Release Notes for
SUTRA Version 2D3D.1

September 1, 2003

Clifford I. Voss (cvoss@usgs.gov) and Alden M. Provost (aprovost@usgs.gov)
U.S. Geological Survey (USGS)

Summary of release

These Release Notes describe a major revision to the U.S. Geological Survey (USGS) **SUTRA** code (Voss, 1984). This revision adds the capability to conduct simulations in three spatial dimensions (3D) to the original two-dimensional (2D) capability. Utility codes for pre- and post-processing in 2D and 3D are provided with this new version of **SUTRA**. These codes are briefly mentioned herein and are described in more detail in the respective Release Notes for each code. Together, all of these utility codes and **SUTRA** are called **SutraSuite**.

The new **SUTRA** code (*Version 2D3D.1*) includes both 2D and 3D simulation capability. The 2D code represents the same hydrologic processes as the earlier version and is conceptually unchanged, and the 3D code represents the same processes as the 2D code but in three spatial dimensions. *Version 2D3D.1* was verified to correctly simulate all represented processes under saturated and unsaturated conditions in a number of tests that appear in the documentation.

Additional changes exist in the new code, primarily in the input and output formats. The new input structure is not compatible with former versions of **SUTRA**.

At present, information on **SUTRA** *Version 2D3D.1* includes these Release Notes and the revised **SUTRA** documentation (Voss and Provost, 2002), which may be downloaded from a US Geological Survey web site (http://water.usgs.gov/nrp/gwsoftware/sutra.html).

The distribution package for **SUTRA** *Version 2D3D.1* contains Fortran-90 source code and a Windows™ executable; this informational file; three example 2D simulations that are described in the documentation (including two from the earlier versions with updated input and output files); and input and output files for three new 3D simulation examples, one of which is described in the documentation. The examples also include setup files for the **SutraSuite** pre- and post-processors.

In this release, **SUTRA** is distributed as an executable code (compiled with Digital® Fortran Version 6.0.A) that runs under Microsoft Windows® XP, 2000, NT, or 95/98 operating systems.

Software Release Notice

SUTRA and related utility codes are public-domain software and are released to you cost-free by USGS for any purposes you choose. This is a general public release.

> **This software and related material (data and documentation) are made available to you by the USGS for any use. You may copy, modify, and distribute this software, any derivative works thereof, and its supporting documentation. You may provide services in connection with the software or derivative works thereof.**
>
> **Disclaimer: The software is provided "as is", and the authors, USGS, or the United States Government have made no warranty, expressed or implied, as to the accuracy or completeness of this software, and are not obligated to**

provide the user with any support, consulting, training or assistance of any kind with regard to the use, operation, and performance of this software nor to provide the user with any updates, revisions, new versions or "bug fixes".

The user assumes all risk for any damages whatsoever resulting from loss of use, data, or profits arising in connection with the access, use, quality, or performance of this software.

USGS much appreciates your support in reporting any problems that may occur so that the codes can be improved.

Utility codes

SUTRA is included with a number of utility codes for both pre- and post-processing (see table below) in a package called **SutraSuite**. The utility codes are described in detail in separate Release Notes that accompany each code.

The pre-processor for 2D problems is the **SutraGUI** (Winston and Voss, 2003, a revision of Voss and others, 1997). For 3D problems, there are two preprocessors: **SutraPrep**, a text-based preprocessor (Provost, 2002), and the **SutraGUI** (Winston and Voss, 2003).

There are three post-processors for 2D problems: **SutraPlot** (Souza, 1999), an upgrade of the former 2D version of **SutraPlot** (Souza, 1987); **ModelViewer** (Hsieh and Winston, 2002) currently restricted to plots for 2D fishnet meshes; and **SutraGUI**. There are two post-processors for 3D problems: **SutraPlot** creates rotatable 3D plots of the mesh, 2D contours of results, and plots of velocity vectors; **ModelViewer** creates rotatable 3D color visualizations of results and plots of velocity vectors.

Two additional post-processing tools for both 2D and 3D simulations are available. **GW_Chart** can graphically display **SUTRA** fluid, solute and energy budgets, as well as hydrographs for **SUTRA** observation node output, showing pressure, concentration, and temperature as a function of time (Winston, 2000). The utility **CheckMatchBC** aids in setting boundary condition conductances for a **SUTRA** simulation; it checks the match of specified and simulated pressures, concentrations and temperatures and reports the number of matching digits. **GW_Chart** and **CheckMatchBC** are distributed with the installation files for **SutraGUI**.

Examples in 2D and/or 3D for pre-processing and post-processing accompany the utility codes.

All of these codes are available free of charge; however, use of **SutraGUI** requires purchase of a commercial product, ArgusONE™ (http://www.argusint.com).

SutraSuite utilities

Utility code	Pre-processor	Post-processor	Description
SutraPrep	*3D*		Generates 3D, regular SUTRA meshes and creates input data files. *Instructions for using SutraPrep can be found in the documentation that accompanies the code.*
SutraGUI	*2D and 3D*	*2D*	Graphical environment for setting up and executing 2D and 3D SUTRA runs, and for visualizing results from 2D SUTRA runs. *Instructions for using SutraGUI can be found in the documentation that accompanies the code.*
SutraPlot		*2D and 3D*	Creates 2D and rotatable 3D plots from 2D and 3D SUTRA output data. *Instructions for using SutraPlot can be found in the documentation that accompanies the code.*
ModelViewer		*2D (fishnet meshes only) and 3D*	Creates rotatable 3D color visualizations and animations from 2D and 3D SUTRA output data. *Instructions for using ModelViewer can be displayed by accessing the Help menu.*
GW_Chart		*2D and 3D*	Graphical display of fluid, solute and energy budgets and observation node output, showing pressure, concentration, and temperature as a function of time (hydrographs). *Instructions for using GW_Chart can be displayed by accessing the Help menu.*
CheckMatchBC		*2D and 3D*	Checks match of specified and simulated pressures, concentrations and temperatures and reports number of matching digits. This aids in setting boundary condition conductances.

General Instructions

Except where stated otherwise, the following instructions assume that **SUTRA** has been installed into the default directory recommended during the installation process, c:\SutraSuite\SUTRA_2D3D_1. If a directory other than the default directory was selected, \SutraSuite will not be located directly under c:\, but the directory structure below \SutraSuite will remain as described below.

Instructions that involve mouse clicks refer to clicking on file names or icons displayed under the Windows Explorer file manager. For example, to "double-click on *file.txt*", use Windows Explorer to display the contents of the directory (folder) that contains the file *file.txt*, then click twice (in rapid succession) on the file name *file.txt* or its corresponding icon using the left mouse button. To "right-click", click once using the right mouse button.

Installation

After the **SUTRA** code (*Version 2D3D.1*) is unzipped, most of the associated files will be in a new directory, c:\SutraSuite\SUTRA_2D3D_1. This directory contains four subdirectories: \source, which contains the Fortran source files; \examples, which contains example datasets; \documentation, which contains this *ReleaseNotes.doc* file; and \bin, which contains the **SUTRA** executable file, *sutra_2D3D_1.exe*, and a shortcut to this executable file. Please note that the \examples subdirectory will be empty until the large zipped example files are separately downloaded and installed.

In c:\SutraSuite\SUTRA_2D3D_1, there is a file called *RunSUTRA.bat*, which is used to run the SUTRA code. *RunSUTRA.bat* contains the line

```
C:\SutraSuite\SUTRA_2D3D_1\bin\sutra_2D3D_1.exe
```

which provides the path for the **SUTRA** executable file, *sutra_2D3D_1.exe*. If **SUTRA** was installed into a directory other than the default directory, please make certain that this line contains the correct pathname to where *SUTRA_2D3D_1.exe* is located. Use a text editor to change the line if necessary.

Running SUTRA

To run **SUTRA** manually, create the file SUTRA filename assignments file, *SUTRA.FIL*, in the desired working directory. There are two ways to run **SUTRA**. 1- Double-click on the *RunSUTRA.bat* in the same directory. 2- Create a *Windows shortcut* to the *sutra_2D3D_1.exe* file in the directory in which you will work, and then double click the shortcut to run **SUTRA**. For both methods, a window will appear indicating that the **SUTRA** code is running.

SUTRA may also be run directly from the **SutraGUI** preprocessing utility and directions for doing this may be found in the **SutraGUI** documentation (Winston and Voss, 2003) and in some example simulations, including step-by-step instructions (see Rocky and Henry examples) provided in the directory \SUTRA_2D3D_1\examples.

New in SUTRA Version 2D3D.1

Version 2D3D.1 is essentially equivalent to former versions of the **SUTRA** code, except that the code is enhanced to allow true 3D simulation. A brief description of the main changes follows:

NEW PARAMETERS IN 3D

* For 3D simulation, all model and finite-element mesh attributes are assigned in an (x,y,z) Cartesian coordinate system. "z" is the additional (third) spatial coordinate. The input data required and formats are the same for 2D and 3D simulation (and quite similar to former versions).

* The new parameter Z replaces the parameter THICK, which in previous versions of SUTRA represented the mesh thickness. Z represents the mesh thickness for 2D simulations (i.e., it is equivalent to THICK) and the z-coordinate of a node for 3D simulations.

* 3D simulation requires an additional (third) component of the gravity vector.

* Assignment of permeability in 3D requires an additional principal value of permeability, PMID, in addition to PMAX and PMIN (which are used in 2D), and two additional angles used to aim the three perpendicular principal components of permeability.

* Dispersion in 3D requires the most complex addition to the input data. The dispersion model used is a 3D extension of the ad-hoc flow-direction-dependent model introduced for the original **SUTRA** code by Voss (1984) (and **SUTRA** update of 1990), which includes 2 longitudinal and 2 transverse values defined for flow in the MAX and MIN permeability directions. 3D requires an additional transverse dispersivity, AT2, acting perpendicular to the first transverse dispersivity, AT1 (formerly AT, which was the only one used in 2D). Further, each dispersivity, AL, AT1 and AT2, is allowed to vary in space depending on local flow direction relative to the principal permeability directions, requiring 3 values (ATmax, ATmid and ATmin) to be set. These 3 values define the semi-axes of a dispersivity ellipsoid aligned with the principal directions of permeability, from which the two transverse dispersivity values and principal directions are computed as functions of flow direction in a manner described in detail in the **SUTRA** documentation (Voss and Provost, 2002).

3D MESHES

* The finite elements used for 3d meshes are generalized hexahedra. These elements are the 3D version of the quadrilaterals used in 2D meshes. Each element is similar to a cube, and has 12 straight edges with a total of eight nodes, one at each vertex. However, the angles between edges do not need to be 90°, and the sides (faces) of an element do not need to be planes.

* To allow easier mesh construction in 3D, the 3D-mesh structure is restricted, in **SUTRA** *Version 2D3D.1*, to be logically rectangular. This 3D connectivity is the same as in a cube containing a regular 3D finite-difference grid (with nodes at the corners of the cells), but the geometry may be more complex. Each internal node (i.e., each node not on the exterior surface of the model domain) is shared by eight elements and is connected by straight element edges to six other nodes. Nodes (or elements) can be thought of as being logically arranged in rows, columns, and layers. However, unlike in a regular finite-difference mesh, the rows and columns need not lie along straight lines, and the layer need not be planar. At its most general, a 3D logically-rectangular mesh has the appearance of a warped 3D finite-difference mesh.

> - Irregularly-connected and logically-rectangular 2D meshes are allowed just as in previous versions of **SUTRA**.

> - At present, the matrix assembly process in **SUTRA** for 3D simulation is the only portion of code that assumes a logically-rectangular mesh, and the remainder of the code applies in general even

to irregular meshes. The restriction to regular meshes for 3D simulation will be relaxed in a later revision.

ITERATIVE SOLVERS

Because of the great computational effort required for direct solution of the matrix equation arising from the application of **SUTRA**'s numerical methods to most 3D meshes, a number of iterative matrix equation solvers are included with **SUTRA** *Version 2D3D.1*. These require considerably less storage and computational effort for large problems. The direct solver for banded matrix equations is still included, and although it will work for 3D meshes (given enough memory and processing time), it is primarily intended for use with 2D meshes.

The three iterative solvers presently included are part of **SLAP** (Sparse Linear Algebra Package; Seager, 1989). The solvers are:

* **CG**
     (IC-Preconditioned Conjugate Gradient for symmetric matrix equation)
     CG may be used only for solving **SUTRA**'s flow equation without upstream weighting.

* **GMRES**
     (ILU-Preconditioned Generalized Minimum Residual for non-symmetric matrix equation)
     GMRES may be used for solving **SUTRA**'s flow or transport equations.

* **ORTHOMIN**
     (ILU-Preconditioned)
     ORTHOMIN may be used for solving **SUTRA**'s flow or transport equations.

SUTRA *Version 2D3D.1* requires the user to select solvers for the flow and transport equations, and when selecting an iterative solver, to provide a convergence tolerance and maximum number of iterations allowed.

Direct solvers almost always work successfully in providing nearly exact answers to large matrix equations, but unfortunately they often take excessive amounts of storage and computational time in order to do so. Iterative solvers take considerably less time to solve large matrix equations, but they do not always succeed in providing an answer. Until fast, robust iterative solvers are developed, difficulties with non-convergence or slow convergence of iterative solutions will plague simulation in 3D. This is a universal problem by no means unique to the **SUTRA** code.

Users of **SUTRA** *Version 2D3D.1* must be prepared to not only spend effort on the hydrologic aspects of the problem they are studying via simulation as they previously have done for 2D, but must also be prepared to occasionally experiment with different strategies for coaxing the iterative solvers to provide good results. For well-discretized problems with appropriately formulated initial conditions, the iterative solvers have been found to be reasonably efficient and robust.

As testing and use of **SUTRA** continues, experience is being gained in how to obtaining good results from the **SLAP** solvers. Some experience-based guidelines relate to:

* Element elongation, parameter anisotropy and heterogeneity - Convergence difficulties in solving the flow equation are most likely to arise when finite elements are long and narrow (elongations of much more than 1000- or 10000-to-1), or when there are large anisotropies or heterogeneities in hydrologic properties, particularly permeability (several orders of magnitude). Ordering node numbers so that consecutive nodes are numbered across the narrow dimension of finite elements, across the direction of lowest permeability (for anisotropy), or across (rather than along) elongate lenses of different permeability, seems to improve convergence.

* Element shape – Elements in the mesh that are extremely 'sheared' may generate solution errors that, although minor in appearance when using a direct solver, may cause non-convergence of iterative solvers

for flow and transport in some models. In particular, vertex angles within a few degrees of 0° or 180° seem to cause such difficulties.

* Initial conditions as initial guesses - Good initial guesses for pressure and concentration or temperature help convergence considerably, particularly for steady-state simulations. When using the direct solver, the initial condition has absolutely no effect on the steady-state result or on the ability to obtain the result at all, and the steady solution is easily calculated given sufficient calculation time. When using an iterative solver, the initial condition affects convergence towards steady state because the solver uses it as an initial guess. For example, providing an approximate hydrostatic pressure distribution as an initial pressure guess for a 3D model is usually much better than giving an initial guess of uniform (e.g., zero) pressure.

* Experimenting with other approaches - Difficult pressure convergence was encountered for an example problem consisting of three intersecting fractures 1000 times more permeable than the host rock. A steady-state flow solution using an initial pressure guess of zero would not converge given thousands of iterations. The only approach found that allowed the iterative solver to obtain the solution was to run three consecutive simulations, using the pressure results of each as the initial condition for the next. In the first run, the permeability ratio was set to 10, in the second run it was increased to 100 and in the last run, it was increased to the desired value of 1000. The iterative solver converged with no difficulty for each of the runs. An alternative approach to aid solver convergence for steady-state pressure may be to increase the fluid and solid-matrix compressibilities to relatively high values and run a series of time steps until steady-state pressures are obtained. This may allow use of a poor initial guess.

FILE NAMES

There are now a number file types directly associated with the **SUTRA** and its utility codes. The filename extensions for all types are standardized and are listed in an accompanying table. These standard extensions must be used to ensure proper operation of the codes.

INPUT DATA

* Both input files (".inp" and ".ics") for SUTRA now consist primarily of unformatted data. This allows the user to decide on any columnization of data that seems useful. This also requires that character data be enclosed by quotes.

* Some data sets near the top of the main input file (".inp") have been rearranged and now appear in different order, while other major datasets (node, element, boundary conditions, and incidence) appear near to their previous locations in the file.

* Comment lines are now allowed before and after each individual dataset in both input files (".inp" and ".ics"), making the files easier to read.

* The main input file (".inp") now includes a description of the mesh type (2D or 3D, IRREGULAR, REGULAR or BLOCKWISE). For REGULAR or BLOCKWISE meshes, it also includes the number of nodes in each mesh direction, and for BLOCKWISE meshes, it also includes the number of elements in each strip of blocks in each direction.

* Matrix equation solvers are now selected in the main input file (".inp") for the flow solution and for the transport solution (either DIRECT, CG, GMRES or ORTHOMIN). Parameters are also provided for the iterative solvers.

* Previously, the initial conditions file (".ics") required values for each node even when the value was uniform for the entire mesh. Now, the initial conditions file may be shortened to only a few lines for the case of uniform initial conditions for pressure, concentration, or temperature, by specifying only a keyword (UNIFORM) and the constant value.

* The new format of the name file, SUTRA.FIL, that lists file types, assigned unit numbers, and filenames (in free format) is:

```
'INP'      50      'project.inp'
'ICS'      55      'project.ics'
'LST'      60      'project.lst'
'RST'      66      'project.rst'                    SUTRA.FIL
'NOD'      70      'project.nod'
'ELE'      80      'project.ele'
'OBS'      90      'project.obs'
'SMY'      95      'project.smy'
```

The order of SUTRA file types need not be exactly as shown here, but the filenames must have the indicated extensions. File types are described in the accompanying table of filename extensions. The unit numbers are user-selectable, and the ones given are only a suggestion. These should generally be chosen in the range 11-99 to avoid conflicts with pre-defined Fortran units. The "project" prefixes are also user-selectable, and these need not all be identical, though this is often desired to unify the naming of a simulation's input and output files.

OUTPUT
* The frequency of output for each output file may now be controlled independently. The order and content of columns in the columnar nodewise and elementwise output files is now controlled by data in the main input file (".inp").

* The observation node output from a **SUTRA** run has been moved from the main output file to a separate observation file with extension ".obs".

* The pre-named SUTRA.ERR file, which, in previous versions of **SUTRA**, had reported only errors in file handling, now is named *project.smy* or (by default) *SUTRA.SMY*. This file keeps a simulation summary, which now also includes a running report of the simulation's progress, including a convergence log for each solver iteration on each time step.

NO PINCH NODES
Pinch nodes were rarely employed by **SUTRA** users and have not been implemented in 3D. The possibility of using pinch nodes in 2D mesh design has also been removed.

Documentation
These Release Notes and the **SUTRA** manual (Voss and Provost, 2002) are the primary documentation of the code. The **SUTRA** manual is available, free of charge, in electronic (*pdf*) format from the USGS website http://water.usgs.gov/nrp/gwsoftware/sutra.html.

3D verification tests
Version 2D3D.1 has been tested on a variety of problems including 3D versions of original 2D problems. These include: the Theis solution in 3D, the Rocky Mountain Arsenal problem extruded in each 3D direction, the Henry seawater intrusion problem extruded in each 3D direction, consistency tests for variable density (Voss and Souza, 1987), a 3D version of the Elder density-driven flow problem, and a problem comparing 2D and 3D solutions to a saturated-unsaturated, variable-density flow problem on a circular island that is described in the **SUTRA** documentation (Voss and Provost, 2002).

2D examples provided

Three example 2D simulations are provided in sub-directories of the directory SutraSuite\SUTRA_2D3D-1\examples. All input and output files for each problem are included there. (The examples may be run manually [see 'RUNNING SUTRA', above] or may be recreated using the pre-processor described in the *ReadMe* file that accompanies each example and run through the pre-processor.) It is recommended that the utility and input files be copied to a new working directory for testing runs, so that the user's new **SUTRA** output does not overwrite the existing output files that were downloaded. The 2D examples include the Rocky Mountain Arsenal contaminant plume problem (\Rocky), and the Henry seawater intrusion problem (\Henry). The input files were created with the utility **SutraGUI**. Information on the setup of the 2D examples may be found in the **SUTRA** documentation (Voss and Provost, 2002) and in the respective *ReadMe* files that accompany each example. Users are also encouraged to download, install, and try using **SutraPlot** and **Model Viewer** to view results of these runs. Creation of the input files and post-processing using **SutraGUI** is described in the **SutraGUI** Release Notes. In addition, the Island2D example is provided for comparison with the 3D version described just below.

3D examples provided

Three example 3D simulations are provided in sub-directories of the directory, SutraSuite\Sutra_2D3D_1\examples. These may be found in sub-directories \Bf , \Pond, and \Island3D. All input and output files for each problem are included there. (The examples may be run manually [see 'RUNNING SUTRA', above] or may be recreated using the pre-processor described in the *ReadMe* file that accompanies each example and run through the pre-processor). The island problem is described in the SUTRA documentation (Voss and Provost, 2002), and the other two problems are described in the ReadMe file that accompanies each example. It is recommended that the utility and input files be copied to a new working directory for testing runs so that the user's new **SUTRA** output does not overwrite the existing output files that were downloaded. Users are encouraged to download, install, and try using both **SutraPlot** and **Model Viewer** to view results of these runs. (Each of these tools also comes with examples.)

References

Burnett, R.D., and Frind, E.O., 1987, Simulation of contaminant transport in three dimensions 2. dimensionality effects, Water Resources Research, 23(4), 695-705.

Hsieh, P.A., and Winston, R.B., 2002, User's guide to **ModelViewer**, a program for three-dimensional visualization of ground-water model results: U.S. Geological Survey Open-File Report 02-106, 18 p. http://water.usgs.gov/nrp/gwsoftware/modelviewer/ModelViewer.html

Provost, A.M., 2002, **SutraPrep**, a pre-processor for SUTRA, a model for ground-water flow with solute or energy transport: U.S. Geological Survey Open-File Report 02-376, 43 p. http://water.usgs.gov/nrp/gwsoftware/sutra.html

Seager, M.K., 1989, A SLAP for the Masses: *in* Parallel Supercomputing: Methods, Algorithms and Applications, Carey, G. F. (editor): Wiley, p.135-155.

Souza, W.R., 1999, **SutraPlot**, a graphical post-processor for SUTRA, a model for ground-water flow with solute or energy transport: U.S. Geological Survey Open-File Report 99-220, 30 p. http://water.usgs.gov/nrp/gwsoftware/sutra.html

Voss, C. I., 1984, **SUTRA**, A finite-element simulation model for saturated-unsaturated fluid density-dependent ground-water flow with energy transport or chemically-reactive single-species solute transport, U.S. Geological Survey Water-Resources Investigations Report 84-4369, 409 p.

Voss, C. I., and Provost, A.M., 2002, **SUTRA**, A model for saturated-unsaturated variable-density ground-water flow with solute or energy transport, U.S. Geological Survey Water-Resources Investigations Report 02-4231, 250 p. http://water.usgs.gov/nrp/gwsoftware/sutra.html

Voss, Clifford I., Boldt, David, and Shapiro, Allen, 1997, **SUTRA-GUI**, A graphical-user interface for the U.S. Geological Survey's SUTRA code using ArgusONE, U.S. Geological Survey Open-File Report 97-421, 106p.

Winston, R.B., 2000, Graphical User Interface for MODFLOW, Version 4: U.S. Geological Survey Open-File Report 00-315, 27 p. http://water.usgs.gov/nrp/gwsoftware/modflow.html

Winston, R.B. and Voss, C.I., 2003, **SutraGUI**, a graphical-user interface for SUTRA, a model for ground-water flow with solute or energy transport: U.S. Geological Survey Open-File Report 03-285, 114 p. http://water.usgs.gov/nrp/gwsoftware/sutra.html

FILENAME EXTENSIONS for SUTRA *Version 2D3D.1*

Version / Code	*1990* (2D) *version*	1997 (2D) version	**2D3D.1** **current version**	*Content*
SUTRA *input*	SUTRA.FIL .D5 .D55	SUTRA.FIL .K1 or .D5 .K2 or .D55	SUTRA.FIL .inp .ics	file assignments main input initial conditions
output	SUTRA.LOG SUTRA.ERR .D6 .D66 .D3 .D4	SUTRA.LOG SUTRA.ERR .K3 or .D6 .K4 or .D66 .K5 or .D3 .K6 or .D4	.smy (defaults to SUTRA.SMY) .lst .rst .nod .ele .obs	simulation summary or file assignment errors main results listing restart file nodewise results elementwise results observation results
SutraPREP *input*			.prp .inp .prl	main input existing SUTRA .inp file SutraPREP log
output			.inp .ics .wrl	 3D blocks or mesh in VRML
SutraGUI *input*		.K5 or .D3 .K6 or .D4	.nod .ele	
output		.K1 or .D5 .K2 or .D55	.inp .ics	
SutraPLOT *input*	.D5 .D3 .D4	.D5 .D3 .D4	.inp .nod .ele	
output			.gr2 .gr3 .spl	2D grid of results 3D grid of results plot layout
ModelViewer *input*			.inp .nod .ele	
output			.mvb .wrl .bmp	Binary version of .nod & .ele 3D view of image bitmap of image
GW_Chart *input*			.obs .lst	For hydrographs For budget charts
output			.wmf	Saved image
CheckMatchBC *input*			.inp .nod	
output				

(Uppercase names are not user selectable.)