 for the chemical. This is done by clicking on BTEX, then clicking the first chemical in the list, and then pressing the “j” key on the keyboard. Use the arrow keys to check your selection.

13) For Row, Column, and Layer input, leave blank. See Figure 5.3 for an example RT3DTutCon ready for import.

14) Press the Build button at the bottom of the screen.

15) A message should come up saying the module was built, and no errors should come up.

16) Choose the Simulation Editor, Environmental Domain.



Look in: GMS50

Files of type: GMS 5.0 Project File

Figure 5.2. **T**he GMS Project Selection Dialog for Test Cases

17) Add a geographic information system (GIS) and a "Chemical Properties" to a simulation in the global (upper) work space area from under the Database heading on the icon palette on the left side of the screen. See Figure 3.11.

18) In the local (lower) work space from the Class "Models," choose icons Source, Aquifer, Exposure Pathways, Receptor Intakes, and Health Impacts.

19) Connect the modules in the local work space in the manner illustrated in Figure 3.11.

20) Choose the "FRAMES Default Chemical Properties" for the Chemical Properties under "General Information."

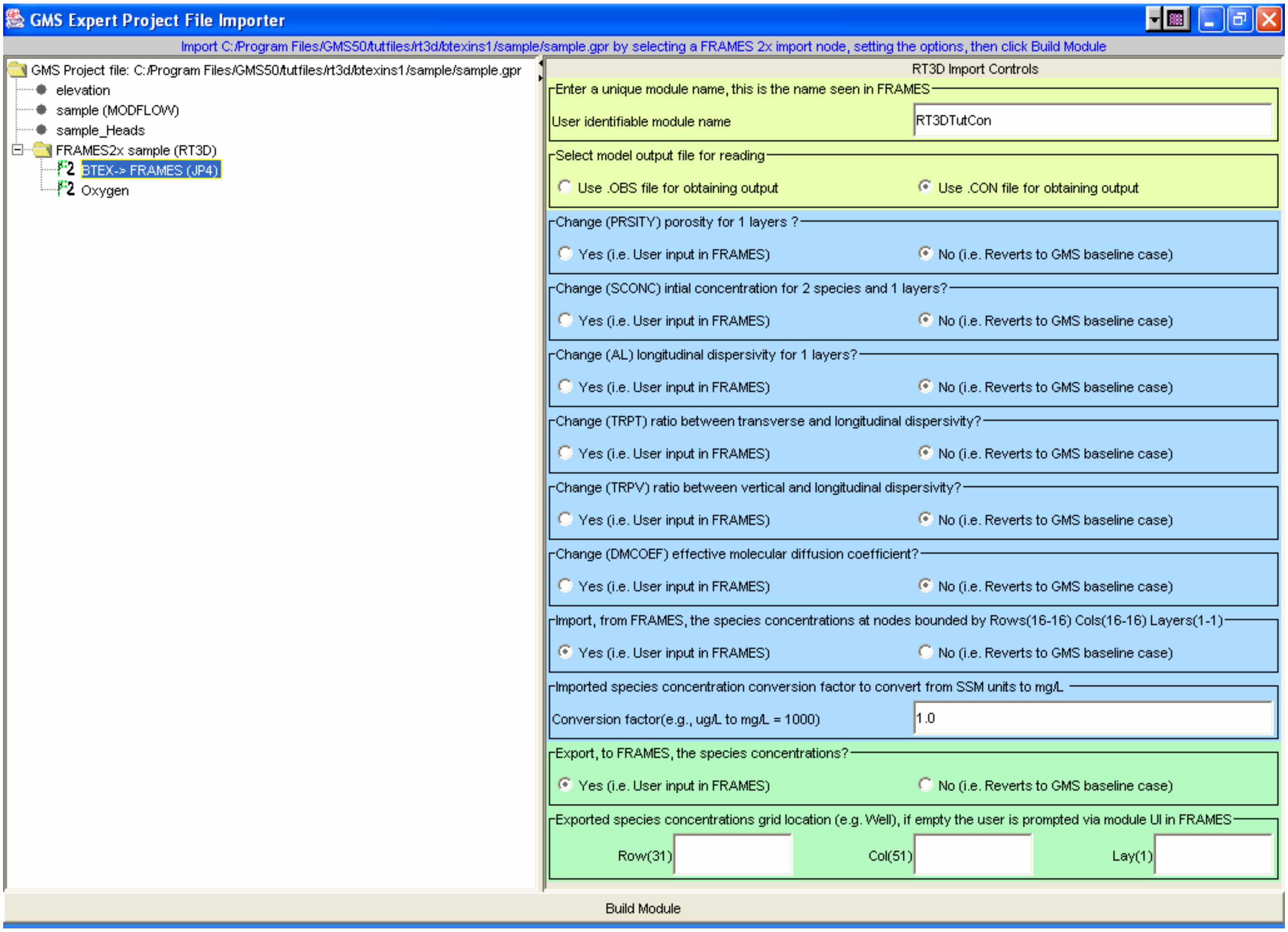
21) Run the User interface for Chemical Properties and choose "JP4." Click "File—>Save and Exit" to save your choices.

22) Run the User interface for the GIS. You need to put "0" for all X, Y, and Z for all modules. "File—>Save and Exit" is the user interface for the GIS.

23) Under the General Info for Aquifer, choose RT3DTutCon. See Figure 3.12.

24) Run the user interface for the RT3DTutCon module. See Figure 3.18.

25) The inputs for RT3DTutCon should be Col=28, Row=16, Lay=1, tfinal=4, and ticks=100.



GMS Expert Project File Importer

Import C:/Program Files/GMS50/tutfiles/rt3d/btexins1/sample/sample.gpr by selecting a FRAMES 2x import node, setting the options, then click Build Module

GMS Project file: C:/Program Files/GMS50/tutfiles/rt3d/btexins1/sample/sample.gpr
- elevation
- sample (MODFLOW)
- sample_Heads
- FRAMES2x sample (RT3D)
  - BTEX-> FRAMES (JP4)
  - Oxygen

RT3D Import Controls

Enter a unique module name, this is the name seen in FRAMES

User identifiable module name: RT3DTutCon

Select model output file for reading

○ Use .OBS file for obtaining output ● Use .CON file for obtaining output

Change (PRSITY) porosity for 1 layers ?

○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (SCONC) intial concentration for 2 species and 1 layers?

○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (AL) longitudinal dispersivity for 1 layers?

○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (TRPT) ratio between transverse and longitudinal dispersivity?

○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (TRPV) ratio between vertical and longitudinal dispersivity?

○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (DMCOEF) effective molecular diffusion coefficient?

○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Import, from FRAMES, the species concentrations at nodes bounded by Rows(16-16) Cols(16-16) Layers(1-1)

● Yes (i.e. User input in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

Imported species concentration conversion factor to convert from SSM units to mg/L

Conversion factor(e.g., ug/L to mg/L = 1000): 1.0

Export, to FRAMES, the species concentrations?

● Yes (i.e. User input in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

Exported species concentrations grid location (e.g. Well), if empty the user is prompted via module UI in FRAMES

Row(31) Col(51) Lay(1)

Build Module

Figure 5.3. The RT3D Test Case #1 RT3DTutCon Example Ready for Import

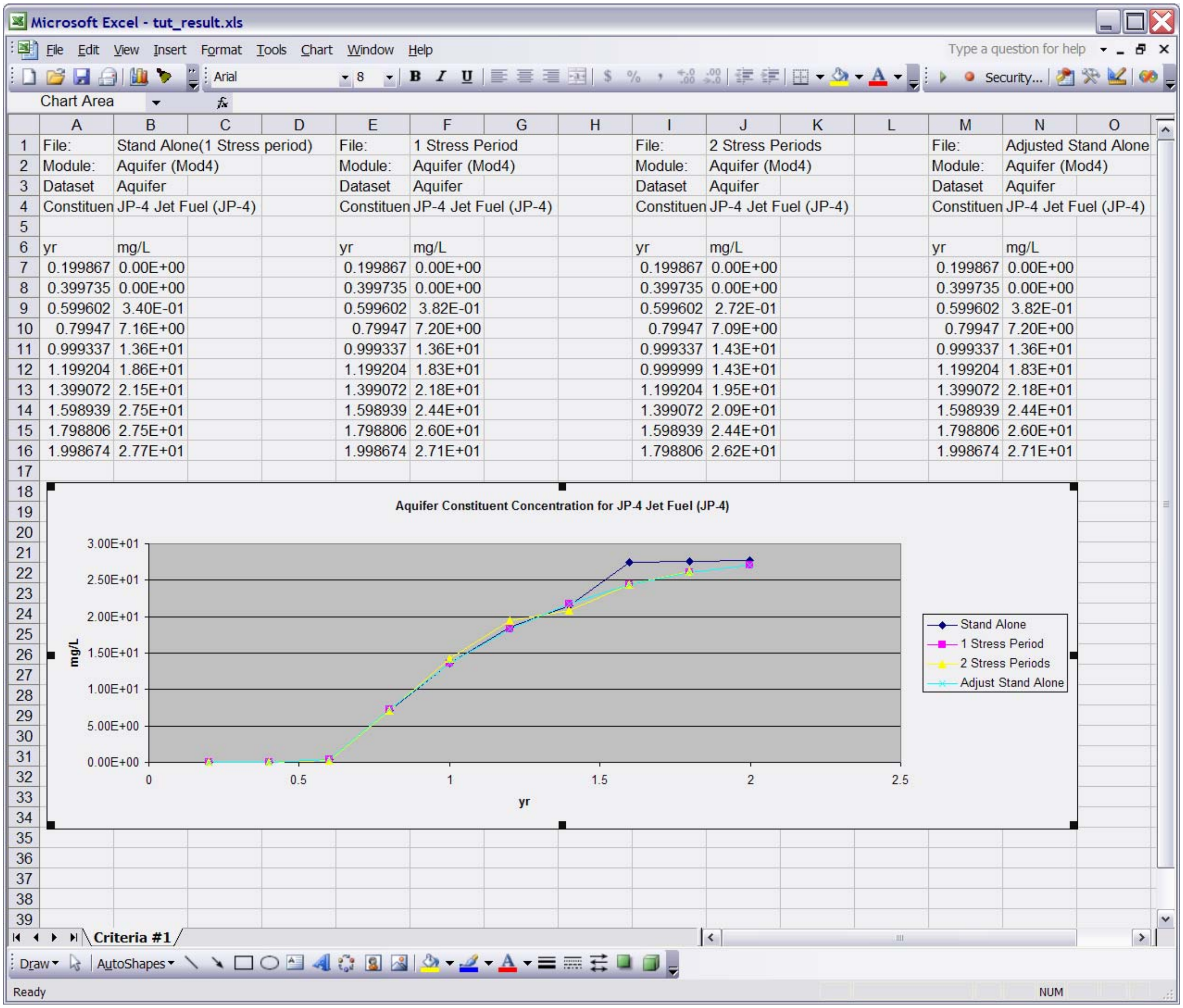
26) Right click and Run the RT3DTutCon icon. A scrolling window will display the status of the pre-processor, RT3D invocation, and the post-processor. A pause may occur between each one of these steps.

27) Right click when the light turns green on the Aquifer icon, and choose the Water Concentration File (WCF) Graphical viewer from the viewer list by using the left click and running the Graphical viewer.

5.1.3 Results

The resulting figure matches the values in Figure 3.22b. These results match the stand-alone version.

An expanded analysis of the results associated with this test case was also performed. The objective of this analysis was to explore the ramifications of increasing the number of stress periods associated with the input, without changing the input. If the input does not change, even though the number of points that describe the input is increased, then the results should not change. The first test looked at a single stress period where the stand-alone RT3D case contained a constant concentration. These results are denoted as “1 Stress Period” in Figure 5.4. The second test divided the source term into two time points with the same constant concentration. These results are denoted as “2 Stress Periods” in Figure 5.4. These results match the stand-alone case, labeled as “Adjust Stand Alone” in Figure 5.4. The third test divided the source term into multiple time points with the same constant concentration. These results also matched the “Adjust Stand Alone” results presented in Figure 5.4. If the total simulation time of the output (i.e., output end time) changes, then the intermediate time points change, and the curve can present a slightly different appearance. For example, although the results for the “Stand Alone” and “Adjust Stand Alone” output in Figure 5.4 are from the same simulated case, the concentrations do not appear identical because the end times are slightly different. The “Adjust Stand Alone” case ran with an end time of 730.484 days, and the “Stand Alone” case ran with an end time of 730 days. In summary, increasing the number of stress periods did not change the overall time-varying output results, other than providing output concentrations at times that were different.



Microsoft Excel - tut_result.xls
File Edit View Insert Format Tools Chart Window Help
Type a question for help
Arial
8
Security...
Chart Area
Criteria #1
Draw
AutoShapes
Ready
NUM

	A	B	E	F	I	J	M	N
1	File:	Stand Alone(1 Stress period)	File:	1 Stress Period	File:	2 Stress Periods	File:	Adjusted Stand Alone
2	Module:	Aquifer (Mod4)	Module:	Aquifer (Mod4)	Module:	Aquifer (Mod4)	Module:	Aquifer (Mod4)
3	Dataset	Aquifer	Dataset	Aquifer	Dataset	Aquifer	Dataset	Aquifer
4	Constituen	JP-4 Jet Fuel (JP-4)	Constituen	JP-4 Jet Fuel (JP-4)	Constituen	JP-4 Jet Fuel (JP-4)	Constituen	JP-4 Jet Fuel (JP-4)
5								
6	yr	mg/L	yr	mg/L	yr	mg/L	yr	mg/L
7	0.199867	0.00E+00	0.199867	0.00E+00	0.199867	0.00E+00	0.199867	0.00E+00
8	0.399735	0.00E+00	0.399735	0.00E+00	0.399735	0.00E+00	0.399735	0.00E+00
9	0.599602	3.40E-01	0.599602	3.82E-01	0.599602	2.72E-01	0.599602	3.82E-01
10	0.79947	7.16E+00	0.79947	7.20E+00	0.79947	7.09E+00	0.79947	7.20E+00
11	0.999337	1.36E+01	0.999337	1.36E+01	0.999337	1.43E+01	0.999337	1.36E+01
12	1.199204	1.86E+01	1.199204	1.83E+01	0.999999	1.43E+01	1.199204	1.83E+01
13	1.399072	2.15E+01	1.399072	2.18E+01	1.199204	1.95E+01	1.399072	2.18E+01
14	1.598939	2.75E+01	1.598939	2.44E+01	1.399072	2.09E+01	1.598939	2.44E+01
15	1.798806	2.75E+01	1.798806	2.60E+01	1.598939	2.44E+01	1.798806	2.60E+01
16	1.998674	2.77E+01	1.998674	2.71E+01	1.798806	2.62E+01	1.998674	2.71E+01

Figure 5.4. RT3D Test Simulation to Explore the Impact of Stress Periods on Output Results

5.2 RT3D Test Case #2: Using the OBS File

Description: RT3D Test Case #2 tests the GMS Import tool as it relates to all of the requirements, except Requirement 15, for the RT3D model within GMS. It does this by importing a GMS case titled sample.gpr and naming it RT3DTutObs.

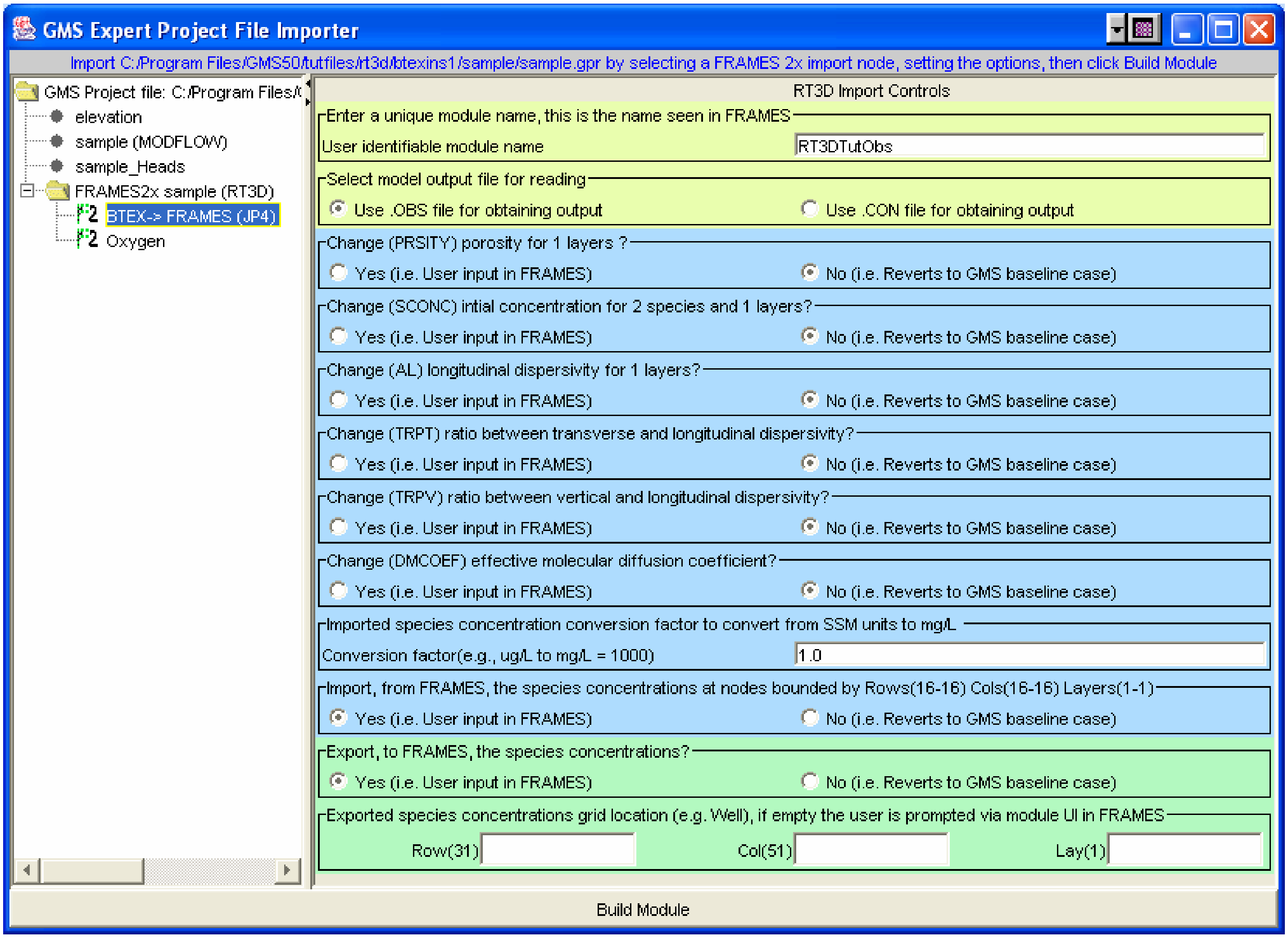
5.2.1 Input Data

The input data for this case is entered by the user in the GMS Import tool and in the DCE. The specific entries associated with each step are provided in the section titled “Conducting the Test.” It is assumed that the source term model provides constant output at 1000 mg/L to the RT3D model for two years. Figure 5.1 presents an illustrative graphic of the plume and the grid system.

Expected Results: The expected result is a time-varying concentration plot. This is a result of using the FRAMES WCF viewer on the output from the imported module.

5.2.2 Conducting the Test

1) Install the GMS models of interest. For RT3D, this includes RT3D.exe (Version 2.5) and RXNS.dll. For MT3D, this includes MT3DMS4.exe (Version 4.0). These files should be added to the following directory, which is the default: C:\Program Files\GMS50\models. If GMS 5.0 is already installed, then this step is not required.
2) Extract the latest version of the GMS Import tool if available. The latest version can be found on COLAB: https://colab.sc.egov.usda.gov/cb/displayDocument/GMSUpdate.zip?doc_id=35261. Extract all contents to the C:\Program Files\FramesV2\GMS directory, provided C:\Program files\FramesV2 was the FRAMES installation directory.
3) Start the FRAMES 2.0 Framework Development Environment (if it is not already up).
4) If the “Simulation Editor” is started, you will need to close that by choosing File—>Exit from the menu.
5) Using the Module Editor, register the GMS Import module using the Open button and browsing to C:\Program Files\FramesV2\GMS\GMSImport.mod file. This will register the GMS Import tool.
6) Close the Module Editor.
7) From the Tools menu, launch the GMS Import tool, labeled as GMS Import.
8) Choose the example C:\Program Files\GMS50\tutfiles\rt3d\btexins1\sample\sample.gpr file. See Figure 5.2.
9) Set the name to “RT3DTutObs.” See Figure 5.5. This will be the name of the new FRAMES 2.0 module.
10) “Select model output file for reading” as “Use OBS file for obtaining output.”
11) Set to “Yes (User input to FRAMES)” to
    a) “Import, from FRAMES, the species concentrations at nodes bounded by Rows (16 to 16), Columns (16 to 16), Layers (1 to 1)”
    b) “Export, to FRAMES, the species concentrations?”



GMS Expert Project File Importer

Import C:/Program Files/GMS50/tutfiles/rt3d/btexins1/sample/sample.gpr by selecting a FRAMES 2x import node, setting the options, then click Build Module

GMS Project file: C:/Program Files/(
- elevation
- sample (MODFLOW)
- sample_Heads
- FRAMES2x sample (RT3D)
  - BTEX-> FRAMES (JP4)
  - Oxygen

RT3D Import Controls

Enter a unique module name, this is the name seen in FRAMES
User identifiable module name: RT3DTutObs

Select model output file for reading
● Use .OBS file for obtaining output ○ Use .CON file for obtaining output

Change (PRSITY) porosity for 1 layers ?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (SCONC) intial concentration for 2 species and 1 layers?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (AL) longitudinal dispersivity for 1 layers?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (TRPT) ratio between transverse and longitudinal dispersivity?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (TRPV) ratio between vertical and longitudinal dispersivity?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (DMCOEF) effective molecular diffusion coefficient?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Imported species concentration conversion factor to convert from SSM units to mg/L
Conversion factor(e.g., ug/L to mg/L = 1000): 1.0

Import, from FRAMES, the species concentrations at nodes bounded by Rows(16-16) Cols(16-16) Layers(1-1)
● Yes (i.e. User input in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

Export, to FRAMES, the species concentrations?
● Yes (i.e. User input in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

Exported species concentrations grid location (e.g. Well), if empty the user is prompted via module UI in FRAMES
Row(31) Col(51) Lay(1)

Build Module

Figure 5.5. The RT3D Test Case #2 RT3DTutObs Example Ready for Import

12) Map BTEX to JP4 for the chemical. This is done by clicking on BTEX, then clicking the first chemical in the list, and then pressing the "j" key on the keyboard. Use the arrow keys to check your selection.

13) For Row, Column, and Layer input, leave blank. See Figure 5.5 for an example RT3DTutObs ready for import.

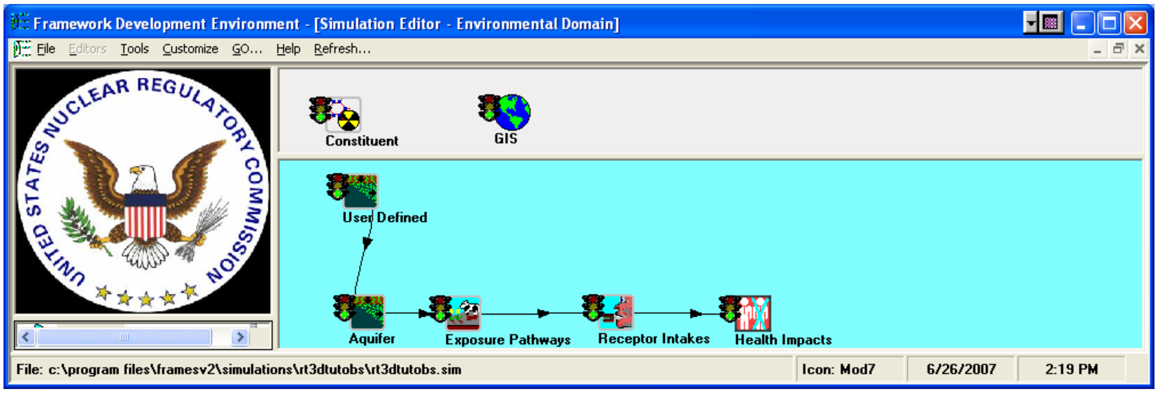
14) Press the build button at the bottom of the screen.

15) A message should come up saying the module was built, and no errors should come up.

16) Choose the Simulation Editor, Environmental Domain.

17) Add a GIS and a "Chemical Properties" to a simulation in the global (upper) work space area from under the Database heading on the icon palette on the left side of the screen. See Figure 5.6.

18) In the local (lower) work space from the Class "Models," choose icons Source, Aquifer, Exposure Pathways, Receptor Intakes, and Health Impacts.


Framework Development Environment - [Simulation Editor - Environmental Domain]
File Editors Tools Customize GO... Help Refresh...
UNITED STATES NUCLEAR REGULATORY COMMISSION
Constituent
GIS
User Defined
Aquifer
Exposure Pathways
Receptor Intakes
Health Impacts

File: c:\program files\framesv2\simulations\rt3dtutobs\rt3dtutobs.sim | Icon: Mod7 | 6/26/2007 | 2:19 PM

Figure 5.6. RT3D Test Simulation RT3DTutObs

19) Connect the modules in the local work space in the manner illustrated in Figure 5.6.

20) Choose the "FRAMES Default Chemical Properties" for the Chemical Properties under "General Information."

21) Run the User interface for Chemical Properties and choose "JP4." Click "File—>Save and Exit" to save your choices.

22) Run the User interface for the GIS. You need to put "0" for all X, Y, and Z for all modules. "File—>Save and Exit" is the user interface for the GIS.

23) Under the General Info for Aquifer, choose RT3DTutObs. See Figure 5.7.

24) Run the user interface for the RT3DTutObs module. See Figure 3.17.

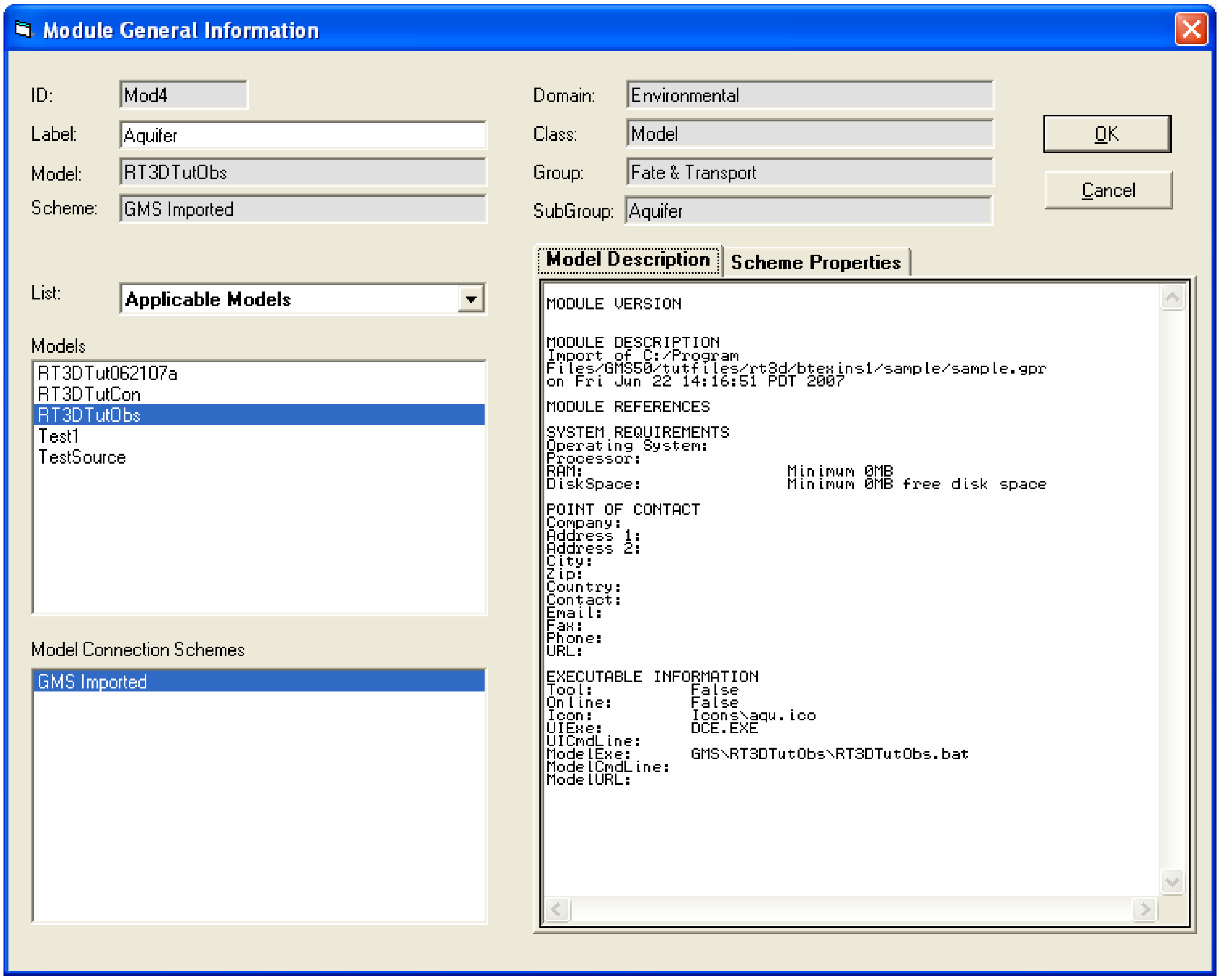
25) The inputs for RT3DTutObs should be Col=28, Row=16, Lay=1, and tfinal=4.

26) Right click and Run the RT3DTutObs icon. A scrolling window will display the status of the pre-processor, RT3D invocation, and the post-processor. A pause may occur between each one of these steps.

27) Right click when the light turns green on the Aquifer icon, and choose the Water Concentration File (WCF) Graphical viewer from the viewer list by using the left click and running the Graphical viewer.

5.2.3 Results

The resulting figure matches the values in Figure 3.20. These results match the stand-alone version.



ID: Mod4
Label: Aquifer
Model: RT3DTutObs
Scheme: GMS Imported
Domain: Environmental
Class: Model
Group: Fate & Transport
SubGroup: Aquifer

List: Applicable Models

Models
- RT3DTut062107a
- RT3DTutCon
- RT3DTutObs
- Test1
- TestSource

Model Connection Schemes
- GMS Imported

```
MODULE VERSION

MODULE DESCRIPTION
Import of C:/Program
Files/GMS50/tutfiles/rt3d/btexins1/sample/sample.gpr
on Fri Jun 22 14:16:51 PDT 2007

MODULE REFERENCES

SYSTEM REQUIREMENTS
Operating System:
Processor:
RAM:                         Minimum 0MB
DiskSpace:                   Minimum 0MB free disk space

POINT OF CONTACT
Company:
Address 1:
Address 2:
City:
Zip:
Country:
Contact:
Email:
Fax:
Phone:
URL:

EXECUTABLE INFORMATION
Tool:          False
Online:        False
Icon:          Icons\aqu.ico
UIExe:         DCE.EXE
UICmdLine:
ModelExe:      GMS\RT3DTutObs\RT3DTutObs.bat
ModelCmdLine:
ModelURL:
```

Figure 5.7. Module General Information Screen for RT3D Test Simulation RT3DTutObs

5.3 MT3DMS Test Case #3

Description: MT3DMS Test Case #3 tests the GMS Import tool as it relates to all of the requirements, except Requirement 15, for the MT3DMS model within GMS. It does this by importing a GMS case titled sample.gpr and naming it MT3DTutObs.

5.3.1 Input Data

The input data for this case are entered by the user in the GMS Import tool and in the DCE. The specific entries associated with each step are provided in the section titled “Conducting the Test.” It is assumed that the source term model provides constant output at 57.9 mg/L to the MT3DMS model for one year. Figure 5.8 presents an illustrative graphic of the plume and the grid system.

Expected Results: The expected result is a time-varying concentration plot. This is a result of using the FRAMES viewer on the output from the imported module.

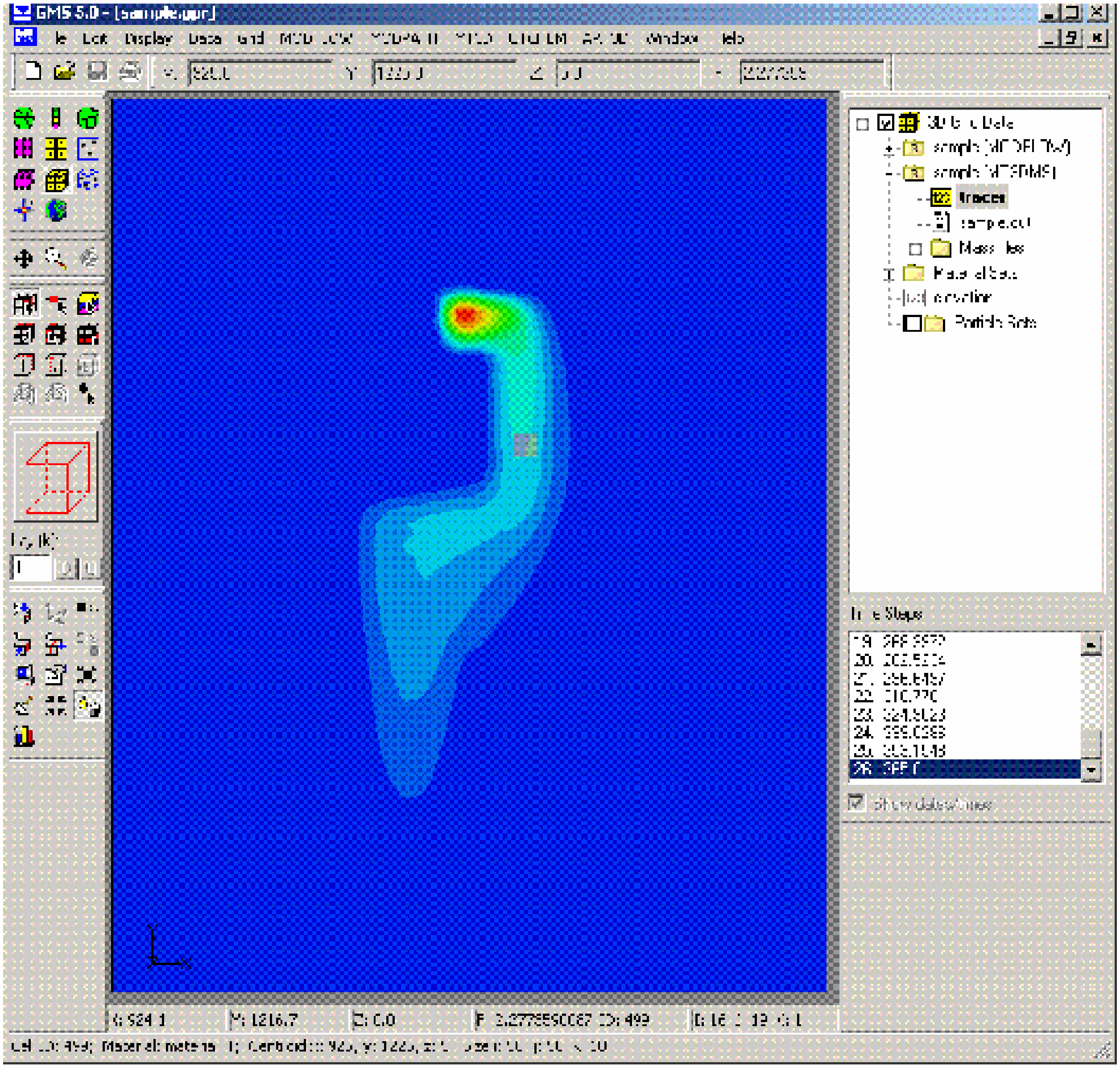


Figure 5.8. Illustrative Graphic of the Plume and the Grid System for MT3DMS Test Case #3

5.3.2 Conducting the Test

1) Install the GMS models of interest. For RT3D, this includes RT3D.exe (Version 2.5) and RXNS.dll. For MT3D, this includes MT3DMS4.exe (Version 4.0). These files should be added to the following directory, which is the default: C:\Program Files\GMS50\models. If GMS 5.0 is already installed, then this step is not required.

2) Extract the latest version of the GMS Import tool if available. The latest version can be found on COLAB: https://colab.sc.egov.usda.gov/cb/displayDocument/GMSUpdate.zip?doc_id=35261. Extract all contents to C:\Program Files\FramesV2\GMS directory, provided C:\Program files\FramesV2 was the FRAMES installation directory.

3) Start the FRAMES 2.0 Framework Development Environment (if it is not already up).

4) If the “Simulation Editor” is started, you will need to close that by choosing File—>Exit from the menu.

5) Using the Module Editor, register the GMS Import module using the Open button and browsing to C:\Program Files\FramesV2\GMS\GMSImport.mod file. This will register the GMS Import tool.

6) Close the Module Editor.

7) From the Tools menu, launch the GMS Import tool, labeled as GMS Import.

8) Choose the example C:\Program Files\GMS50\tutfiles\mt3dgrid\sample\sample.gpr file. See Figure 5.2.

9) Set the name to “MT3DTutObs.” See Figure 5.9. This will be the name of the new FRAMES 2.0 module.

10) “Select model output file for reading” as “Use OBS file for obtaining output.”

11) Set to “Yes (User input to FRAMES)” to

    a) “Import, from FRAMES, the species concentrations at nodes bounded by Rows (10 to 10), Columns (16 to 16), Layers (1 to 1)”

    b) “Export, to FRAMES, the species concentrations?”

12) Map Tracer to Benzene for the chemical. This is most quickly done by clicking on Tracer, then clicking the first chemical in the list, and then pressing the “b” key on the keyboard. Use the arrow keys to check your selection.

13) For Row, Col, and Layer input, leave blank. See Figure 5.9 for an example MT3DTutObs ready for import.

14) Press the build button at the bottom of the screen.

15) A message should come up saying the module was built, and no errors should come up.

16) Choose the Simulation Editor, Environmental Domain.

17) Add a GIS and a “Chemical Properties” to a simulation in the global (upper) work space area from under the Database heading on the icon palette on the left side of the screen. See Figure 5.10.

18) In the local (lower) work space from the Class “Models,” choose icons Source, Aquifer, Exposure Pathways, Receptor Intakes, and Health Impacts.

19) Connect the modules in the local work space in the manner illustrated in Figure 5.10.

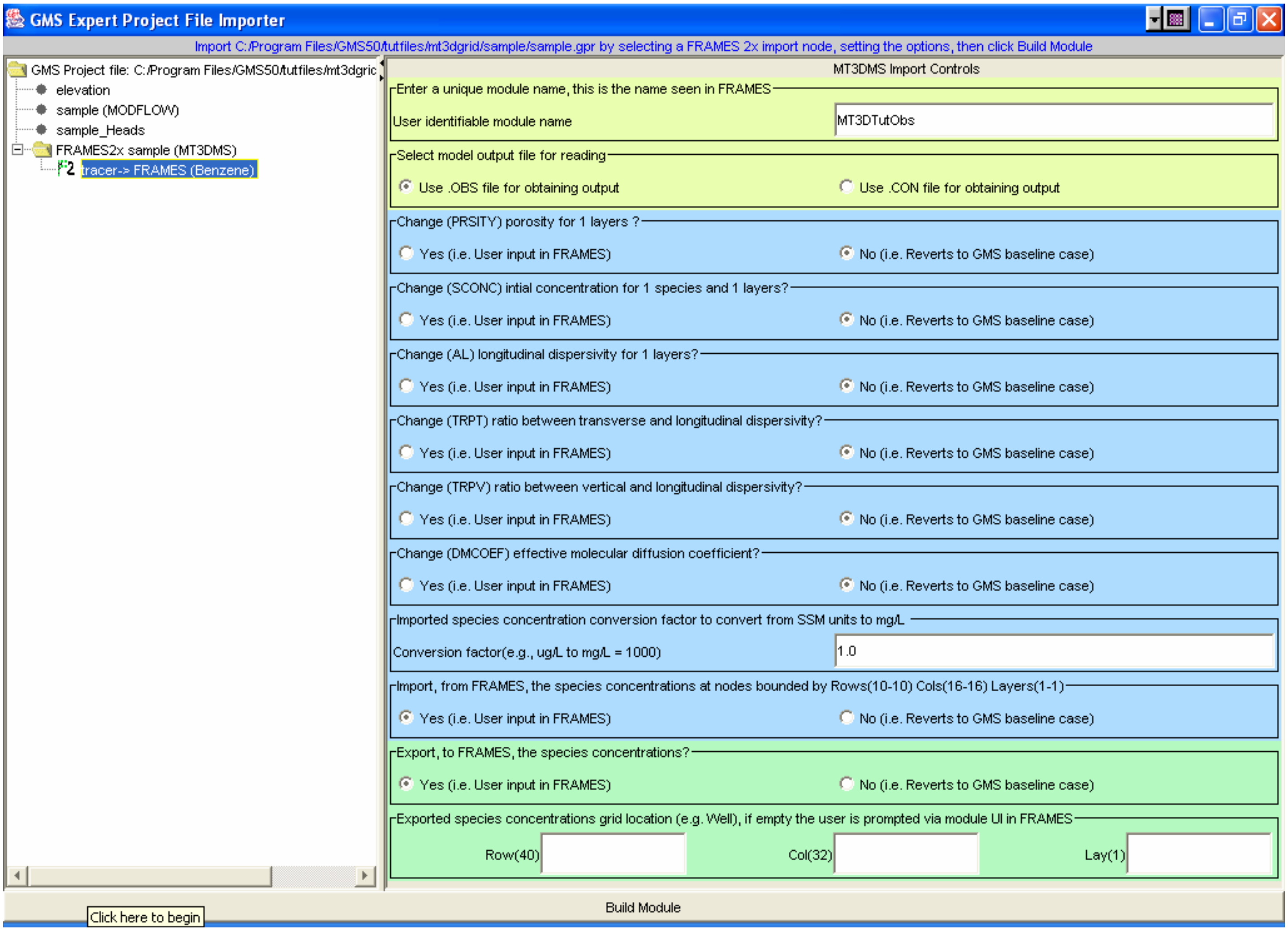
20) Choose the “FRAMES Default Chemical Properties” for the Chemical Properties under “General Information.”

21) Run the User interface for Chemical Properties and choose “Benzene.” Click “File—>Save and Exit” to save your choices.

22) Run the User interface for the GIS. You need to put “0” for all X, Y, and Z for all modules. “File—>Save and Exit” the user interface for the GIS.

23) Under the General Info for Aquifer, choose MT3DTutObs.

24) Run the user interface for the MT3DTutObs module.



GMS Expert Project File Importer

Import C:/Program Files/GMS50/tutfiles/mt3dgrid/sample/sample.gpr by selecting a FRAMES 2x import node, setting the options, then click Build Module

- GMS Project file: C:/Program Files/GMS50/tutfiles/mt3dgric
  - elevation
  - sample (MODFLOW)
  - sample_Heads
  - FRAMES2x sample (MT3DMS)
    - 2 tracer-> FRAMES (Benzene)

MT3DMS Import Controls

Enter a unique module name, this is the name seen in FRAMES

User identifiable module name: MT3DTutObs

Select model output file for reading

● Use .OBS file for obtaining output ○ Use .CON file for obtaining output

Change (PRSITY) porosity for 1 layers ?

○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (SCONC) intial concentration for 1 species and 1 layers?

○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (AL) longitudinal dispersivity for 1 layers?

○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (TRPT) ratio between transverse and longitudinal dispersivity?

○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (TRPV) ratio between vertical and longitudinal dispersivity?

○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (DMCOEF) effective molecular diffusion coefficient?

○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Imported species concentration conversion factor to convert from SSM units to mg/L

Conversion factor(e.g., ug/L to mg/L = 1000): 1.0

Import, from FRAMES, the species concentrations at nodes bounded by Rows(10-10) Cols(16-16) Layers(1-1)

● Yes (i.e. User input in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

Export, to FRAMES, the species concentrations?

● Yes (i.e. User input in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

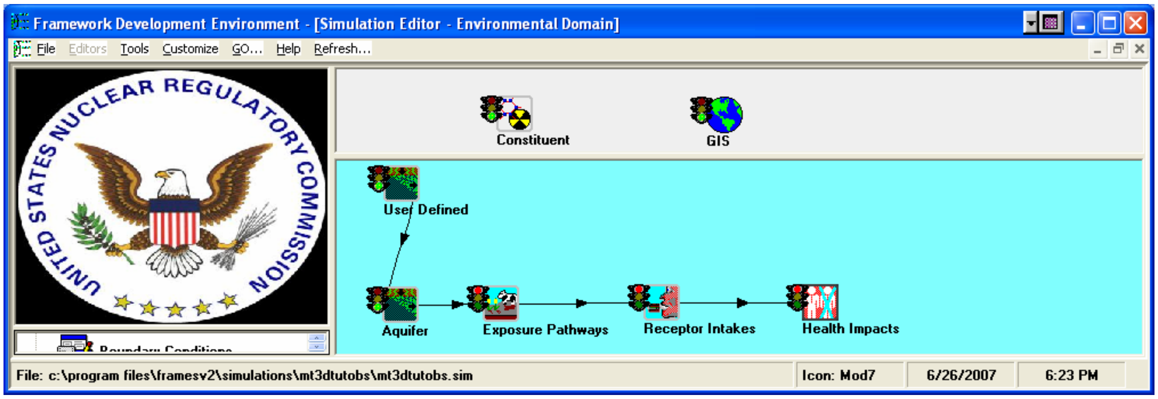
Exported species concentrations grid location (e.g. Well), if empty the user is prompted via module UI in FRAMES

Row(40) Col(32) Lay(1)

Click here to begin

Build Module

Figure 5.9. The MT3DMS Example Case #3 Ready for Import



File: c:\program files\framesv2\simulations\mt3dtutobs\mt3dtutobs.sim Icon: Mod7 6/26/2007 6:23 PM

Figure 5.10. MT3DMS Test Case #3 Simulation Conceptual Site Model

25) The inputs for MT3DTutObs should be Col=19, Row=16, and Layer=1, and tfinal=5.

26) Right click and run the MT3DTutObs icon. A scrolling window will display the status of the preprocessor, MT3DMS invocation, and the post-processor. Currently, a pause occurs between each one of these steps.

27) When the light turns green on the Aquifer icon, right click and choose the WCF Graphical viewer from the viewer list by using the left click and running the Graphical viewer.

5.3.3 Results

The resulting figure matches the values in Figure 5.11.

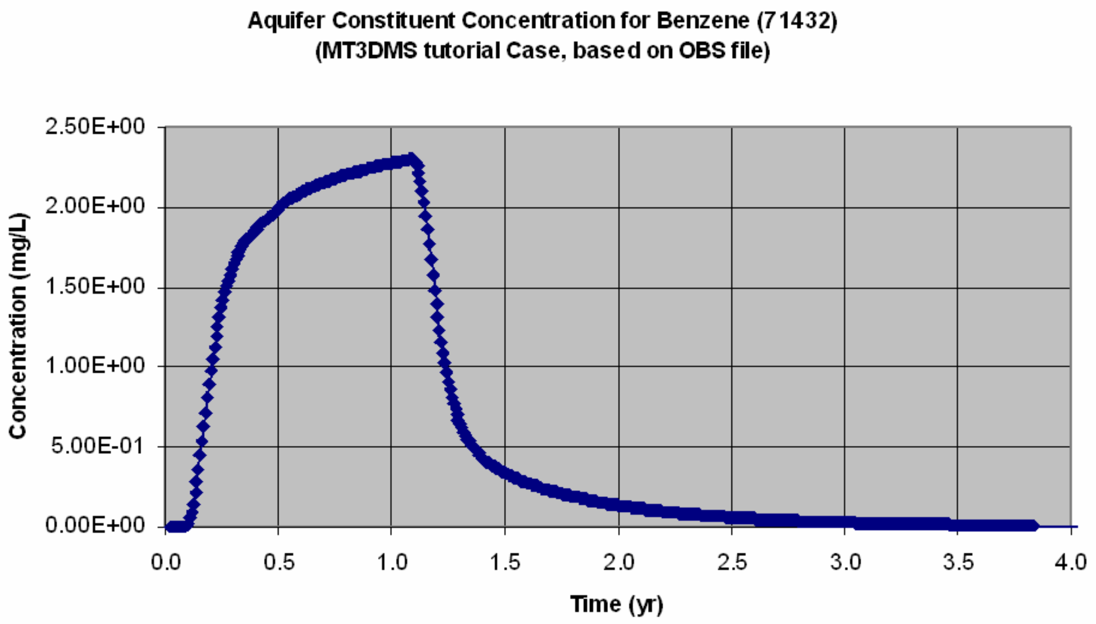


Figure 5.11. Expected Results from MT3DMS Test Case #3

5.4 Area Source RT3D Test Case #4

Description: This test case is intended to test the capability of FRAMES to provide time-varying area source concentrations to the RT3D model within GMS and the capability to handle multiple constituents within the same application. This test meets all of the requirements, except Requirement 15, for the RT3D model within GMS. It does this by importing a GMS case named B25_.gpr and naming it ChrisObs. In this case, an area source, as represented by a landfill, is initially uniformly contaminated at a constant concentration with TCE, DCE, VC, and a non-reactive tracer (i.e., chloride). The source provides contamination to an aquifer. The aquifer transports the contamination to a nearby extraction well where the contaminated water is withdrawn and consumed through various exposure routes, impacting the health of the receptors.

5.4.1 Input Data

The RT3D baseline simulation characteristics of the problem are as follows:

- Constituents: TCE, DCE, VC, and a non-reactive tracer (i.e., chloride)
- Area Source: 1500 ft by 225 ft by 16.718 ft
- Waste Site Porosity: 0.24
- Waste Site Bulk Density: 1.64 g/cm^3 (With a combination of porosity and bulk density, the particle density appears a bit low at 2.16 g/cm^3, but this is what is in the modeling case.)
- Waste Site Darcy Velocity: 2 cm/yr
- Waste site Kd: 2 mL/g, 2 mL/g, and 2 mL/g for TCE, DCE, and VC, respectively (possible parameter change)
- Initial masses in the source for TCE, DCE, and VC are 1 kg each.
- Solubility Limits: 1100 mg/L, 2250 mg/L, 2670 mg/L for TCE, DCE, and VC, respectively. Because solubility limits are never reached, variations in solubility limits between constituents have no impact on the source modeling.
- Pore Water Velocity: 2.74 cm/d at the waste site and 9.14 to 21.3 cm/d at a lower level.
- Porosity: 0.3
- Bulk Density: 1.6 g/cm^3 (With this combination, the particle density appears a bit low at 2.29 g/cm^3, but this is what is in the modeling case.)
- Initial groundwater TCE, DCE, and VC Concentrations: 4.17E-03 mg/L
- Half-Life at the source for TCE, DCE, and VC: 27.4 yr (10000 d), which equates to a first-order degradation rate of 0.025 yr^{-1} (6.93E-05 d^{-1})
- Kd at the source for TCE, DCE, and VC: 2 mL/g. In the groundwater model, the Kds were assumed as 0.57 mL/g, 0.25 mL/g, and 0.17 mL/g for TCE, DCE, and VC, respectively (possible parameter change).

Figure 5.12 presents the GMS RT3D spatial gridding, and Figure 5.13 presents the water-surface elevations with flow from the top of the page to the bottom of the page.

The input data for this case are entered by the user in the GMS Expert Project File Import tool and in the DCE. The specific entries associated with each step are provided in the section titled “Conducting the Test.”

Expected Results: The expected result is a time-varying concentration plot. This is a result of using the FRAMES viewer on the output from the imported module.

5.4.2 Conducting the Test

1. Install the GMS models of interest. For RT3D, this includes RT3D.exe (Version 2.5) and RXNS.dll. For MT3D, this includes MT3DMS4.exe (Version 4.0). These files should be added to the following directory, which is the default: C:\Program Files\GMS50\models. If GMS 5.0 is already installed, then this step is not required.

2. Extract the latest version of the GMS Import tool if available. The latest version can be found on COLAB: https://colab.sc.egov.usda.gov/cb/displayDocument/GMSUpdate.zip?doc_id=35261.

Extract all contents to the C:\Program Files\FramesV2\GMS directory, provided C:\Program files\FramesV2 was the FRAMES installation directory.

3. Start the FRAMES V2 Framework Development Environment (if it is not already up).

4. If the "Simulation Editor" is started, you will need to close that by choosing File—>Exit from the menu.

5. Using the module editor, open the C:\Program Files\FramesV2\GMS\GMSImport.mod file. This will register the GMS Import tool.

6. Close the module Editor.

7. From the Tools menu, launch the GMS Import tool labeled as GMS Import.

8. Choose the example C:\Program Files\GMS50\tutfiles\Example\B25_.gpr file. See Figure 5.2.

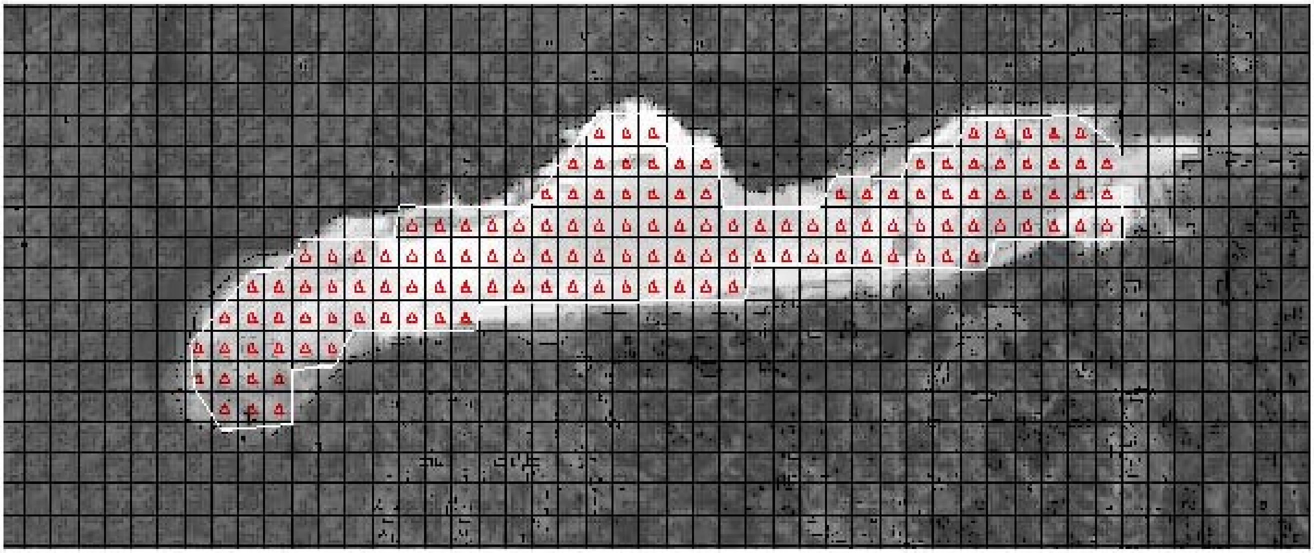


Figure 5.12. GMS RT3D Test Case #4 Source-Term Spatial Description

9. Set the name to "ChrisObs." This will be the name of the new FRAMES 2.0 module.

10. Set to "yes" the "FRAMES provides concentration" and "FRAMES uses concentrations."

11. Map TCE, DCE, and VC to trichloroethylene, 1,1 dichloroethylene, and vinyl chloride, respectively. This is most quickly done by clicking on TCE, then scrolling down to trichloroethylene, and then pressing the "set" button. This procedure can be repeated for the other chemicals.

12. Leave blank for Row, Column, and Layer input. See Figure 5.14 for an example GMS RT3D ready for import.

13. Press the build button at the bottom of the screen.

14. A message should come up saying the module was built, and no errors should come up.

15. Choose the Simulation Editor, Environmental Domain.

16. Add a GIS and a “Chemical Properties” to a simulation in the global (upper) work-space area from under the Database heading on the icon palette on the left side of the screen.

17. In the local (lower) work space from the Class “Models,” choose icons Source, Aquifer, Exposure Pathways, Receptor Intakes, and Health Impacts.

18. Connect the modules in the local work space in the manner illustrated in Figure 5.15.

19. Choose the “FRAMES Default Chemical Properties” for the Chemical Properties under “General Information.”

20. Run the User interface for Chemical Properties and choose “1,1 Dichloroethylene,” “TCE,” and “Vinyl chloride.” Click “File—>Save and Exit” to save your choices.

21. Run the User interface for the GIS. You need to put “0” for all X, Y, and Z for all modules. Click “File—>Save and Exit,” the user interface for the GIS.

22. Under General Info of the Source icon, choose MEPAS 5.0 Source in an Aquifer. Save and exit.

23. Under User Input of the Source icon, enter the source input data. Save and exit.

24. Run the Source Icon. The output results from the Source icon represent input to the Aquifer icon. Figure 5.16 presents the output for DCE from the source-term module (and hence input to the RT3D model). Because TCE, DCE, and vinyl chloride have the same properties, their output results are the same.

25. Under General Info for Aquifer, choose ChrisObs.

26. Run the user interface for the ChrisObs module.

27. The inputs for ChrisObs should be Col=35, Row=20, Layer=1, and tfinal=150.

28. Right click and run the ChrisObs icon. A scrolling window will display the status of the pre-processor, RT3D invocation, and the post-processor. A pause may occur between each one of these steps.

29. When the light turns green on the Aquifer icon, right click and choose the WCF Graphical viewer from the viewer list by using the left click and running the Graphical viewer.

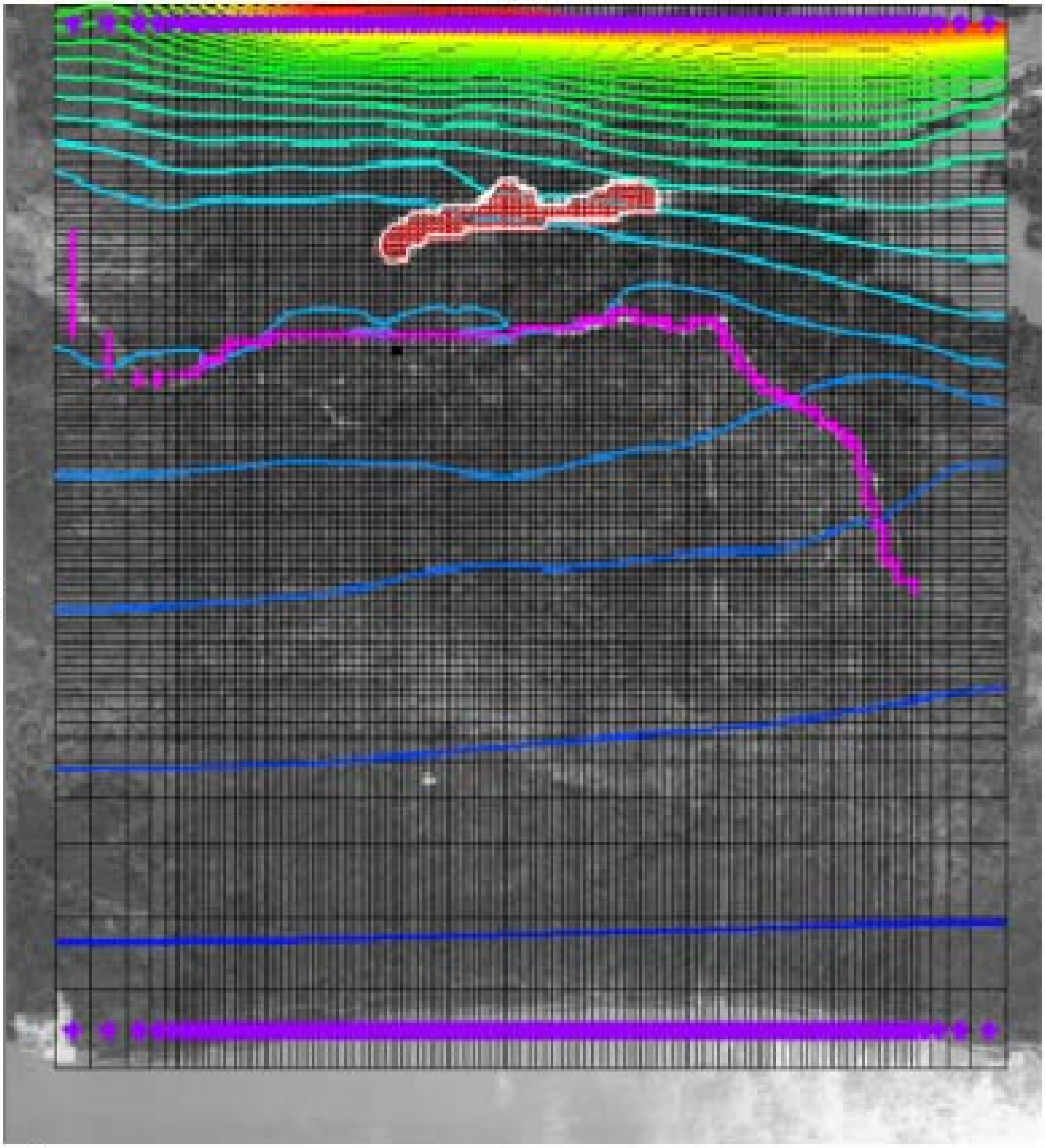
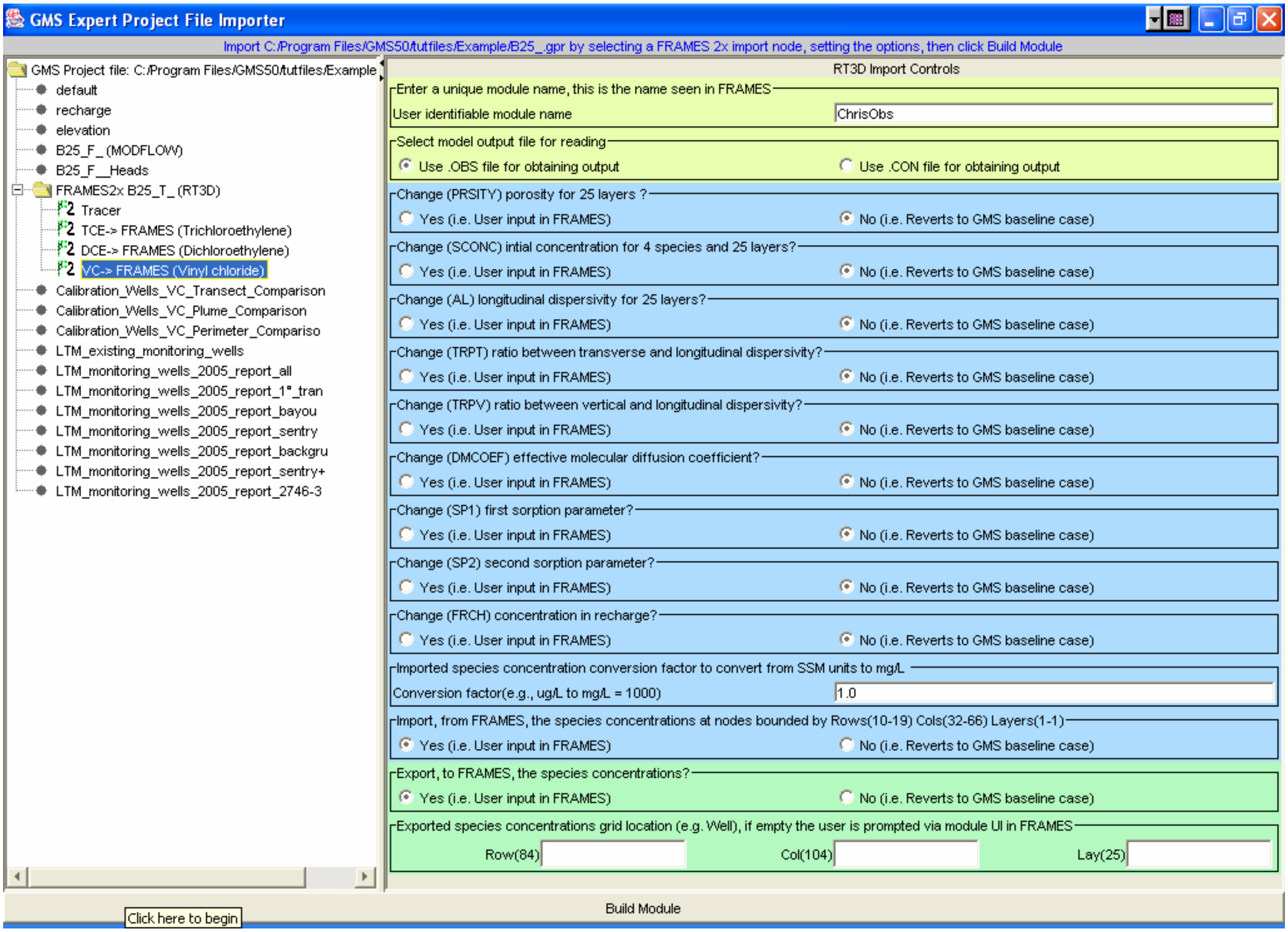


Figure 5.13. Water-Surface Elevation Contours (flow from top to bottom)



GMS Expert Project File Importer

Import C:/Program Files/GMS50/tutfiles/Example/B25_.gpr by selecting a FRAMES 2x import node, setting the options, then click Build Module

GMS Project file: C:/Program Files/GMS50/tutfiles/Example
- default
- recharge
- elevation
- B25_F_ (MODFLOW)
- B25_F__Heads
- FRAMES2x B25_T_ (RT3D)
  - Tracer
  - TCE-> FRAMES (Trichloroethylene)
  - DCE-> FRAMES (Dichloroethylene)
  - VC-> FRAMES (Vinyl chloride)
- Calibration_Wells_VC_Transect_Comparison
- Calibration_Wells_VC_Plume_Comparison
- Calibration_Wells_VC_Perimeter_Compariso
- LTM_existing_monitoring_wells
- LTM_monitoring_wells_2005_report_all
- LTM_monitoring_wells_2005_report_1°_tran
- LTM_monitoring_wells_2005_report_bayou
- LTM_monitoring_wells_2005_report_sentry
- LTM_monitoring_wells_2005_report_backgru
- LTM_monitoring_wells_2005_report_sentry+
- LTM_monitoring_wells_2005_report_2746-3

RT3D Import Controls

Enter a unique module name, this is the name seen in FRAMES
User identifiable module name: ChrisObs

Select model output file for reading
● Use .OBS file for obtaining output ○ Use .CON file for obtaining output

Change (PRSITY) porosity for 25 layers ?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (SCONC) intial concentration for 4 species and 25 layers?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (AL) longitudinal dispersivity for 25 layers?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (TRPT) ratio between transverse and longitudinal dispersivity?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (TRPV) ratio between vertical and longitudinal dispersivity?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (DMCOEF) effective molecular diffusion coefficient?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (SP1) first sorption parameter?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (SP2) second sorption parameter?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Change (FRCH) concentration in recharge?
○ Yes (i.e. User input in FRAMES) ● No (i.e. Reverts to GMS baseline case)

Imported species concentration conversion factor to convert from SSM units to mg/L
Conversion factor(e.g., ug/L to mg/L = 1000): 1.0

Import, from FRAMES, the species concentrations at nodes bounded by Rows(10-19) Cols(32-66) Layers(1-1)
● Yes (i.e. User input in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

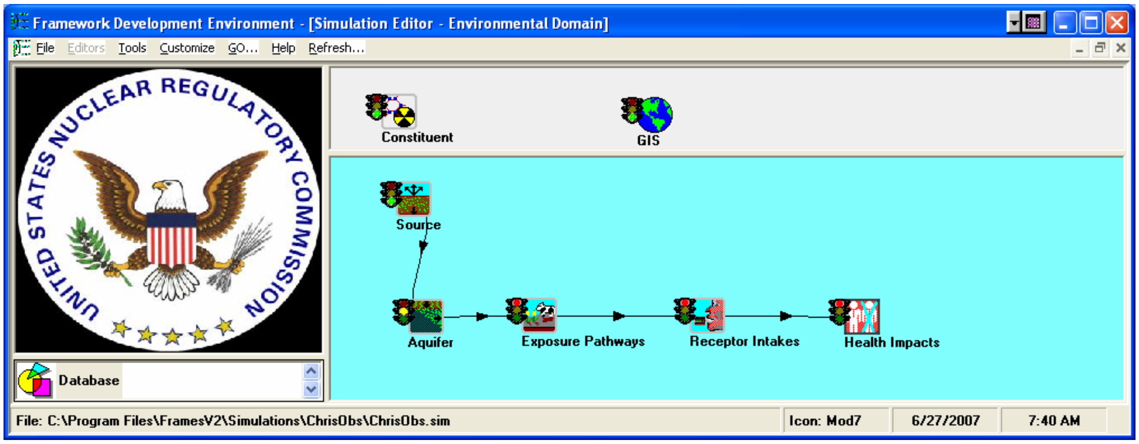
Export, to FRAMES, the species concentrations?
● Yes (i.e. User input in FRAMES) ○ No (i.e. Reverts to GMS baseline case)

Exported species concentrations grid location (e.g. Well), if empty the user is prompted via module UI in FRAMES
Row(84) Col(104) Lay(25)

Click here to begin

Build Module

Figure 5.14. RT3D Example Case #4 Ready for Import



Framework Development Environment - [Simulation Editor - Environmental Domain]

File: C:\Program Files\FramesV2\Simulations\ChrisObs\ChrisObs.sim | Icon: Mod7 | 6/27/2007 | 7:40 AM

Figure 5.15. RT3D Test Cast #4 Simulation

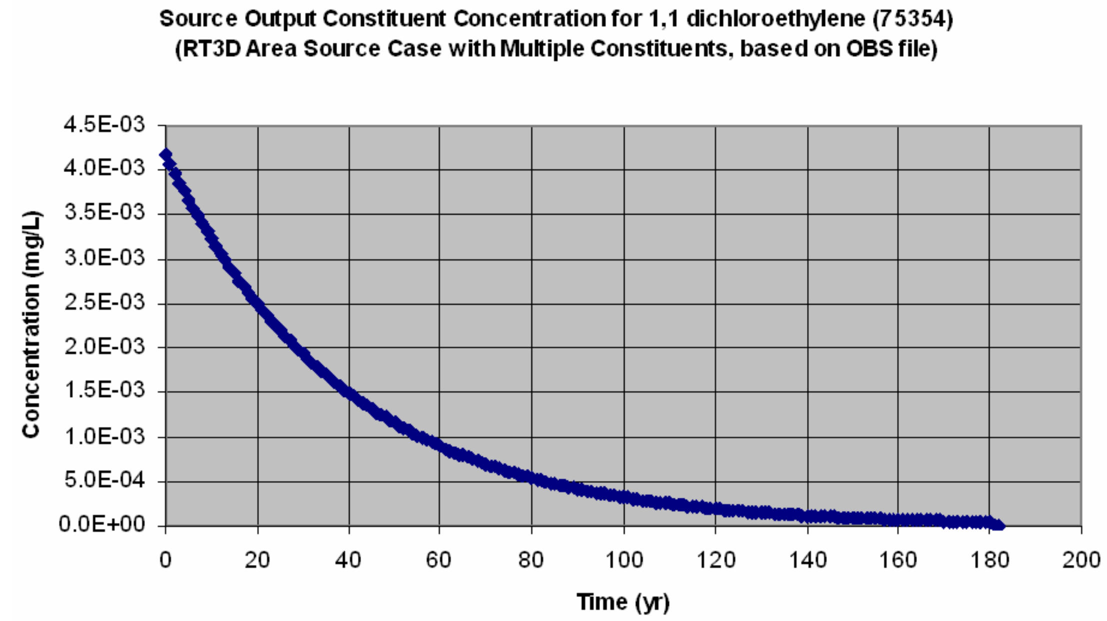


Figure 5.16. Output for DCE from the Source-Term Module, and Input to the RT3D Model (TCE, DCE, and Vinyl Chloride have the same output)

5.4.3 Results

The resulting figure does match the values in Figure 5.17.

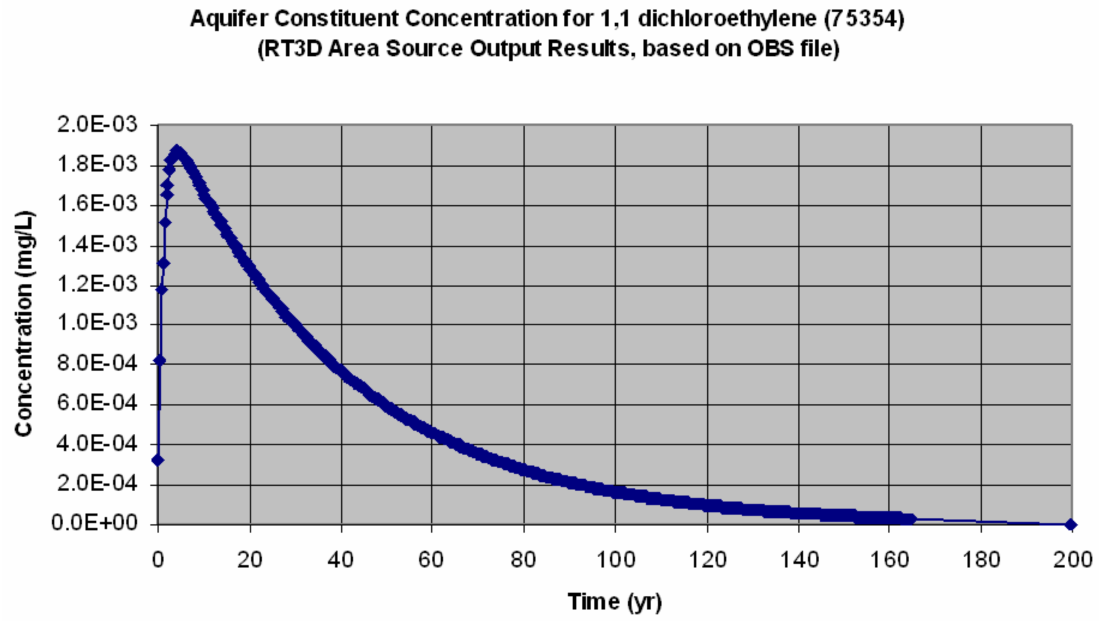


Figure 5.17. Expected Results for RT3D Test Case #4

6.0 Bibliography

Army Corps of Engineers (ACOE). 2007. Adaptive Risk Assessment Modeling System (ARAMS), Engineer Research and Development Center, Vicksburg, MS. Available at (accessed January 3, 2007): http://el.erdc.usace.army.mil/arams/

U.S. Environmental Protection Agency (EPA). 2007a. *Hazardous Waste Identification Rule Assessment*. National Exposure Research Laboratory. Available at (accessed July 4, 2007): http://www.epa.gov/nerl/factsheets/2001/goal5_assessment.pdf

U.S. Environmental Protection Agency (EPA). 2007b. *Supercomputer for Model Uncertainty and Sensitivity Evaluation.* Ecosystems Research Division. Available at (accessed July 4, 2007): http://www.epa.gov/athens/research/modeling/supermuse/supermuse.html

Whelan G., K.J. Castleton, M.A. Pelton, and R.Y. Taira. 2006. *Concepts Associated with Transferring Temporal and Spatial Boundary Conditions between Modules in the Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES)*. PNNL-16145. Pacific Northwest National Laboratory, Richland, Washington.

Whelan G, KJ Castleton, JW Buck, GM Gelston, BL Hoopes, MA Pelton, DL Strenge, and RN Kickert. 1997. *Concepts of a Framework for Risk Analysis in Multimedia Environmental Systems (FRAMES)*. PNNL-11748, Pacific Northwest National Laboratory, Richland, Washington.

6.1 Documentation for the FRAMES-3MRA Technology Software System

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

6.2 Quality Assurance Program Document

Gelston GM, RE Lundgren, JP McDonald, and BL Hoopes. 1998. *An Approach to Ensuring Quality in Environmental Software*. PNNL-11880, Pacific Northwest National Laboratory, Richland, Washington.

6.3 Additional Sources

Buck JW, BL Hoopes, KJ Castleton, and RY Taira. 1999. *Requirements for the FRAMES User Interface*. PNNL-SA-32277, Pacific Northwest National Laboratory, Richland, Washington (in publication).

Draper N, and H Smith. 1981. *Applied Regression Analysis*. 2nd ed, John Wiley and Sons, New York.

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

Faraway JJ. *Practical Regression and Anova Using R.* Available at: http://cran.r-project.org/doc/contrib/Faraway-PRA.pdf. Accessed 04-10-2007.

Gelston, GM, RE Lundgren, KJ Castleton, RY Taira, TE Swanson, JP Mcdonald, and BL Hoopes. 2006. *Software Integration Frameworks: An Approach to Ensuring Quality*. PNWD-3776. Pacific Northwest National Laboratory, Richland, Washington.

Gelston GM, RE Lundgren, JP McDonald, and BL Hoopes. 1998. *An Approach to Ensuring Quality in Environmental Software*. PNNL-11880, Pacific Northwest National Laboratory, Richland, Washington.

Gelston GM,MA Pelton, R Lundgren, KJ Castleton, G Whelan, BL Hoopes, J . Kirk AJ Pospical, M Eslinger, JG Droppo, Jr., and DL Strenge. 2004. Documentation for the Dictionary Editor of the FRAMEwork System (FRAMES). PNWD-3504, Pacific Northwest National Laboratory, Richland, Washington.

Hoopes BL, MA Pelton, KJ Castleton, GM Gelston, G Whelan, RY Taira. 2004. Documentation for the FRAMEwork Development Environment. PNWD-3509, Pacific Northwest Laboratory, Richland, Washington.

Neter J, W Wasserman, and MH Kutner. 1985. *Applied Linear Statistical Models*. 2nd ed., Homewood, Ill., Richard D. Irwin, Inc. pp. 288-290, 417-444, 508.

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

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