THE DEPARTMENT OF DEFENSE

Groundwater Modeling System

GMS *v3.0*

File Formats

GMS *3.0 File Formats*

GMS is a product of the Environmental Modeling Research Laboratory (EMRL) of Brigham Young University.

www.emrl.byu.edu

Last Revision: November 17, 1999

TABLE OF CONTENTS

GMS

File Formats

This document contains the file formats for most of the files used by GMS. Files that are used by analysis codes such as FEMWATER and MODFLOW are not described in this document since they are described in the documentation for the codes.

1 Card Style Input

Most of the files used by GMS use a "card" type format. With this format, the components of the file are grouped into logical groups called "cards." The first component of each card is a short name which serves as the card identifier. The remaining fields on the line contain the information associated with the card. In some cases, such as lists, a card can use multiple lines.

There are many advantages associated with the card type approach to formatting files. Some of the advantages are:

1. Card identifiers make the file easier to read. Each input line has a label which helps to identify the data on the line.

2. The cards names are useful as text strings for searching in a large file. All input lines of a particular type can be located quickly using a find command in a text editor.

3. Cards allow the data to be input in any order in many cases, i.e., the order that the cards appear in the file is usually not important.

4. Cards make it easy to modify a file format. New data can be included simply by defining a new card type. If the new card is optional (which is often the case for new cards) old files are still compatible. If an old card type is no longer used, the card can simply be ignored without causing input errors.

2 File Extensions

Each of the files used in a GMS modeling project must have a unique extension. The extensions recognized by GMS are summarized in Table 1. The list includes files native to GMS, files that can be imported to or exported from GMS, and files associated with models such as MODFLOW and FEMWATER.

OPEN FILES		**MODFLOW**	
GMS Project File	gpr	MODFLOW Super File	mfs
Super Files	sup	Basic File	bas
Map Files	map	BCF File	bcf
TIN Files	tin	River File	riv
Material Files	mat	Well File	wel
Borehole Files	bor	Drain File	drn
Solid Files	sol	HFB File	hfb
2D Mesh Files	2dm	Stream File	str
2D Grid Files	2dg	Specified Head File	chd
2D Scatter Point Files	xy	General Head File	ghb
3D Mesh Files	3dm	Recharge File	rch
3D Grid Files	3dg	Evapotranspiration File	et
3D Scatter Point Files	xyz	Strongly Implicit Proc.	sip
Image Files	img	Slice Succ. Over-Rel	sso
Data Set Files	dat	Pre-Cond. Conj. Grad.	pcg
GMS Data Set Super File	dss	Output Control File	oc
Settings File	ini	Cell Centered Flow File	ccf
Cross Section Files	xsc	Head File	hed
Material Set Files	tjb	Drawdown File	drw
MODFLOW Super File	mfs	True Layer File	lay
MODPATH Response File	rsp	**MODPATH**	
MT3D Super File	mts	MODPATH Response File	rsp
RT3D Super File	rts	Main File	mdf
SEAM3D Super File	sms	Name File	nam
UTCHEM Super File	uts	Starting Locations	loc
NUFT File	nft	Time Data	tim
FEMWATER Super File	fws	Composite Budget File	cbd
SEEP2D Super File	sps	Endpoint File	ept
IMPORT FILES		Path Line File	pth
Hogentogler Sample Data	int	Time series	ts
SCAPS Sample Data	sfx	Summary File	sum
SCAPS 2 Sample Data	scn	**MT3D/RT3D/SEAM3D**	
ARC/INFO Ascii Grid	asc	MT3D Super File	mts
Grass Grid	ggd	RT3D Super File	rts
Surfer Ascii Grid	grd	SEAM3D Super File	sms
GMS Sample Data	gsd	Biodegradation File	bio
Tabular Scatter Point -2D	sp2	NAPL File	npl
Tabular Scatter Point -3D	sp3	Basic Transport File	btn
Tabular Observation Point	top	Output Control	out
DXF Files	dxf	GCG Solver File	gcg
Tiff Files	tif	Advection Package File	adv
EXPORT FILES		Dispersion Package File	dsp
ARC/INFO Ascii Grid	asc	Source/Sink Mixing File	ssm
Grass Grid	ggd	Reaction Package File	rct
Shapefile	shp	Head Flow File	hff

Tabular Scatter Point -2D	sp2	**UTCHEM**	
Tabular Scatter Point -3D	sp3	UTCHEM Super File	chs
DXF File	dxf	Super File	uts
TIFF File	tif	Input File	utc
UTEXAS Pore Pressures File	upp	**NUFT**	
UTEXAS Profile Lines File	upl	NUFT Super File	nft
SEEP2D		State File	sta
SEEP2D Super File	sps	**OTHERS**	
Input File	s2d	Pest Parameter File	par
Geometry File	geo	Range File	ran
Solution File	ssl	Concentration File	con
FEMWATER		XY Series File	xys
FEMWATER Super File	fws	GMS Path Line File	gpt
BC File	3bc	Time Series File	ts
Pressure Head File	phd	Animation File	alp
Moisture File	mcn	World Coordinates File	tfw
Velocity File	vel	Film Loop File	avi
Flux File	flx	Variogram File	var

Table 1 File Extensions Recognized by GMS.

3 Project Files

All of the data in modeling project can be saved at once using the *Save* commands in the *File* menu. The project is saved to a set of files including a grid file, a map file, and data set files. All of these files are organized using a project file. The project file contains a list of each of the files associated with the project. An entire project can be read back into GMS by selecting the Open command in the File menu and selecting the project file.

A sample project file is shown in Figure 3.1. The first line in the file is a PROJECT card that identifies the file as a super file. Each of the other cards contains a file type identifier and a filename. Some file types can be listed multiple times. For example, a project may contain several RT3D solution files (RT3DSOL "filename").

```
PROJECT
MODFLOW  "./modf/run1.mfs"
MODFSOL  "./modf/run1.*"
MT3D     "./mt3d/mod1.mts"
MT3DSOL  "./mt3d/mod1.*"
MODPATH  "./modpath/p3.rsp"
MODSOL   "./modpath/p3.*"
SEEP2D   "./seep2d/dam1.sps"
SEEPSOL  "./seep2d/dam1.*"
FEMWATER "./femw/sim1.fws"
FEMWSOL  "./femw/sim1.*"
NUFT     "./nuft/run2.nfs"
NUFTSOL  "./nuft/run2.*"
UTCHEM   "./utchem/mwell.utc"
UTCHSOL  "./utchem/mwell.*"
MAT      "redriver.mat"
BHOLE    "redriver.bor"
SOLID    "redriver.sol"
MAP      "redriver.map"
IMAGE    "redriver.img"
TIN      "redriver.tin"
MESH2D   "redriver.2dm"
MESH3D   "redriver.3dm"
GRID2D   "redriver.2dg"
GRID3D   "redriver.3dg"
SCAT2D   "redriver.xy"
SCAT3D   "redriver.xyz"
XSECT    "redriver.sec"
DATA     "redrivertn.dat"
DATA     "redriver2m.dat"
DATA     "redriver3m.dat"
DATA     "redriver2g.dat"
DATA     "redriver3g.dat"
DATA     "redriver2s.dat"
DATA     "redriver3s.dat"
PLOT     "redriver.plt"
DXF      "./drawings/rivers.dxf"
DXF      "./drawings/roads.dxf"
STNGS    "redriver.ini"
```

Figure 3.1 Sample Project File.

4 Material Files

Each of the basic object types supported in GMS (TINs, solids, boreholes, etc.) refer to materials using material ids. A material id represents an index to a global list of materials. The material file associates general attributes such as a name, color, and pattern with each of the materials. The format for a material file is shown in Figure 4.1. A sample material file is shown in Figure 4.2

```
MAT      /* File type identifier */
MN id "name"     /* Material name */
MS id stippleid  /* Material stipple (fill pattern) */
OPAQ id opacity  /* Material opacity */
MC id red green blue    /* Material color */
```

Figure 4.1 Material File Format.

```
MAT
MN       3       "silt"
MS       3       0
MC       3       158     35      35
OPAQ     3       1.0
MN       5       "Sandstone Boulders"
MS       5       0
MC       5       254     254     102
OPAQ     5       1.0
```

Figure 4.2 Sample Material File.

Each card in the material file represents an attribute for a material. The attribute cards can be repeated as many times as necessary to define each material being used. The cards used in the material file are as follows:

Card Type	**MAT**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**MN**		
Description	Identifies a name to be associated with the material.		
Required	NO		
Format	`MN id "name"`		
Sample	`MN 5 "bedrock"`		
Field	*Variable*	*Value*	*Description*
1	`id`	+	The id of the material.
2	`name`	str	The name of the material

Card Type	**MC**		
Description	Identifies a color to be associated with the material.		
Required	NO		
Format	`MN id red green blue`		
Sample	`MN 5 124 67 245`		
Field	*Variable*	*Value*	*Description*
1	`id`	+	The id of the material.
2	`red`	0-255	The value of the red component of the color.
3	`green`	0-255	The value of the green component of the color.
4	`blue`	0-255	The value of the blue component of the color.

Card Type	**MS**		
Description	Identifies a stipple (fill pattern) to be associated with the material. This stipple is used whenever an object is being drawn using color filled polygons.		
Required	NO		
Format	`MN id stippleid`		
Sample	`MN 5 13`		
Field	*Variable*	*Value*	*Description*
1	`id`	+	The id of the material.
2	`stippleid`	+	The id of the stipple.

Card Type	**OPAQ**

Description	Identifies opacity to be associated with the material. This opacity is used whenever an object is raytraced.		
Required	NO		
Format	`OPAQ id opacity`		
Sample	`OPAQ 1 0.85`		
Field	*Variable*	*Value*	*Description*
1	`id`	+	The id of the material.
2	`opacity`	0-1	Opacity value. 0 = transparent, 1 = opaque.

5 Material Set Files

When using the indicator simulation option for Kriging, a set of material ids is generated. Each set represents a unique array of material ids for the 3D grid. These material ids are saved to a material set file. A material set file is a binary file that is similar to a binary data set file. A separate material set file is saved for each simulation. The format of the material set file is shown in Table 2.

Card	**Item**	**Size**	**Description**
	version	4 byte integer	The GMS binary material set file format version. value = 4000.
100	objecttype	4 byte integer	Identifies the type of objects that the data sets in the file are associated with. Options are as follows: 7 3D grids
110	SINT	4 byte integer	The number of bytes that will be used in the remainder of the file for each integer value (4, 8, or 16).
120	SFLG	4 byte integer	The number of bytes that will be used in the remainder of the file for status flags (1, 2, or 4)
180	NUMCELLS	4 byte integer	This number should correspond to the number of cells in the grid.
190	NAME	40 bytes	The name of the simulation. Use one character per byte. Mark the end of the string with the '\0' character.
130	BEG		Marks the beginning of a set of cards defining a material data set.
220	ACT	SFLT real	Marks the active material set
160	IDCARD	4 byte integer	
	val1	INT real	Integer value for item 1
	val2	INT real	integer value for item 2
	….		
210	ENDSET		Signals the end of a material set.

Table 2 File Format for Material Set Files.

6 TIN Files

TIN files are used for storing triangulated irregular networks. Multiple TINs can be stored to a single file. The TIN file format is shown in Figure 6.1. A sample TIN file is shown in Figure 6.2.

```
TIN       /* File type identifier */
BEGT      /* Beginning of TIN group */
ACTIVETIN /* Marks the active TIN */
ID id     /* The id of the TIN */
TNAM "name"      /* Name of TIN */
MAT id    /* TIN material id */
VERT nv   /* Beg. of vertices */
x1 y1 z1 lf1     /* Vertex coords. */
x2 y2 z2 lf2
.
.
.
xnv ynv znv lfnv
TRI nt    /* Beg. of triangles */
v11 v12 v13     /* Triangle vertices */
v21 v22 v23
.
.
.
vnt1 vnt2 vnt3
ENDT      /* End of TIN group */
/* Repeat TIN group for other TINs */
```

Figure 6.1 TIN File Format.

```
TIN
BEGT
ACTIVETIN
ID  25463
TNAM "TopClay4.gr"
MAT 2
VERT 19
625.00000000       400.00000000        1211.30005000          0
700.00000000       450.00000000        1212.50000000          0
775.00000000       550.00000000        1213.90002000          0
.
.
.
625.00000000       550.00000000        1214.69995000          0
550.00000000       475.00000000        1211.30005000          0
590.63000000       442.49899000        1215.02002000          0
TRI 21
  8       9       14
  6       8       14
  6       7        8
.
.
.
 12      13       16
  1      19       11
 17      19       12
ENDT
```

Figure 6.2 Sample TIN file.

The cards used in the TIN file are as follows:

Card Type	**TIN**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**BEGT**
Description	Marks the beginning of a group of cards describing a TIN. There should be a corresponding ENDT card at a latter point in the file. No fields.

Required	YES

Card Type	**ACTIVETIN**		
Description	Marks the active TIN		
Required	NO		

Card Type	**ID**		
Description	Provides an id to be associated with the TIN.		
Required	YES		
Format	`ID id`		
Sample	`ID 153`		
Field	*Variable*	*Value*	*Description*
1	id	+	The id of the TIN.

Card Type	**TNAM**		
Description	Provides a name to be associated with the TIN.		
Required	NO		
Format	`TNAM "name"`		
Sample	`TNAM "top of bedrock"`		
Field	*Variable*	*Value*	*Description*
1	name	str	The name of the TIN.

Card Type	**MAT**		
Description	Associates a material id with the TIN. This is typically the id of the material which is below the TIN.		
Required	NO		
Format	`MAT id`		
Sample	`MAT 3`		
Field	*Variable*	*Value*	*Description*
1	id	+	The material id.

Card Type	**VERT**		
Description	Lists the vertices in the TIN		
Required	YES		
Format	VERT nv x_1 y_1 z_1 lf_1 x_2 y_2 z_2 lf_2 . . x_{nv} y_{nv} z_{nv} lf_{nv}		
Sample	VERT 4 0.0 3.1 7.8 0 5.3 8.7 4.0 1 2.4 4.4 9.0 1 3.9 1.2 3.6 0		
Field	*Variable*	*Value*	*Description*
1	nv	+	The number of vertices in the TIN
2-4	x,y,z	±	Coords. of vertex
5	lf	0,1	Locked / unlocked flag for vertex (optional). 0=unlocked, 1=locked. Repeat fields 2-5 nv times.

Card Type	**TRI**

Description	Lists the triangles in the TIN		
Required	NO (A TIN may contain only vertices)		
Format	`TRI nt` v_{11} v_{12} v_{13} v_{21} v_{23} v_{23} . . v_{nt1} v_{nt2} v_{nt3}		

Sample	`TRI 4`
	`5 1 4`
	`4 1 2`
	`4 2 3`
	`5 4 3`		

Field	*Variable*	*Value*	*Description*
1	`nt`	+	The number of triangles in the TIN.
2-4	v_1,v_2,v_3	+	Vertices of triangle listed in a counter-clockwise order. Repeat `nt` times.

Card Type	**ENDT**
Description	Marks the end of a group of cards describing a TIN. There should be a corresponding BEGT card at a previous point in the file. No fields.
Required	YES

7 Borehole Files

A borehole file contains stratigraphy data and may include sample data. However, it is generally more efficient to use the borehole file to import stratigraphy data and use one of the formats described in Section 8 to import sample data.

Multiple holes can be defined in a single file. The borehole stratigraphy is defined as a list of contacts representing the boundaries between different materials. The contacts are defined by XYZ coordinates and an associated material. Sample data are defined as sample points and data sets. Each sample point is defined by an XYZ coordinate and a value for each data set. The borehole file format is shown in Figure 7.1. A sample borehole file is shown in Figure 7.2.

```
BHOLE     /* File type identifier */
DISW min max     /* Min and Max borehole display widths */
DSETS nds "nameds1" "nameds2"... "namedsnds"  /* Data set info */
ACTDS "name"
BEGH      /* Beginning of borehole group */
BNAM "name"      /* Name of borehole */
ID id     /* Unique identification number for the hole */
WTE elev  /* Distance from top of hole to watertable */
CONT nc   /* Beg. of contacts */
x1 y1 z1 matid   /* Contact coords and material id */
x2 y2 z2 matid
.
.
xnc ync znc matid
SAMPPTS np       /* Number of sample points */
x1 y1 z1 d1,1 d1,2 ...d1,nds  /* Point coordinates and data values */
x2 y2 z2 d2,1 d2,2 ...d2,nds
.
.
xnp ynp znp dnp,1 dnp,2 ...dnp,nds
ENDH      /* End of borehole group */
/* Repeat borehole group for other boreholes */
```

Figure 7.1 Borehole File Format.

```
BHOLE
DISW 1.00000000e+000    3.00000000e+000
DSETS 11 "Depth" "Qc(avg)" "Fs(avg)" "Rf"
ACTDS "Rf"
BEGH
BNAM "L1-2-94"
ID 684
WTE 0.00000000e+000
CONT 7
  -7.12509100e+001      -2.45159017e+001      0.00000000e+000       1
  -7.12509100e+001      -2.45159017e+001      -4.03235766e-001      2
  -7.12509100e+001      -2.45159017e+001      -2.03983567e+000      3
  -7.12509100e+001      -2.45159017e+001      -4.21363727e+000      2
  -7.12509100e+001      -2.45159017e+001      -4.99160542e+000      1
  -7.12509100e+001      -2.45159017e+001      -1.34317221e+001      2
  -7.12509100e+001      -2.45159017e+001      -1.71000000e+001      1
SAMPPTS 342
-2.45159017e+001 -5.00000000e-002  5.00000007e-002 1.64000005e-001
-2.45159017e+001 -1.00000000e-001  1.00000001e-001 3.28000009e-001
.
.
.
ENDH
```

Figure 7.2 Sample Borehole File.

The cards used in the borehole file are as follows:

Card Type	**BHOLE**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**DISW**		
Description	Borehole min and max display widths.		
Required	NO		
Format	`DISW min max`		
Sample	`DISW 1.0 3.0`		
Field	*Variable*	*Value*	*Description*

1	min	+	Minimum borehole display width.
2	max	+	Maximum borehole display width.

Card Type	**DSETS**		
Description	Lists the number of data sets for the sample data and the data set names.		
Required	NO		
Format	`DSETS nds "nameds1" "nameds2"... "namedsnds"`		
Sample	`DSETS 4 "Depth" "Cone" "Sleeve" "Soil Class."`		
Field	*Variable*	*Value*	*Description*
1	nds	+	The number of data sets for the borehole.
2	nameds	str	The names of the data sets.

Card Type	**ACTDS**		
Description	Identifies the active data set.		
Required	NO		
Format	`ACTDS "name"`		
Sample	`ACTDS "depth"`		
Field	*Variable*	*Value*	*Description*
1	name	str	The name of the active data set. This name must be one of the names listed on the DSETS card.

Card Type	**BEGH**
Description	Marks the beginning of a group of cards describing a borehole. There should be a corresponding ENDH card at a latter point in the file. No fields.
Required	YES

Card Type	**BNAM**		
Description	Provides a name to be associated with the borehole.		
Required	NO		
Format	`BNAM "name"`		
Sample	`BNAM "hole #1"`		
Field	*Variable*	*Value*	*Description*
1	name	str	The name of the borehole.

Card Type	**ID**		
Description	Provides a unique id number to be associated with the borehole.		
Required	NO		
Format	`ID id`		
Sample	`ID 5782`		
Field	*Variable*	*Value*	*Description*
1	id	+	The id number of the borehole.

Card Type	**WTE**		
Description	Defines a water table elevation for the borehole.		
Required	NO		
Format	`WTE elev`		
Sample	`WTE 343.5`		
Field	*Variable*	*Value*	*Description*
	elev	±	The water table elevation.

Card Type	**CONT**		
Description	Lists the contacts in the borehole.		
Required	YES. At least two contacts must be supplied per hole.		
Format	CONT nc x_1 y_1 z_1 matid x_2 y_2 z_2 matid . . . x_{nc} y_{nc} z_{nc} matid		
Sample	CONT 4 10.0 30.1 210.7 1 10.0 30.1 208.2 4 10.0 30.1 200.4 3 10.0 30.1 195.0 6		
Field	*Variable*	*Value*	*Description*
1	nc	+	The number of contacts in the borehole.
2-4	x,y,z	±	Coords. of contact. The contacts should be listed in order from the top to the bottom of the hole.
5	matid	+	Material id of the soil/rock below the contact.
Repeat 2-5 nc times.			

Card Type	**SAMPPTS**		
Description	Lists the number of sample points and their location and data set values. The points should be listed in order of decreasing elevation starting at the top of the borehole.		
Required	NO		
Format	SAMPPTS np x1 y1 z1 d1,1 d1,2 ...d1,nds x2 y2 z2 d2,1 d2,2 ...d2,nds . . xnp ynp znp dnp,1 dnp,2 ...dnp,nds		
Sample	SAMPPTS 4 0.0 0.0 0.0 1 20 .01 100 0.0 0.0 -0.1 4 28 .03 110 0.0 0.0 -0.2 3 24 .05 107 0.0 0.0 -0.3 6 19 .02 99		
Field	*Variable*	*Value*	*Description*
1	np	+	The number of sample points for the borehole.
2-4	x,y,z	±	Coordinates of the sample point.
5	d	±	Data set values for the sample point.
Repeat 2-5 np times.			

Card Type	**ENDH**

Description	Marks the end of a group of cards describing a borehole. There should be a corresponding BEGH card at a previous point in the file. No fields.
Required	YES

8 Importing Borehole Sample Data

Borehole files can contain either stratigraphy data or sample data or both. Sample data typically represents cone penetrometer or geophysical data. One method of importing sample data is to prepare the sample data using the standard GMS borehole file format described in Section 7. Another method is to import the sample data using the *Import* command in the *File* menu. The *Import* command supports three options for importing cone penetrometer or geophysical data. The three options are Hogentogler files (*.int), SCAPS files (*.sfx and *.scn), and GMS Sample Data (*.gsd) files. Each of the file types is identified by the file extension.

8.1 Hogentogler Files

Hogentogler files contain cone penetrometer data. A sample Hogentogler file is shown in Figure 8.1. The information at the beginning of the file is ignored by GMS. Only the columns of data are read. Each column becomes a data set.

With the Hogentogler format, each cone test is contained in a separate file. As each file is imported to GMS, the user is prompted for the XYZ coordinates of the top of the hole. This information is used to assign XYZ coordinates to each of the sample points for display purposes.

The file extension for all Hogentogler files should be *.int.

```
"   Output file from CPTINT - Version 4.1
"   =====================================
"   INPUT FILE: A:L1-1-94.LAP
"   -------------------------
"
"   Interpreter Name: ESS
"   File Number: CPT-23                     Date: 10-05-94 13:44
"   Operator: CROSBY                        On Site Location: L1-1-94.LAP
"   Cone Type: 342                          Comment: LAAP
"
"   SUMMARY SHEET
"   -------------
"   'a' for calculating Qt:                 0.750
"   Value for Water Table (in m):           1.500
"   Method for Friction Angle:              Robertson & Campanella
"   Method for calculating Su:              Nk
"   Value of the constant Nk:               12.000
"
"   Soil Behavior Type Zone Numbers
"   For Rf Zone & Bq Zone Classification
"   ------------------------------------
"   Zone #1 = Sensitive fine grained      Zone #7  = Silty sand to sandy silt
"   Zone #2 = Organic material            Zone #8  = Sandy to silty sand
"   Zone #3 = Clay                        Zone #9  = Sand
"   Zone #4 = Silty clay to clay          Zone #10 = Gravelly sand to sand
"   Zone #5 = Clayey silt to silty clay   Zone #11 = Very stiff fine grained *
"   Zone #6 = Sandy silt to clayey silt   Zone #12 = Sand to clayey sand *
"   * Overconsolidated and/or cemented
"
"   NOTE:
"   -----
"   For soil classification, Rf values greater than 8 are assumed to be 8.
"
"   NOTE:
"   -----
"   Since U2 (pore pressure) has not been defined, Qt cannot be calculated,
"   therefore, the value of Qt has been made equal to Qc.
"
"   ( Note: 9E9 means Out Of Range )
"
"---|  INPUT FILE: A:L1-1-94.LAP  |------------------------------------------------------------
---------------
" Depth    Depth    Qc(avg)  Fs(avg)  Rf       Rf(Norm) Rf Zone  Spt N    Spt N1   Phi      Su
" (meter)  (feet)   (TSF)    (TSF)    (%)     (%)      (zone #) (blow/ft) (blow/ft) (degree) (psf)
"---------------------------------------------------------------------------------------------------
  0.050    0.164    37.600   0.262    0.697    0.697    7        12       18       9E9      9E9
  0.100    0.328    65.700   0.447    0.680    0.681    8        16       24       9E9      9E9
  0.150    0.492    61.300   0.506    0.825    0.826    8        15       23       9E9      9E9
.
.
.
```

Figure 8.1 The Hogentogler File Format.

8.2 SCAPS Files

SCAPS files also contain cone penetrometer data. Two types of SCAPS file formats are supported. The *.sfx format and the *.scn format. Both types are imported using the *Import* command in the *File* menu. GMS automatically determines the type of file based on the contents of the file.

A sample *.sfx SCAPS file is shown in Figure 8.2. As with the Hogentogler files, each cone test is contained in a separate file. As each file is imported to GMS, the user is prompted for the XYZ coordinates of the top of the hole.

```
  CPT         Depth        Cone        Sleeve     Soil Class.
               (ft)        (tsf)       (tsf)
-------------------------------------------------------------
laap-d1        1.20        16.20        3.25         0.76
laap-d1        1.30       134.21        3.23         1.00
laap-d1        1.40       190.60        4.35         1.00
laap-d1        1.50       198.25        4.56         1.00
laap-d1        1.60       194.60        4.60         1.00
laap-d1        1.70       184.45        4.51         1.00
laap-d1        1.80       163.80        4.41         1.00
.
.
.
```

Figure 8.2 The SCAPS (.sfx) File Format.*

A sample *.scn SCAPS file is shown in Figure 8.3. Again, each cone test is contained in a separate file. The first number represents the number of points, the probe value is the name of the hole, and the easting, northing, and elevation values represent the x, y, and z coordinates of the top of the hole. The column headings are not read by GMS but are hard-coded to the names shown. The depth column actually represents elevation values.

```
156
Probe          =  2-219
Easting(ft.)    =      21.00
Northing(ft.)   =     307.00
Elevation(ft.)   =     199.22
     depth    cone        sleeve      soil clsfn     resistivity
-----------------------------------------------------------------
    198.12      0.00        0.48          0.00           0.00
    198.02      0.00        0.76          0.00           0.00
    197.92      0.00        1.21          0.00           0.00
    197.82      0.00        1.38          0.00           0.00.
.
.
```

Figure 8.3 The SCAPS (.scn) File Format.*

8.3 GMS Sample Data Files

Although GMS can directly import *Hogentogler* and *SCAPS* cone penetrometer files, there are several other file formats that are used with sample data. The GMS *Sample Data File* format has been designed to make it easy to put any existing sample data file into a format that is recognizable by GMS (Figure 8.4). Since sample data are usually written out in columns, the GMS sample data file lists the sample data in columns. The points should be listed in order of decreasing elevation starting at the top of the borehole. A few cards at the top of the file define the borehole name and XYZ location, number of columns and number of rows. The points are then listed, one per row. The first column should always represent the depth of the sample points from the top of the hole.

Typically, a spreadsheet application is used to put an existing sample data file into the GMS sample data file format. The file can be either space, comma or tab delimited. Only one hole can be represented per file. An example GMS sample data file is shown in Figure 8.5.

The filename used for a GMS sample data file should always have the extension *.gsd.

```
NAME "name"      /* Borehole name */
XYZ x y z/* xyz loc. of the top of the hole */
NUMCOLS ncol     /* # cols of data (not inc. xyz) */
NUMROWS nrow     /* Number of rows of data */
"name1" "name2" ... "namencol"      /* Column names */
d1,1 d1,2 ... d1,ncol               /* Sample pts and values */
d2,1 d2,2 ... d2,ncol               /* Sample pts and values */
.
.

dnrow,1 dnrow,2 ... dnrow,ncol
```

Figure 8.4 GMS *Sample Data File Format.*

```
NAME "Hole #1"
XYZ 348.3 235.3 657.3
NUMCOLS 3
NUMROWS 6
"depth"      "tip"      "sleeve"
   0.0      37.600        0.262
   0.5      39.100        0.285
   1.0      37.600        0.262
   2.0      37.600        0.262
   2.5      37.600        0.262
   3.0      37.600        0.262
```

Figure 8.5 Sample GMS *Sample Data File.*

The card types used in GMS sample data files are as follows:

Card Type	**NAME**		
Description	Provides a name to be associated with the borehole.		
Required	NO		
Format	`NAME "name"`		
Sample	`BNAM "hole #1"`		
Field	*Variable*	*Value*	*Description*
1	name	str	The name of the borehole.

Card Type	**XYZ**		
Description	The xyz location of the top of the borehole.		
Required	YES		
Format	`XYZ x y z`		
Sample	`XYZ 0.0 5.0 100.0`		
Field	*Variable*	*Value*	*Description*
1	x	±	The x coordinate
2	y	±	The y coordinate
3	z	±	The z coordinate.

Card Type	**NUMCOLS**		
Description	The number of columns of data (including the depth column).		
Required	YES		
Format	`NUMCOLS ncol`		
Sample	`NUMCOLS 10`		
Field	*Variable*	*Value*	*Description*
1	ncol	+	Number of columns of data

Card Type	**NUMROWS**		
Description	The number of rows of data. The data set names and values follow this card.		
Required	YES		
Format	`NUMROWS nrow`		
Sample	`NUMROWS 100`		
Field	Variable	Value	Description
1	nrow	+	Number of rows of data

9 Range Files

Range files are used when doing a 1D soil classification in the *Sample Data* -> *Stratigraphy* dialog in the borehole module. A range file can consist of several range sets. Each range set can contain any number of values that define material ranges. The range file format is illustrated in Figure 9.1 and a sample range file is shown in Figure 9.2.

```
RANGE    /* File type identifier */
BEGINSET /* Beginning of a range set */
NAME "name"      /* Name of range set */
NUMVALS num      /* Number of values in this range set */
val1 matid       /* Value and material id */
val2 matid
.
.
valnum matid
ENDSET   /* End of range set */
/* Repeat for other range sets */
```

Figure 9.1 Range File Format.

```
RANGE
BEGINSET
NAME "Range Set 1"
NUMVALS 3
1.574257e+000 1
2.702970e+000 2
5.614000e+000 3
ENDSET
```

Figure 9.2 Sample Range File.

The card types used in range files are as follows:

Card Type	**RANGE**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**BEGINSET**
Description	Marks the beginning of a group of cards describing a range set. There should be a corresponding ENDSET card at a latter point in the file. No fields.

Required	YES

Card Type	**NAME**		
Description	Name of range set.		
Required	YES		
Format	`NAME "name"`		
Sample	`BNAM "set #1"`		
Field	*Variable*	*Value*	*Description*
1	name	str	The name of the range set.

Card Type	**NUMVALS**		
Description	The number of values defining the range set and the list of ranges.		
Required	YES		
Format	NUMVALS num val1 matid val2 matid . . val$_{num}$ matid		
Sample	NUMVALS 3 45 1 110 2 190 3		
Field	*Variable*	*Value*	*Description*
1	num	+	Number of values
2	val	±	Value
3	matid	+	Material associated with this value
Repeat 2-3 num times			

Card Type	**ENDSET**
Description	Marks the end of a group of cards describing a range set. There should be a corresponding BEGINSET card at a previous point in the file. No fields.
Required	YES

10 Solid Files

Solid files are used to store solid models. Solid models typically represent stratigraphic units. Multiple solids can be saved to a single file. The solids are boundary representation solids; each solid is defined by a set of polygonal boundaries. Each solid is composed of a set of triangles. The edges of the triangles corresponding to the polygon boundary are visible. The remaining edges are invisible. The solid file format is illustrated in Figure 10.1. A sample solid file is shown in Figure 10.2.

```
SOLID     /* File type identifier */
BEGS      /* Beginning of solid group */
HIDDEN flag      /* Visible status */
ID id     /* The id of the solid */
MAT id    /* Material id */
SNAM "name"      /* Name of solid */
VERT nv   /* Beg. of triangle vertices */
x0 y0 z0 /* Vertex coords */
x1 y1 z1
.
.
.
xnv-1 ynv-1 znv-1
TRI nt    /* Beg. of triangles */
v01 v02 v03 vis0        /* Tri. vert.s and visibility flag*/
v11 v12 v13 vis1
.
.
.
vnt-1 1 vnt-1 2 vnt-1 3 visnt-1
ENDS      /* End of solid group */
/* Repeat solid group for other solids. */
```

Figure 10.1 Solid File Format.

```
SOLID
BEGS
HIDDEN 0
ID 21
MAT 4
SNAM "BedRock"
VERT 67
357.0680200   485.60208000   1120.63489000
310.9947500   373.29840000   1156.72400000
.
.
1200.0000000   750.00000000   1115.00000000
1100.0000000   900.00000000   1115.00000000
TRI 130
  42   0    23   0
  23   0    38   0
.
.
  21   20   47   1
  47   48   21   2
ENDS
```

Figure 10.2 Sample Solid File.

The card types used in the solid file format are as follows:

Card Type	**SOLID**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**BEGS**
Description	Marks the beginning of a group of cards describing a solid. There should be a corresponding ENDS card at a latter point in the file. No fields.
Required	YES

Card Type	**HIDDEN**
Description	Indicates the visibility status of the solid
Required	NO

Format	`HIDDEN flag`		
Sample	`HIDDEN 0`		
Field	*Variable*	*Value*	*Description*
1	flag	 0 1	Visibility Status Visible Hidden

Card Type	**ID**		
Description	Indicates the id of the solid		
Required	YES		
Format	`ID id`		
Sample	`ID 12`		
Field	*Variable*	*Value*	*Description*
1	id	+	The id of the solid.

Card Type	**SNAM**		
Description	Provides a name to be associated with the solid.		
Required	NO		
Format	`SNAM "name"`		
Sample	`SNAM "Shale"`		
Field	*Variable*	Value	Description
1	name	str	The name of the solid.

Card Type	**MAT**		
Description	Associates a material id with the solid.		
Required	NO		
Format	`MAT id`		
Sample	`MAT 3`		
Field	*Variable*	*Value*	*Description*
1	id	+	The material id.

Card Type	**VERT**		
Description	Lists the vertices in the solid		
Required	YES		
Format	VERT nv x_0 y_0 z_0 x_1 y_1 z_1 . . . x_{nv-1} y_{nv-1} z_{nv-1}		
Sample	VERT 4 0.0 3.1 7.8 5.3 8.7 4.0 2.4 4.4 9.0 3.9 1.2 3.6		
Field	*Variable*	*Value*	*Description*
1	nv	+	The number of vertices in the solid
2-4	x,y,z	±	Coords. Of vertex. Repeat nv times.

Card Type	**TRI**
Description	Lists the triangles in the solid.
Required	YES

Format	TRI nt v_{01} v_{02} v_{03} vis_0 v_{11} v_{12} v_{13} vis_1 . . . $v_{nt-1\ 1}$ $v_{nt-1\ 2}$ $v_{nt-1\ 3}$ vis_{nt-1}		
Sample	TRI 4 5 1 4 4 1 2 4 2 3 5 4 3		
Field	*Variable*	*Value*	*Description*
1	nt	+	The number of triangles in the Solid.
2-4	v_1,v_2,v_3	+	Vertices of triangle listed in a counter-clockwise order.
5	vis	0-3	Visibility code for triangle 0: All edges are visible 1: Last edge is invisible 2: Last 2 edges are invisible 3: All edges are invisible
Repeat 2-5 nt times.			

Card Type	**ENDS**
Description	Marks the end of a group of cards describing a solid. There should be a corresponding BEGS card at a previous point in the file. No fields.
Required	YES

11 Cross Section Files

Cross section files are used to store cross sections that have been constructed from solids, 3D meshes, or 3D grids. Multiple cross sections may be saved to a single file. Cross sections are stored using lists of triangles that are similar to the list used in solid files. A cross section may pass through regions of a mesh, solid, or grid with differing material ids. Each portion of a cross section corresponding to a unique material id is stored in an individual triangle list.

Since cross sections can be constructed from a solid, mesh, or grid, the type of cross section must be saved in the cross section file. This ensures that when a cross section is read back into memory, data sets can be interpolated from the mesh or grid so that contours, vectors, and color fringes can be displayed on the cross section.

The format of the cross section file is shown in Figure 11.1. The cards shown in Figure 11.1 correspond to a cross section constructed from a set of solids. A sample cross section file is shown in Figure 11.2.

```
XSECT     /* File type identifier */
BEGX      /* Beginning of cross-sect. group */
SOLID     /* Type of cross-sect. */
BEGL      /* Beginning of layer group */
MAT id    /* Material id for layer*/
VERT nv   /* Beg. of triangle vertices */
x0 y0 z0 /* Vertex coords */
x1 y1 z1
.
.
.
xnv-1 ynv-1 znv-1
TRI nt    /* Beg. of triangles */
v01 v02 v03 vis0        /* Tri. vert.s and visibility flag*/
v11 v12 v13 vis1
.
.
.
vnt-1 1 vnt-1 2 vnt-1 3 visnt-1
ENDL      /* End of layer group */
/* Repeat layer group for other layers. */
ENDX      /* End of cross section group */
/* Repeat cross section group for other cross sections. */
```

Figure 11.1 Cross Section File Format.

```
XSECT
BEGX
SOLID
BEGL
MAT 1
VERT 24
2.7448339274795171e+02  2.8383306431588525e+02  1.1868026884250573e+03
3.2461352630716533e+02  2.3305469627099308e+02  1.1897183192849493e+03
.
.
4.0898516883781082e+02  1.4759204112141688e+02  1.1990448984018385e+03
4.3375415157219481e+02  1.2250276981719169e+02  1.2011064660148900e+03
TRI 22
0    1    2    2
3    0    2    2
.
.
22   23   19   1
16   22   18   2
ENDL
ENDX
```

Figure 11.2 Sample Cross Section File.

The card types used in the cross section file format are as follows:

Card Type	**XSECT**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**BEGX**
Description	Marks the beginning of a group of cards describing a cross section. There should be a corresponding ENDX card at a latter point in the file. No fields.
Required	YES

Card Type	**SOLID**
Description	Indicates that the cross section was constructed from a solid. No fields.
Required	YES (if the cross section was constructed from a solid)

Card Type	**MESH**		
Description	Indicates that the cross section was constructed from a 3D mesh.		
Required	YES (if the cross section was constructed from a 3D mesh)		
Format	`MESH nn ne`		
Sample	`MESH 1200 1350`		
Field	*Variable*	*Value*	*Description*
1	`nn`	+	The number of nodes in the 3D mesh.
2	`ne`	+	The number of elements in the 3D mesh. The numbers nn and ne are used to ensure that the current 3D mesh matches the mesh used to construct the cross section.

Card Type	**GRID**		
Description	Indicates that the cross section was constructed from a 3D grid.		
Required	YES (if the cross section was constructed from a 3D grid)		
Format	`Grid cellsi cellsj cellsk`		
Sample	`Grid 200 200 250`		
Field	*Variable*	*Value*	*Description*
1	`cellsi`	+	The number of cells in the i direction in the 3D grid.
2	`cellsj`	+	The number of cells in the j direction in the 3D grid.
3	`cellsk`	+	The number of cells in the k direction in the 3D grid. Fields 1-3 are used to ensure that the current 3D grid matches the grid used to construct the cross section. The numbers represent cells even if the grid is mesh-centered.

Card Type	**BEGL**
Description	Marks the beginning of a group of cards describing a layer. Layers are grouped according to material id. There should be a corresponding ENDL card at a latter point in the file. No fields.
Required	YES

Card Type	**MAT**		
Description	Associates a material id with the layer.		
Required	NO		
Format	`MAT id`		
Sample	`MAT 3`		
Field	*Variable*	*Value*	*Description*
1	`id`	+	The material id.

Card Type	**VERT**
Description	Lists the vertices in the layer.
Required	YES

Format	VERT nv x_0 y_0 z_0 id x_1 y_1 z_1 id . . . x_{nv-1} y_{nv-1} z_{nv-1} id		
Sample	VERT 4 0.0 3.1 7.8 5.3 8.7 4.0 2.4 4.4 9.0 3.9 1.2 3.6		
Field	*Variable*	*Value*	*Description*
1	nv	+	The number of vertices in the layer.
2-4	x,y,z	±	Coords. of vertex. Repeat nv times.
5	id	+	This field is not used for cross sections constructed from solids. If the cross section was constructed from a 3D mesh, it represents the id of one of the elements adjacent to the edge where the vertex is located. If the cross section was constructed from a 3D grid, it represents the id of one of the cells adjacent to the edge where the vertex is located.
Fields 2-5 are repeated nv times.			

Card Type	**TRI**
Description	Lists the triangles in the layer.
Required	YES
Format	(see the TRI card described in Section 10)

Card Type	**ENDL**
Description	Marks the end of a group of cards describing a layer. There should be a corresponding BEGL card at a previous point in the file. No fields.
Required	YES

Card Type	**ENDX**
Description	Marks the end of a group of cards describing a cross section. There should be a corresponding BEGX card at a previous point in the file. No fields.
Required	YES

12 2D Mesh Files

Two-dimensional finite element meshes are stored in 2D mesh files. The types of elements supported in GMS and the numbering sequence for the elements are shown in Figure 12.1. The file format is shown in Figure 12.2 and a sample 2D mesh file is shown in Figure 12.3.

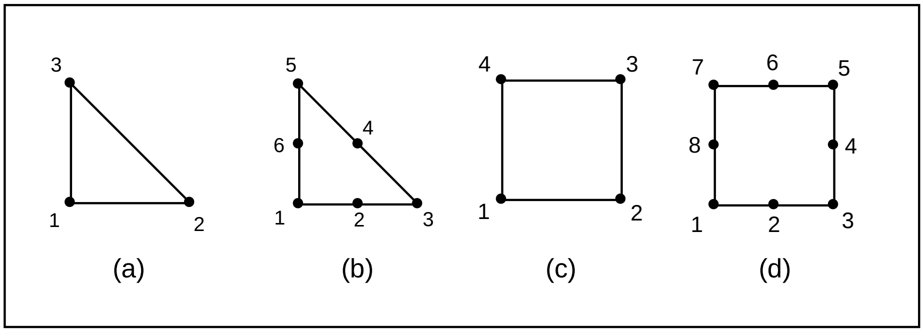

Figure 12.1 The Four Basic 2D Elements. (a) Linear Triangles (b) Quadratic Triangles (c) Linear Quadrilaterals (d) Quadratic Quadrilaterals.

```
MESH2D   /* File type identifier */
E3T id n1 n2 n3 mat                 /* 3 node triangle  */
E6T id n1 n2 n3 n4 n5 n6 mat        /* 6 node triangle  */
E4Q id n1 n2 n3 n4 mat              /* 4 node quad  */
E8Q id n1 n2 n3 n4 n5 n6 n7 n8 mat  /* 8 node quad  */
ND id x y z      /* Nodal coordinates */
```

Figure 12.2 2D Mesh File Format.

```
MESH2D
E4Q     1     4     5     2     1     0
E4Q     2     5     6     3     2     0
E3T     3     7     5     4     0
E3T     4     7     8     5     0
E3T     5     8     6     5     0
E3T     6     8     9     6     0
ND     1 2.00000000e+001 1.00000000e+002 0.00000000e+000
ND     2 2.00000000e+001 9.00000000e+001 0.00000000e+000
ND     3 2.00000000e+001 8.00000000e+001 0.00000000e+000
ND     4 1.00000000e+001 1.00000000e+002 0.00000000e+000
ND     5 1.00000000e+001 9.00000000e+001 0.00000000e+000
ND     6 1.00000000e+001 8.00000000e+001 0.00000000e+000
ND     7 0.00000000e+000 1.00000000e+002 0.00000000e+000
ND     8 0.00000000e+000 9.00000000e+001 0.00000000e+000
ND     9 0.00000000e+000 8.00000000e+001 0.00000000e+000
```

Figure 12.3 Sample 2D Mesh File.

The card types used in the 2D mesh file are as follows:

Card Type	**MESH2D**		
Description	File type identifier. Must be on first line of file. No fields.		
Required	YES		
Card Type	**E3T**		
Description	Defines a three node (linear) triangular element.		
Required	NO		
Format	`E3T id n1 n2 n3 mat`		
Sample	`E3T 283 13 32 27 4`		
Field	*Variable*	*Value*	*Description*
1	id	+	The id of the element.
2-4	n1-n3	+	The nodal indices of the element ordered counterclockwise.
5	mat	+	The material id for the element.

Card Type	**E6T**		
Description	Defines a six node (quadratic) triangular element.		
Required	NO		
Format	`E6T id n1 n2 n3 n4 n5 n6 mat`		
Sample	`E6T 283 13 32 27 22 25 30 4`		
Field	*Variable*	*Value*	*Description*
1	id	+	The id of the element.
2-7	n1-n6	+	The nodal indices of the element ordered counterclockwise starting at a corner node.
8	mat	+	The material id for the element.

Card Type	**E4Q**		
Description	Defines a four node (linear) quadrilateral element.		
Required	NO		
Format	`E4Q id n1 n2 n3 n4 mat`		
Sample	`E4Q 283 13 32 27 30 4`		
Field	*Variable*	*Value*	*Description*
1	id	+	The id of the element.
2-5	n1-n4	+	The nodal indices of the element ordered counterclockwise.
6	mat	+	The material id for the element.

Card Type	**E8Q**		
Description	Defines an eight node (quadratic) quadrilateral element.		
Required	NO		
Format	`E8T id n1 n2 n3 n4 n5 n6 n7 n8 mat`		
Sample	`E8T 283 13 32 27 22 25 30 29 31 4`		
Field	*Variable*	*Value*	*Description*
1	id	+	The id of the element.
2-9	n1-n8	+	The nodal indices of the element ordered counterclockwise starting at a corner node.
10	mat	+	The material id for the element.

Card Type	**ND**
Description	Defines the coordinates of a node.
Required	NO
Format	`ND id x y z`
Sample	`ND 84 120.4 380.3 5632.0`

Field	*Variable*	*Value*	*Description*
1	id	+	The id of the node.
2-4	x,y,z	±	The nodal coordinates.

13 3D Mesh Files

Three-dimensional finite element meshes are stored in 3D mesh files. The types of elements supported in GMS and the element numbering are shown in Figure 13.1. The file format is shown in Figure 13.2 and a sample 3D mesh file is shown in Figure 13.3.

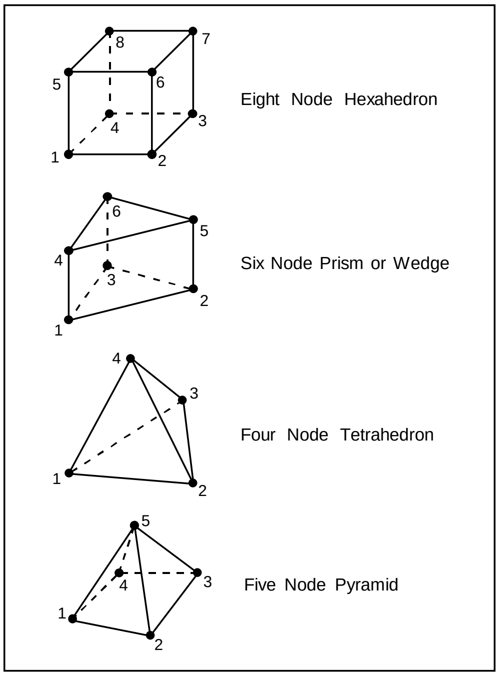

Figure 13.1 Node Ordering for Three-Dimensional Elements Supported by GMS*.*

```
MESH3D   /* File type identifier */
E8H id n1 n2 n3 n4 n5 n6 n7 n8 mat  /* 8 node hexahedron  */
E6W id n1 n2 n3 n4 n5 n6 mat        /* 6 node wedge  */
E4T id n1 n2 n3 n4 mat              /* 4 node tetrahedron */
E5P id n1 n2 n3 n4 n5 mat           /* 5 node pyramid  */
ND id x y z      /* Nodal coordinates */
```

Figure 13.2 3D Mesh File Format.

```
MESH3D
E8H 1 14 11 10 13 5 2 1 4 0
.
.
E8H 8 27 24 23 26 18 15 14 17 0
ND 1 -3.711554e+00 1.046370e+02 4.600000e+01
.
.
ND 27 1.044770e+02 1.371743e+01 6.000000e+00
```

Figure 13.3 Sample 3D Mesh File.

The card types used in the 3D mesh file are as follows:

Card Type	**MESH3D**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**E8H**		
Description	Defines an eight node hexahedral element.		
Required	NO		
Format	`E8H id n1 n2 n3 n4 n5 n6 n7 n8 mat`		
Sample	`E8H 283 13 32 27 22 25 30 29 31 4`		
Field	*Variable*	*Value*	*Description*
1	`id`	+	The id of the element.
2-9	`n1-n8`	+	The nodal indices of the element ordered as shown in Figure 13.1.
10	`mat`	+	The material id for the element.

Card Type	**E6W**		
Description	Defines a six node wedge element.		
Required	NO		
Format	`E6W id n1 n2 n3 n4 n5 n6 mat`		
Sample	`E6W 283 13 32 27 22 25 30 4`		
Field	*Variable*	*Value*	*Description*
1	`id`	+	The id of the element.
2-7	`n1-n6`	+	The nodal indices of the element ordered as shown in Figure 13.1.
8	`mat`	+	The material id for the element.

Card Type	**E4T**		
Description	Defines a four node tetrahedral element.		
Required	NO		
Format	`E4T id n1 n2 n3 n4 mat`		
Sample	`E4T 283 13 32 27 30 4`		
Field	*Variable*	*Value*	*Description*
1	`id`	+	The id of the element.
2-5	`n1-n4`	+	The nodal indices of the element ordered as shown in Figure 13.1.
6	`mat`	+	The material id for the element.

Card Type	**E5P**		
Description	Defines a five node pyramid element.		
Required	NO		
Format	`E5P id n1 n2 n3 n4 n5 mat`		
Sample	`E5P 283 13 32 27 30 33 4`		
Field	*Variable*	*Value*	*Description*
1	`id`	+	The id of the element.
2-6	`n1-n5`	+	The nodal indices of the element ordered as shown in Figure 13.1.
7	`mat`	+	The material id for the element.

Card Type	**ND**		
Description	Defines the coordinates of a node.		
Required	NO		
Format	`ND id x y z`		
Sample	`ND 84 120.4 380.3 5632.0`		
Field	*Variable*	*Value*	*Description*
1	id	+	The id of the node.
2-4	x,y,z	±	The nodal coordinates.

14 2D Grid Files

The 2D grid file format is shown in Figure 14.1. A sample 2D grid file is shown in Figure 14.2. Two-dimensional grids can be either cell-centered or mesh-centered. If the grid is mesh-centered, a set of material ids may be included in the file. If some of the cells are inactive, active/inactive flags are stored in the file.

```
GRID2D    /* File type */
TYPE i    /* Type of grid. Mesh or Cell centered. */
IJ ±idir ±jdir /* Card for defining rows, columns. */
ORIGIN x y z   /* Coordinates of the grid origin*/
ROTZ θ    /* Rotation angle of the grid */
DIM nx ny /* # of cell boundaries in each direction. */
x1        /* X coord. of cell boundaries. */
x2
.
.
xnx
y1        /* Y coord. of cell boundaries. */
y2
.
.
yny
DMAT id   /* The default material type. */
MAT       /* Explicit definition of all material ids.*/
id1 id2 . . idn
ACTIVE    /* Activity of the grid */
stat1 stat2 . . statn
```

Figure 14.1 2D Grid File Format.

```
GRID2D
TYPE 1
IJ -y +x
ORIGIN 1.500000000e+00 3.600000000e+00 3.800000000e+00
ROTZ 10.2
DIM 4 4
0.000000000e+00
3.333333334e+01
.
.
6.666666667e+02
1.000000000e+02
MAT
1 1 1 6 6 1 2 2 1
ACTIVE
1 1 . . 1 1
```

Figure 14.2 Sample 2D Grid File.

The card types used in the 2D grid file format are as follows:

Card Type	**GRID2D**		
Description	File type identifier. Must be on first line of file. No fields.		
Required	YES		

Card Type	**TYPE**		
Description	Defines the type of grid as either cell- or mesh-centered.		
Required	YES		
Format	`TYPE I`		
Sample	`TYPE 0`		
Field	*Variable*	*Value*	*Description*
`1`	`I`	0,1	The type code: I = 0 for mesh-centered I = 1 for cell-centered

Card Type	**IJ**		
Description	Defines the orientation of the i,j indices.		
Required	YES		
Format	`IJ ±idir ±jdir`		
Sample	`IJ +x -y`		
Field	*Variable*	*Value*	*Description*
`1`	`±idir`	±x,±y	The direction corresponding to an increasing i index.
`2`	`±jdir`	±x,±y	The direction corresponding to an increasing j index.

Card Type	**ORIGIN**		
Description	Defines both the coordinates of the grid origin and its default elevation.		
Required	NO		
Format	`ORIGIN x y z`		
Sample	`ORIGIN 1.5000000000e+00 3.6000000000e+00 3.8000000000e+00`		
Field	*Variable*	*Value*	*Description*
`1 to 3`	`x y z`	±	The xy coordinates define the minx, miny corner of the unrotated grid. The z coordinate is the default elevation of the grid.

Card Type	**ROTZ**		
Description	Defines a rotation angle (degrees) to which the grid will be rotated counter-clockwise around the z axis.		
Required	NO		
Format	`ROTZ θ`		
Sample	`ELEV 10.2`		
Field	*Variable*	*Value*	*Description*
`1`	θ	±	Rotation angle (degrees counter-clockwise) around the z axis to which the grid will be rotated.

Card Type	**DIM**		
Description	Defines the dimensions of the grid.		
Required	YES		
Format	DIM nx ny x_1 x_2 . . x_{nx} y_1 y_2 . . y_{ny}		
Sample	DIM 4 6 0.0 1.0 2.0 4.0 10.0 12.0 14.0 16.0 18.0 20.0		
Field	*Variable*	*Value*	*Description*
1	nx	+	The number of cell boundaries in the x direction.
2	ny	+	The number of cell boundaries in the y direction.
3 to (nx+2)	x_1-x_{nx}	±	The coordinates of the x boundaries. The values are relative values. They should always start at zero.
(nx+3) to (nx+ny+2)	y_1-y_{ny}	±	The coordinates of the y boundaries. The values are relative values. They should always start at zero.

Card Type	**DMAT**		
Description	Defines a default material id for the grid if all cells in the grid are of the same material.		
Required	NO		
Format	DMAT id		
Sample	DMAT 1		
Field	*Variable*	*Value*	*Description*
1	id	+	The default material id.

Card Type	**MAT**		
Description	Defines a set of material ids for each cell in the grid. Use in place of the DMAT card when not all cells have the same material id.		
Required	NO		
Format	MAT id_1 id_2 . . id_n		
Sample	MAT 1 1 1 2 2 2		
Field	*Variable*	*Value*	*Description*
1-n	id	+	The material ids are listed row by row (i direction first). One id is listed for each cell. Can span multiple lines if necessary.

Card Type	ACTIVE		
Description	Indicates that a set of status flags exist for the cells of the grid.		
Required	NO		
Format	ACTIVE $stat_1$ $stat_2$ . . $stat_n$		
Sample	ACTIVE 1 0 . . 1		
Field	*Variable*	*Value*	*Description*
1-n	stat		Status of each cell. Can span multiple lines
		0	Inactive
		1	Active

15 3D Grid Files

The 3D grid file format is shown in Figure 15.1. A sample 3D grid file is shown in Figure 15.2. The grids can be either cell-centered or mesh-centered. A set of material ids may be included in the file, one id for each grid cell.

```
GRID3D    /* File type */
TYPE i    /* Type of grid. Mesh or Cell centered. */
IJK ±idir ±jdir ±jdir    /* The ijk orientation. */
ORIGIN x y z    /* Coordinates of the grid origin*/
ROTZ θ    /* Rotation angle of the grid */
DIM nx ny nz    /* # of cell boundaries in each direction. */
x1        /* X coord. of cell boundaries. */
x2
.
.
xnx
y1        /* Y coord. of cell boundaries. */
y2
.
.
yny
z1        /* Z coord. of cell boundaries. */
z2
.
.
znz
DMAT id   /* The default material type. */
MAT       /* Explicit definition of all material ids.*/
id1 id2 . . idn
ACTIVE    /* Activity of the grid */
stat1 stat2 . . statn
```

Figure 15.1 3D Grid File Format.

```
GRID3D
TYPE 1
IJK -y +x -z
ORIGIN -3.91517473684211e+02 3.89803789473684e+02 -9.71000000000000e+02
ROTZ 0
DIM 74 46 2
0.000000000000000e+00
2.679734304524754e+02
.
.
0.000000000000000e+00
1.943000000000000e+03
DMAT 0
ACTIVE
1 0 . . 0 1
```

Figure 15.2 Sample 3D Grid File.

The card types used in the 3D grid file format are as follows:

Card Type	**GRID3D**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**TYPE**		
Description	Defines the type of grid as either cell- or mesh-centered.		
Required	YES		
Format	`TYPE I`		
Sample	`TYPE 0`		
Field	*Variable*	*Value*	*Description*
1	I	 0 1	The type code: mesh-centered cell-centered

Card Type	**IJK**		
Description	Defines the orientation of the ijk indices.		
Required	YES		
Format	`IJK ±idir ±jdir ±kdir`		
Sample	`IJK +x -y -z`		
Field	*Variable*	*Value*	*Description*
1	±idir	±x,±y, ±z	The direction corresponding to an increasing i index.
2	±jdir	±x,±y, ±z	The direction corresponding to an increasing j index.
3	±kdir	±x,±y, ±z	The direction corresponding to an increasing k index.

Card Type	**ORIGIN**		
Description	Defines x y z the coordinates of the grid origin.		
Required	NO		
Format	`ORIGIN x y z`		
Sample	`ORIGIN 1.5000000000e+00 3.6000000000e+00 3.8000000000e+00`		
Field	*Variable*	*Value*	*Description*
1 to 3	x y z	±	Defines the x y z coordinates of the grid origin (the minx, miny, minz corner of the unrotated grid).

Card Type	**ROTZ**		
Description	Defines a rotation angle (degrees) to which the grid will be rotated counter-clockwise around the z axis.		
Required	NO		
Format	`ROTZ θ`		
Sample	`ELEV 10.2`		
Field	*Variable*	*Value*	*Description*
1	θ	±	Rotation angle (degrees counter-clockwise) around the z axis to which the grid will be rotated.

Card Type	**DIM**
Description	Defines the dimensions of the grid.

Required	YES		
Format	DIM nx ny nz x_1 x_2 . . x_{nx} y_1 y_2 . . y_{ny} z_1 z_2 . . z_{nz}		
Sample	DIM 4 6 2 0.0 1.0 2.0 4.0 10.0 12.0 14.0 16.0 18.0 20.0 0.0 1.0		
Field	*Variable*	*Value*	*Description*
1	nx	+	The number of cell boundaries in the x direction.
2	ny	+	The number of cell boundaries in the y direction.
3	nz	+	The number of cell boundaries in the z direction.
4 to (nx+3)	x_1-x_{nx}	±	The coordinates of the x boundaries. These are relative values and should begin at zero.
(nx+4) to (nx+ny+3)	y_1-y_{ny}	±	The coordinates of the y boundaries. These are relative values and should begin at zero.
(nx+ny+4) to (nx+ny+nz+3)	z_1-z_{nz}	±	The coordinates of the z boundaries. These are relative values and should begin at zero.

Card Type	**DMAT**		
Description	Defines a default material id for the grid if all cells in the grid are of the same material.		
Required	NO		
Format	DMAT id		
Sample	DMAT 1		
Field	*Variable*	*Value*	*Description*
1	id	+	The default material id.

Card Type	**MAT**		
Description	Defines a set of material ids for each cell in the grid.		
Required	NO		
Format	MAT id_1 id_2 . . id_n		
Sample	MAT 1 1 1 2 2 2		
Field	*Variable*	*Value*	*Description*

1-n	id	+	The material ids are listed by row, then column, then layer. One id is listed for each cell. Can span mulitple lines if necessary.

Card Type	**ACTIVE**		
Description	Indicates that a set of status flags exist for the cells of the grid.		
Required	NO		
Format	ACTIVE $stat_1$ $stat_2$ . . $stat_n$		
Sample	ACTIVE 1 0 . . 1		
Field	*Variable*	*Value*	*Description*
1-n	stat	 0 1	Status of each cell. Can span multiple lines. Inactive Active

16 2D Scatter Point Files

The format of the 2D scatter file is shown in Figure 16.1. A sample 2D scatter point file is shown in Figure 16.2. Multiple scatter point sets can be stored in a single file. Each point in a scatter point set is defined by a pair of xy coordinates.

```
SCAT2D    /* File type identifier */
BEGSET    /* Beginning of cards for scatter point set */
ACTIVESET /* Marks the active scatter point set */
NAME "name"      /* Name of scatter point set */
ID id     /* ID of scatter point set */
DELEV elev1      /* Default elevation */
IXY np    /* Number of points in set, begin point listing */
id1 x1 y1 mat1/* Point id, coordinatess, and material one per line */
id2 x2 y2 mat2
.
.
idnp xnp ynp matnp
ENDSET    /* End of cards for scatter point set */
/* Repeat point set cards as many times as necessary */
```

Figure 16.1 2D Scatter Point File Format.

```
SCAT2D
BEGSET
ACTIVESET
NAME   "wells"
ID   8493
DELEV 0.00000000000e+00
IXY 25
"1" 1.47000000000e+02 3.90000000000e+02 1
"2" 8.82000000000e+02 9.49000000000e+02 3
.
.
"24" 1.73000000000e+02 7.01000000000e+02 2
"25" 5.39000000000e+02 8.98000000000e+02 2
ENDSET
```

Figure 16.2 Sample 2D Scatter Point File.

The cards used in the 2D scatter point file are as follows:

Card Type	**SCAT2D**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**BEGSET**
Description	Identifies the beginning of a scatter point set. No fields.
Required	NO

Card Type	**ACTIVESET**
Description	Marks the active scatter point set.
Required	NO

Card Type	**NAME**		
Description	Defines the name for the following scatter point set.		
Required	NO		
Format	`NAME "name"`		
Sample	`NAME "wells"`		
Field	*Variable*	*Value*	*Description*
1	name	str	The name for the following scatter points. Remains as default until new NAME card is encountered.

Card Type	**ID**		
Description	Defines the ID for the scatter point set.		
Required	YES		
Format	`ID id`		
Sample	`ID 43098`		
Field	*Variable*	*Value*	*Description*
1	id	+	The ID for the following scatter point set.

Card Type	**DELEV**		
Description	Defines the default elevation for the scatter point set.		
Required	NO		
Format	`DELEV el`		
Sample	`DELEV 9.0`		
Field	*Variable*	*Value*	*Description*
1	el	±	The default elevation for the following scatter points. Remains as default until new DELEV card is encountered.

Card Type	**IXY**
Description	Defines a scatter point set.
Required	YES
Format	IXY np id_1 x_1 y_1 mat_1 id_2 x_2 y_2 mat_2 . . id_{np} x_{np} y_{np} mat_{np}

Sample	IXY 4 "1" 12.3 34.5 1 "2" 52.2 23.5 2 "3" 63.2 27.4 2 "4" 91.1 29.3 1		
Field	*Variable*	*Value*	*Description*
1	np	+	The number of scatter points in the scatter point set.
2	id	+	The ids of the points. Can be a number or a text string.
3-4	x,y	±	The coordinates of the points.
5	mat	+	The material id of the scatter point. Used for indicator krigin.
Repeat fields 2-4 np times			

Card Type	**ENDSET**
Description	Identifies the end of a scatter point set. No fields.
Required	NO

17 3D Scatter Point Files

Three-dimensional scatter point sets are stored in 3D scatter point files. Multiple scatter point sets can be stored in a single file. Each point in a scatter point set is defined by a set of XYZ coordinates. The format of the 3D scatter file is shown in Figure 17.1. A sample 3D scatter point file is shown in Figure 17.2.

```
SCAT3D    /* File type identifier */
BEGSET    /* Beginning of cards for scatter point set */
ACTIVESET /* Marks the active scatter point set */
NAME "name"      /* Name of the scatter point set */
ID id     /* ID of the scatter point set */
IXYZ np   /* Number of points in set, begin point listing */
id1 x1 y1 z1 mat1       /* Point id, coord.'s, and mat, one per line */
id2 x2 y2 z2 mat2
.
.
idnp xnp ynp znp matnp
ENDSET
```

Figure 17.1 3D Scatter Point File Format.

```
SCAT3D
BEGSET
ACTIVESET
NAME  "tank"
ID   25151
IXYZ 230
"1" 4.50000000000e+01 4.4000000000e+01 -5.0000000000e+00 2
"2" 4.50000000000e+01 4.4000000000e+01 -1.0000000000e+01 2
.
.
"229" 2.6700000000e+02 1.9000000000e+01 -4.5000000000e+01 2
"230" 2.6700000000e+02 1.9000000000e+01 -5.0000000000e+01 1
ENDSET
```

Figure 17.2 Sample 3D Scatter Point File.

The cards used in the 3D scatter point file are as follows:

Card Type	**SCAT3D**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**BEGSET**
Description	Identifies the beginning of a scatter point set. No fields.
Required	NO

Card Type	**ACTIVESET**
Description	Marks the active scatter point set.
Required	NO

Card Type	**NAME**		
Description	Defines the name for the following scatter point set.		
Required	NO		
Format	`NAME "name"`		
Sample	`NAME "wells"`		
Field	*Variable*	*Value*	*Description*
1	name	str	The name for the following scatter points. Remains as default until new NAME card is encountered.

Card Type	**ID**		
Description	Defines the ID for th e following scatter point set.		
Required	YES		
Format	`ID id`		
Sample	`ID 42501`		
Field	*Variable*	*Value*	*Description*
1	id	+	The ID for the following scatter point set.

Card Type	**IXYZ**		
Description	Defines a scatter point set.		
Required	YES		
Format	IXYZ np id_1 x_1 y_1 z_1 mat_1 id_2 x_2 y_2 z_2 mat_2 . . id_{np} x_{np} y_{np} z_{np} mat_{np}		
Sample	IXYZ 4 "1" 12.3 34.5 10.5 3 "2" 52.2 23.5 32.2 3 "3" 63.2 27.4 83.7 3 "4" 91.1 29.3 52.1 3		
Field	*Variable*	*Value*	*Description*
1	np	+	The number of scatter points in the scatter point set.
2	id	+	The id of the point. Can be a number or a text string.

3-5	x,y,z	±	The point coordinates.
6	mat	+	The material id of the point. Used for indicator simulations.
Repeat fields 3-5 np times			

Card Type	**ENDSET**
Description	Identifies the end of a scatter point set. No fields.
Required	NO

18 Tabular Scatter Point Files

When scatter point sets are saved by GMS, the scatter point coordinates are saved in one file and the values at the scatter points are saved in a data set file. When importing scatter point data into GMS for the first time, this format may be used if desired. However, scatter point sets are typically created by importing from a text file using the *Import* command in the *File* menu. Two options are provided for importing scatter point sets that have been saved in a tabular format: *Multiple data sets* and *Transient data sets*. Both formats are referred to as "tabular scatter point files." The easiest way to prepare such a file is to enter the data in a spreadsheet and save the spreadsheet to a text (*.txt) file. The extension to the file name for both types of tabular scatter point files should always be set to *.sp2 for 2D files and *.sp3 for 3D files.

18.1 Multiple Data Sets

The multiple data sets option is used to import a scatter point set with one or more steady state data sets. The first column is an optional id column, the next two columns represent the x and y coordinates, and each of the remaining columns represents a data set. Each of the data sets can be named. If the id column is omitted, the points will be automatically numbered. The multiple data set file format is shown in Figure 18.1. A sample file is shown in Figure 18.2.

The formats shown in Figures Figure 18.1Figure 18.2 are for the 2D case. The 3D format is identical except that a column is added for the z coordinate.

```
NONDETECT value  /* Nondetect value */
"id" "x" "y" "name1" "name2" "name3" ...    /* Names of fields */
id x y val1 val2 val3 ...                   /* Scatter pts w/ values */
.
.
```

Figure 18.1 The Multiple Data Set Option for Tabular Scatter Point Input.

```
NONDETECT -999
"id"     "x"      "y"      "xylene"   "toluene"   "benzene"
"OW-21"  32.4     5234.3   300        -999        120
"OW-22"  93.4     5832.3   84         398         389
"OW-23"  83.3     8438.2   89         47          482
```

Figure 18.2 Sample Multiple Data Set Tabular Scatter Point File.

The NONDETECT card is used to define a key value to identify inactive values in the data sets. The NONDETECT card is optional and should be placed at the beginning of the file. The "x" and "y" headers are NOT optional.

An optional material column may be added to the right of the last data set to associate material ids with scatter points for use in indicator simulations. The column should contain integer values and the heading for the column should be spelled precisely as "material".

18.2 Transient Data Sets

With the transient data sets format, the first column is an optional id column, the next two columns represent the x and y coordinates, and the remaining columns represent time steps within a single data set. If the id column is omitted, the points will be automatically numbered. The transient data set file format is shown in Figure 18.3. A sample file is shown in Figure 18.4.

```
NONDETECT value  /* Nondetect value */
"data set name"  /* Name of data set */
       time1 time2 time3 ...              /* List of times */
id x y val1 val2 val3 ...                 /* Scatter pts and values */
.
.
```

Figure 18.3 The Transient Data Set Option for Tabular Scatter Point Input.

```
NONDETECT -999
"Ground water elev."
                            100.0      110.0      120.0
"OW-21"  32.4     5234.3    300        200        120
"OW-21"  93.4     5832.3    84         398        -999
"OW-21"  83.3     8438.2    89         47         482
```

Figure 18.4 Sample Transient Data Set Tabular Scatter Point File.

Once again, the NONDETECT card and the id column are optional. The "x" and "y" headers are NOT optional.

An optional material column may be added to the right of the last data set to associate material ids with scatter points for use in indicator simulations. The column should contain integer values and no heading is required for the column.

19 Tabular Observation Point Files

Observation points are used for model calibration. Observation points can be created one at a time using the *Create Point* tool and the point attributes can be entered using the *Attributes* dialog. However, for sites with large numbers of points, this type of entry can be time consuming. In such cases, it is more efficient to organize the observation point data in a spreadsheet, export the spreadsheet as a text file, and import the points using the *Import* command in the *File* menu. If an observation coverage already exists and is the active coverage, the points are added to the coverage. Otherwise, a new one is created. Once the file is imported, the *Observation Coverage Options* dialog appears and the general options for the coverage can be established.

Before importing the file to GMS, the file extension should be changed to **.tob*. When importing files, GMS determines the file type from the three character extension. The text file should be formatted as shown in Figure 19.1. A sample file is shown in Figure 19.2.

```
NODATA value
"name" "x" "y" "z" "layer" "mtype1" "int" "conf" "mtype2"...
"name" x y z id val1 int2 conf1 val2...
.
.
```

Figure 19.1 Observation Point Import File Format.

```
NODATA -999
"name" "x" "y" "z" "layer" "head" "int" "conf"
"OBS Q5" 234.3 44.2 323.2 1 567.5 1.2 95
"OBS Q6" 833.3 842.3 320.2 1 555.3 1.4 90
"OBS Q8" 855.3 898.3 322.2 1 -999 0 0
.
.
```

Figure 19.2 Sample Observation Point Import File.

The file can be delimited using spaces, tabs, or commas. The items in the text file are as follows:

19.1 NODATA Record

The NODATA record is an **optional** record used to list a key value which is used to flag values which where not observed. For example, if a NODATA value of -999 is listed and one of the observed values in a measurement type column has a value of -999, the value is assumed to be not observed at that point. This is equivalent to turning off the *Observed* toggle in the *Observation Point Attributes* dialog.

19.2 Name Column

The name column is an **optional** column used to list the name or id of each point. If the column is missing, GMS will assign a default name to each point. For example, the first point would be named "point_1" by default. If the column is present, it must begin with a column header of "name" (case insensitive). The quotes on the column header are optional. The quotes on the point names are also optional, but must be used if the name contains spaces.

19.3 X, Y, and Z Columns

The x, y, and z columns are **required** columns and should be used to list the coordinates of each point. The quotes on the column header are optional.

19.4 Layer Column

The layer column is an **optional** column which can be used for points associated with 3D grids. Each value in the column must be an integer. The layer ids are used to list the layer of the 3D grid each point is to be associated with. If this column is missing for a 3D grid, GMS uses the elevations of the points (z values) to determine the appropriate layer. If the column is present, it must begin with a column header of "layer" (case insensitive). The quotes on the column header are optional.

19.5 Observed Values Column

After the layer column (or the z column if the layer column does not exist), a column of observed values may be listed. The column header provides the name of the measurement type. The values should then be listed in the column. The NODATA value may be used to signify an unobserved value as explained above. Quotes must be used on the column header if it contains spaces but are optional otherwise.

19.6 Interval and Confidence or Standard Deviation Columns

After the observed values column, the calibration target should be specified using one or two columns. If the interval and confidence option is used, two columns should be listed. The column header for the first column must begin with the letters "Int" (case insensitive) and the column should contain the interval. The header for the second column must begin with the letters "con" and the column should contain the confidence values (0 - 100). If the standard deviation option is chosen, only one column should be listed. The column header must begin with the letters "St" (case insensitive) and the standard deviation values should be listed. In each case, the quotes on the column names are optional. The quotes should be used if the names contain spaces.

19.7 Multiple Measurement Types

If multiple measurement types exist, they should be listed in subsequent columns using the value-interval-confidence or value-std. deviation sequence described above.

20 Map Files

Map files are used to store feature object and drawing object data. Feature objects include points, nodes, vertices, arcs, and polygons. Drawing objects include rectangles, ovals, lines, and text. The map file also includes the grid frame. The map file includes not only the geometric description of such objects, but also the attributes associated with each object.

The number of different types of cards used in the map file is quite large. Thus, it is impractical to illustrate the file format in a figure as is done for most other formats described in this document. However, each of the cards in the file are fully described below. A sample map file is contained in the tutorial/overview directory of the files distributed with GMS.

The basic approach used to define an object in the map file is to list a card marking the beginning of a set of cards defining the object, then list the cards defining the geometry and attributes of the object, and then a card to mark the end of the set of cards defining the object.

The cards in the map file are as follows:

20.1 General

The following cards are used to define general map data.

Card Type	**MAP**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**GRIDFRAME**		
Description	Specifies the location and dimension of the gridframe		
Required	NO		
Format	`GRIDFRAME x y z dx dy dz θ`		
Sample	`GRIDFRAME -604.982 117112 -996 14075 7075 1992 0`		
Field	*Variable*	*Value*	*Description*
1-7	x, y, z dx, dy, dz θ	± ± ±	The coordinates of the gridframe origin. The dimensions of the gridframe. Rotation angle (degrees counter-clockwise) around the z axis to which the grid will be rotated.

20.2 Transport Models

For conceptual models associated with MT3DMS/RT3D/SEAM3D, the following cards are listed at the top of the map file.

Card Type	**TRANMOD**		
Description	Defines the transport model.		
Required	NO (if card is omitted, the default will be MT3DMS)		
Format	`TRANMOD option`		
Sample	`TRANMOD 1`		
Field	*Variable*	*Value*	*Description*
1	option	0,1,2	The transport model 0 = MT3DMS 1 = RT3D 2 = SEAM3D

Card Type	**REACTTYPE**		
Description	Defines the type of reaction.		
Required	YES (if the transport model = RT3D)		
Format	`REACTTYPE type`		
Sample	`REACTTPE 1`		
Field	*Variable*	*Value*	*Description*
1	type	1-9	The type of reaction

Card Type	**MT3DSPECIES**		
Description	Defines the number of MT3DMS/RT3D species.		
Required	NO		
Format	`MT3DSPECIES ncomp mcomp`		
Sample	`MT3DSPECIES 10 6`		
Field	*Variable*	*Value*	*Description*
1	ncomp	+	The total number of species
2	mcomp	+	The number of mobile species

Card Type	**SPC**		
Description	MT3DMS/RT3D/SEAM3D species card.		
Required	YES (if MT3D/RT3D/SEAM3D is being used)		
Format	`SPC "name" id mflag`		
Sample	`SPC "tce" 1 1`		
Field	*Variable*	*Value*	*Description*
1	name	str	The name of the specie
2	id	+	The id of the specie
3	mflag	0 1	The specie is immobile The specie is mobile

Card Type	**REACTTYPE**		
Description	Defines the type of reaction.		
Required	YES (if the transport model = RT3D)		
Format	`REACTTYPE type`		
Sample	`REACTTPE 1`		
Field	*Variable*	*Value*	*Description*
1	type	1-9	The type of reaction

Card Type	**NVRXNDATA**		
Description	Defines the number of spatially variable reaction parameters.		
Required	YES (if the transport model = RT3D and the user-defined reaction type is selected)		
Format	`NVRXNDATA num`		
Sample	`NVRXNDATA 3`		
Field	*Variable*	*Value*	*Description*
1	num	+	The number of spatially variable parameters

Card Type	**S3DSPEC**		
Description	Defines the main SEAM3D species.		
Required	YES, if SEAM3D is the selected model.		
Format	`SEAM3DOPTS ntrac nhcar nnutr ndaut`		
Sample	`SEAM3DOPTS 2 5 3 4`		
Field	*Variable*	*Value*	*Description*
1	`ntrac`	0,+	The number of tracers
2	`nhcar`	0,+	The number of hydrocarbons
3	`nnutr`	0,+	The number of nutrients
4	`ndaut`	0,+	The number of daughter products

Card Type	**S3DMCOL**		
Description	Defines the SEAM3D microbial processes.		
Required	YES, if SEAM3D is the selected model.		
Format	`S3DMCOL o2 no3 mn fe so4 ch4`		
Sample	`S3DMCOL 1 1 1 0 0 0`		
Field	*Variable*	*Value*	*Description*
1	`o2`	0/1	Flag for aerobes
2	`no3`	0/1	Flag for NO3 reducers
3	`mn`	0/1	Flag for Mn(IV) reducers
4	`fe`	0/1	Flag for Fe(III) reducers
5	`so4`	0/1	Flag for SO4 reducers
6	`ch4`	0/1	Flag for methanogens

Card Type	**S3DPROD**		
Description	Defines the SEAM3D product options.		
Required	YES, if SEAM3D is the selected model.		
Format	`S3DPROD nox mn fe h2s`		
Sample	`S3DPROD nox mn fe h2s`		
Field	*Variable*	*Value*	*Description*
1	`nox`	0/1	Flag for tracking NOX
2	`mn`	0/1	Flag for tracking Mn(II)
3	`fe`	0/1	Flag for tracking Fe(II)
4	`h2s`	0/1	Flag for tracking H2S

20.3 Coverages

The following cards are used to mark the beginning and ending of a set of cards defining a coverage, and some other general attributes of a coverage. All objects defined between the beginning and ending cards belong to the coverage.

Card Type	**BEGCOV**

Description	Beginning of a series of cards definining a coverage.
Required	YES

Card Type	**ACTCOV**
Description	Defines the active coverage.
Required	NO

Card Type	**ACTCOV**
Description	Indicates that the coverage being defined is the active coverage.
Required	NO

Card Type	**COVNAME**		
Description	Coverage name.		
Required	NO		
Format	`COVNAME "name"`		
Sample	`COVNAME "MODFLOW bc"`		
Field	*Variable*	*Value*	*Description*
1	`name`	str	Coverage name.

Card Type	**COVELEV**		
Description	Coverage display elevation.		
Required	NO		
Format	`COVELEV elev`		
Sample	`COVNAME 100.0`		
Field	*Variable*	*Value*	*Description*
1	`elev`	±	Coverage display elevation (for oblique viewing).

Card Type	**COVATTS**		
Description	Coverage model attributes set.		
Required	YES		
Format	`COVATTS type`		
Sample	`COVATTS 2DMESH`		
Field	*Variable*	*Value*	*Description*
1	`type`	GENERAL 2DGRID SEEP2D FEMWATER MFAREAL MFLAYER MFLOCAL OBSERVE	General 2D grid. 2D mesh FEMWATER attributes MODFLOW/MT3D areal attributes MODFLOW/MT3D/MODPATH layer attributes MODFLOW/MT3D local source/sink attributes Oberservation coverage

Card Type	**TRUELAYER**		
Description	Used with MODFLOW models. Defines the layer data entry method.		
Required	YES		
Format	`TRUELAYER flag`		
Sample	`TRUELAYER 0`		
Field	*Variable*	*Value*	*Description*
1	`flag`	0	Standard MODFLOW approach

		1	True layer method

Card Type	**ENDCOV**		
Description	End of cards defining a coverage.		
Required	YES		

20.4 Feature Object Geometry

The following cards are used to define the geometry of feature objects.

Card Type	**POINT**
Description	Beginning of a series of cards defining a point.
Required	NO

Card Type	**XY**		
Description	xy coordinates of a point or node.		
Required	YES		
Format	`XY x y`		
Sample	`XY 10.0 20.0`		
Field	*Variable*	*Value*	*Description*
`1-2`	`x, y`	±	xy coordinates of the point or node.

Card Type	**ID**		
Description	id of a feature object.		
Required	YES		
Format	`ID id`		
Sample	`ID 10`		
Field	*Variable*	*Value*	*Description*
`1`	`id`	+	id of the feature object

Card Type	**END**
Description	End of a series of cards defining a feature object or drawing object.
Required	YES

Card Type	**NODE**
Description	Beginning of a set of cards defining a node.
Required	NO

Card Type	**ARC**
Description	Beginning of a set of cards defining an arc.
Required	NO

Card Type	**NODES**
Description	Beginning and ending nodes of arc.
Required	YES
Format	`NODES n1 n2`

Sample	NODES 10 15		
Field	*Variable*	*Value*	*Description*
1-2	n1, n2	+	id of beginning and ending nodes of an arc.

Card Type	**ARCVERTICES**		
Description	Vertices between nodes of an arc identifier.		
Required	Only if the arc has vertices		
Format	ARCVERTICES nvert x_1 y_1 x_2 y_2 . . . x_n y_n		
Sample	ARCVERTICES 3 5.0 10.0 12.0 8.0 2.5 7.6		
Field	*Variable*	*Value*	*Description*
1	nvert	+	Number of intermediate vertices
2-3	x, y	±	Vertex coordinates. Fields 2-3 repeated for each vertex

Card Type	**POLYGON**
Description	Polygon Type idenfier.
Required	NO

Card Type	**ARCS**		
Description	Defines the arcs forming the outer boundary of a polygon. There should only be one card of this type for each polygon, except for the universal polygon.		
Required	YES		
Format	ARCS count		
Sample	ARCS 2 10 12		
Field	*Variable*	*Value*	*Description*
1	count	+	Number of arcs in the polygon.
2	id	+	Arc id. Field 2 is repeated for each arc in the polygon. The arcs should be listed in clockwise order.

Card Type	**HARCS**		
Description	Defines the arcs bounding the holes in the polygon. There should be one card of this type for each hole in the polygon.		
Required	NO.		
Format	HARCS count		
Sample	HARCS 2 10 12		
Field	*Variable*	*Value*	*Description*
1	count	+	Number of hole-arcs in the polygon.
2	id	+	Arc id. Field 2 is repeated for each arc in the polygon. The arcs should be listed in counter-clockwise order.

Card Type	**ARCGROUP**
Description	Beginning of a set of cards defining an arc group.
Required	NO

Card Type	**ARCS**
Description	Defines the set of arcs belonging to an arc group. The format of this card is identical to the ARCS card used to define polygons.
Required	YES, if an arc group is defined.

20.5 Source/Sink Coverage Attributes

The following cards are used to specify the attributes associated with points, arcs, and polygons in a *MODF/MT3D Local Source/Sink* coverage. They should be placed in the file between the beginning and ending cards for the associated objects.

Card Type	**FLUXOPTS**		
Description	Options for observed fluxes for the coverage		
Required	Only for MODFLOW/MODPATH/MT3D Source/Sink type coverages.		
Format	`FLUXOPTS fluxtype transtype`		
Sample	`FLUXOPTS TRANS FLUXRATE`		
Field	*Variable*	*Value*	*Description*
1	`fluxtype`	SS TRANS	Steady state Transient
2	`transtype`	RATE VOLUME	Transient flux rate Accumulated volume over a period of time (this field is only required if fluxtype=TRANS)

Card Type	**SPECHHEAD, GENHHEAD, RIVERSTAGE, WELLFLUX, STRFLOW, STRSTAGE,**		
Description	Specified head, general head, river stage, well flux, incoming stream flow rate, stream stage.		
Required	YES if a feature object is of the specified type.		
Format	`SPECHHEAD  constflag  value/id`		
Sample	`SPECHHEAD 1 138.6`		
Field	*Variable*	*Value*	*Description*
1	`constflag`	0,1	The type code: 0 = Transient attribute represented by xy series. 1 = Constant attribute represented by a single value.
2	`value/id`	±/+	The constant value or the id of an xy series.

Card Type	**WELLELEVS**		
Description	The elevation of the ground surface and the screened interval for the well.		
Required	NO. The presence of this card indicates that the Q should be partitioned to the layers using the well screen.		
Format	`WELLELEVS  groundsurf topscreen botscreen`		
Sample	`WELLELEVS 390.0 345.0 340.0`		
Field	*Variable*	*Value*	*Description*
1	`groundsurf`	±	The elevation of the ground surface.
2	`topscreen`	±	The elevation of the top of the well screen
3	`botscreen`	±	The elevation of the bottom of the well screen

Card Type	**SPECHCONC, GENHCONC, RIVCONC, WELLCONC**		
Description	Specified head concentration, general head concentration, river concentration, well concentration.		
Required	NO		
Format	`SPECHCONC constflag value/id species`		
Sample	`SPECHCONC 1 138.6 5`		
Field	*Variable*	*Value*	*Description*
1	constflag	0,1	The type code: 0 = transient value represented by xy series 1 = contant value
2	value/id	±/+	The constant value or the id of an xy series
3	species	+	The id of the species

Card Type	**GENHCOND, DRAINELEV, DRAINCOND, RIVERELEV, RIVERCOND, STRCOND, STRWIDTH, STRSIN, STRROUGH, STRTOP, STRBOT, HFB**		
Description	General head conductance, drain elevation, drain conductance, river elevation, river conductance, stream conductance, stream width, stream sinuosity, stream roughness coefficient, elevation of the top of the stream bed, elevation of the bottom of the stream bed, hydraulic characteristic of barrier		
Required	YES if a feature object is of the specified type.		
Format	`GENHCOND value`		
Sample	`GENHCOND 138.6`		
Field	*Variable*	*Value*	*Description*
1	`value`	±	The value of the attribute.

Card Type	**STRDIV**		
Description	Stream diversion flag.		
Required	Only if the arc is a diversion.		
Format	`STRDIV flag`		
Sample	`STRDIV 1`		
Field	*Variable*	*Value*	*Description*
1	`flag`	0 1	The arc is NOT a diversion The arc is a diversion

Card Type	**BELAYERS**		
Description	Values for the beginning and ending layer between which the coverage or point source/sink is to be applied.		
Required	NO		
Format	`BELAYERS beginlayer endlayer`		
Sample	`BELAYERS 1 4`		
Field	*Variable*	*Value*	*Description*
1 2	`beginlayer` `endlayer`	+ +	Beginning layer value Ending layer value

Card Type	**CBX, CBY**		
Description	Refine point attributes for refinement in the x and y directions starting with a cell boundary at the specified location		
Required	NO		
Format	`CBX basesize bias maxcellsize`		
Sample	`CBX 10.0 1.1 200.0`		
Field	*Variable*	*Value*	*Description*

1	basesize	+	The base cell size
2	bia	+	The bias term
3	maxcellsize	+	The maximum cell size

Card Type	**CCX, CCY**
Description	Refine point attributes for refinement in the x and y directions starting with a cell center at the specified location. Format is similar to CBX, CBY cards.

Card Type	**NAPL**		
Description	Defines a NAPL polygon. Used with SEAM3D		
Required	NO.		
Format	`NAPL conc dissrate`		
Sample	`NAPL 10.0 0.01`		
Field	*Variable*	*Value*	*Description*
1	conc	+	The initial concentration
2	dissrate	+	The dissolution rate

Card Type	**OBSFLUX**		
Description	Observed flux values.		
Required	NO		
Format	`OBSFLUX val/id/volume targtype int/stdev conf`		
Sample	`OBS 1 0 2.0 0.85`		
Field	*Variable*	*Value*	*Description*
1	val/id/volume	±/+/±	Either the value of the flux, or the id of an xy series, or the accumulated volume, depending on whether the fluxes are steady state or transient and whether the flux rate or volume method is being used for transient observed fluxes. If the volume method is chosen, the begin and end time cards must immediately follow this card.
2	targtype	0 1	Specified confidence interval Standard deviation
3	int/stdev	±	Either the interval or the standard deviation
4	conf	0-1	The confidence in the interval. Not required for std. dev. option.

Card Type	**FLUXBEGT**
Description	Beginning time for the flux accumulation period. The fields are identical to the fields used in the REFTIME card.
Required	This card must be used directly after the OBSFLUX card if the accumulated volume method is being used for transient flux observations.

Card Type	**FLUXENDT**
Description	Ending time for the flux accumulation period. The fields are identical to the fields used in the REFTIME card.
Required	This card must be used directly after the FLUXBEGT card.

20.6 Areal Coverage Attributes

The following cards are used to specify the attributes associated with polygons in a *MODF/MT3D Areal Attributes* coverage. They should be placed in the file between the beginning and ending cards for the associated polygons.

Card Type	**RCHRATE, ETRATE, ETELEV, ETEXTINCT**		
Description	Recharge rate, evapotranspiration rate, . evapotranspiration elevation, evapotranspiration extinction depth attributes.		
Required	YES if a feature object of the corresponding type is being defined.		
Format	`RCHRATE constflag  value/id`		
Sample	`RCHRATE 1 138.6`		
Field	*Variable*	*Value*	*Description*
1	`constflag`	0,1	The type code: 0 = Transient attribute represented by xy series. 1 = Constant attribute represented by a single value.
2	`value/id`	±/+	The constant value or the id of an xy series.

Card Type	**RCHCONC, ETCONC**		
Description	Recharge concentration, evapotranspiration concentration.		
Required	NO		
Format	`RCHCONC constflag value/id species`		
Sample	`RCHCONC 1 138.6 5`		
Field	*Variable*	*Value*	*Description*
1	constflag	0,1	The type code: 0 = transient value represented by xy series 1 = contant value
2	value/id	±/+	The constant value or the id of an xy series
3	species	+	The id of the species

Card Type	**HTOP**		
Description	Defines properties to apply for *top elevation* attributes.		
Required	NO, if a card is present top elevation will be assigned, otherwise it is not used.		
Format	`HTOP value`		
Sample	`HTOP 138.6`		
Field	*Variable*	*Value*	*Description*
2	`value`	+	The value of the elevation attribute.

20.7 Layer Coverage Attributes

The following cards are used to specify the attributes associated with polygons in a *MODF/MODP/MT3D Layer Attributes* coverage. They should be placed in the file between the beginning and ending cards for the associated polygons.

The following cards are used when the layer data entry method is the standard MODFLOW approach:

Card Type	**MTRN, MHC, MLK, MTE, MBE, MPSC, MSSC**		
Description	Defines properties to apply for *transmissivity, hydraulic conductivity, leakance, top elevation, bottom elevation, primary storage coefficient, secondary storage coefficient* layer attributes		
Required	NO, if a card is present that attribute will be assigned, otherwise it is not used.		
Format	`MTRN value`		
Sample	`MTRN 138.6`		
Field	*Variable*	*Value*	*Description*
1	`value`	+	The value of the layer attribute.

The following cards are used when the layer data entry method is the standard true layer approach:

Card Type	**MHK, MVK, MSS, MSY, MTE, MBE, MWDF, MAPOR, MLD, MZC, MBD,**		
Description	Defines properties to apply for *horiz. hydraulic conductivity, vert. hydraulic conductivity, specific storage, specific yield, top elevation, bottom elevation, wet/dry flags, porosity, longitudinal dispersivity, zone code, and bulk density* layer attributes		
Required	NO, if a card is present that attribute will be assigned, otherwise it is not used.		
Format	`MKH value`		
Sample	`MKH 138.6`		
Field	*Variable*	*Value*	*Description*
1	`value`	+	The value of the layer attribute.

Card Type	**RXNPARAM**		
Description	Defines the value of a spatially variable reaction parameter for RT3D.		
Required	NO		
Format	`RXNPARAM id value`		
Sample	`RXNPARAM 2 0.002`		
`Field`	*Variable*	*Value*	*Description*
1	id	+	The id of the reaction parameter.
2	value	±	The value of the reaction parameter

Card Type	**SHEAD, ICSORP, SP1, SP2, RC1, RC2**		
Description	Defines the value of a specie depedent parameter: *starting head, init. conc. sorbed phase, sorption constant 1, sorption constant 2, reaction rate constant 1, reaction rate constant 2.* One card should be listed for each specie.		
Required	NO		
Format	`SHEAD sid value`		
Sample	`SHEAD 1 0.0`		
`Field`	*Variable*	*Value*	*Description*
1	sid	+	The id of the specie
2	value	±	The value of the parameter

Card Type	**MS3DSUB**		
Description	Defines the max rate of substrate utilization for SEAM3D. One card should be entered for each acceptor-substrate combination.		
Required	NO		
Format	`MS3DSUB value`		
Sample	`MS3DSUB 0.001`		
`Field`	*Variable*	*Value*	*Description*
1	value	±	The value of the parameter

Card Type	**BELAYERS**
Description	See description in Section 20.5.

20.8 2D Grid Coverage Attributes

The following cards are used to specify the attributes associated with points in a *2D Grid* coverage. They should be placed in the file between the beginning and ending cards for the associated points.

Card Type	**CBX, CBY, CCX, CCY**
Description	Refine point attributes for refinement in the x and y directions. See card descriptions in the Section 20.5.

20.9 SEEP2D Coverage Attributes

The following cards are used to specify the attributes associated with objects in a *SEEP2D* coverage. They should be placed in the file between the beginning and ending cards for the associated objects.

Card Type	**CCX**
Description	Used to signify refinement around a point. The first field of this card (basesize) is used to define the refinement size. The presence or absence of the card defines whether or not the mesh is to be refined around the point. See card descriptions in Section 20.5.

Card Type	**MATID**		
Description	Material id assigned to a polygon		
Required	YES.		
Format	`MATID id`		
Sample	`MATID 1`		
Field	*Variable*	*Value*	*Description*
1	id	+	The material id.

20.10 FEMWATER Coverage Attributes

The following cards are used to specify attributes associated with objects in a *FEMWATER* coverage. The attributes should be placed in the file between the beginning and ending cards for the associated objects.

The following cards are associated with polygons:

Card Type	**FLUX**
Description	Fluid flux type bc. Assigned to polygons.
Required	NO

Format	`FLUX type constflag value/id`		
Sample	`FLUX 0 1 12.0`		
Field	*Variable*	*Value*	*Description*
1	`type`	0,1	The type of FEMWATER flux bc: 0 = specified flux 1 = variable
2	`constflag`	0,1	The type code: 0 = Transient attribute represented by xy series. 1 = Constant attribute represented by a single value.
3	`value/id`	±/+	The constant value or the id of an xy series.

Card Type	**CONC**		
Description	Contaminant type bc. Assigned to polygons		
Required	NO		
Format	`CONC type constflag value/id`		
Sample	`CONC 0 1 12.0`		
Field	*Variable*	*Value*	*Description*
1	`type`	0,1	The type of FEMWATER conc. bc: 0 = specified mass flux rate 1 = variable
2	`constflag`	0,1	The type code: 0 = Transient attribute represented by xy series. 1 = Constant attribute represented by a single value.
3	`value/id`	±/+	The constant value or the id of an xy series.

The following cards are associated with arcs. If no card exists, the arc is assumed to be a no-flow arc.

Card Type	**HEAD**
Description	A specified head boundary for FEMWATER. No fields. The head values are stored with the nodes.
Required	NO

Card Type	**FLUX**		
Description	Fluid flux type bc.		
Required	NO		
Format	`FLUX type constflag value/id`		
Sample	`FLUX 0 1 12.0`		
Field	*Variable*	*Value*	*Description*
1	`type`	0,1	The type of FEMWATER flux bc: 0 = specified flux 1 = variable
2	`constflag`	0,1	The type code: 0 = Transient attribute represented by xy series. 1 = Constant attribute represented by a single value.
3	`value/id`	±/+	The constant value or the id of an xy series.

Card Type	**CONC**
Description	Contaminant type bc.
Required	NO

Format	`CONC type constflag value/id`		
Sample	`CONC 0 1 12.0`		
Field	*Variable*	*Value*	*Description*
1	`type`	0,1,2	The type of FEMWATER conc. bc: 0 = specified concentration 1 = specified mass flux rate 2 = variable
2	`constflag`	0,1	The type code: 0 = Transient attribute represented by xy series. 1 = Constant attribute represented by a single value.
3	`value/id`	±/+	The constant value or the id of an xy series.

If the following card is not present, we will assume the *All Zones* option.

Card Type	**ZONES**		
Description	Defines the zones the arc bc is to be assigned to. In the absence of this card, the bc is applied to all zones.		
Required	NO.		

Format	`ZONES n`
	`z1`
	`z2`
	`.`
	`zn`		

Sample	`ZONES 2`
	`2`
	`5`		

Field	*Variable*	*Value*	*Description*
1	`n`	+	The number of zones
2+	`z1-zn`	+	The zone ids

The following cards are associated with nodes:

Card Type	**HEAD**		
Description	A specified head boundary for FEMWATER.		
Required	YES, if the node is attached to one or more specified head arcs.		
Format	`HEAD constflag value/id`		
Sample	`HEAD 1 23.0`		
Field	*Variable*	*Value*	*Description*
1	`constflag`	0,1	The type code: 0 = Transient attribute represented by xy series. 1 = Constant attribute represented by a single value.
2	`value/id`	±/+	The constant value or the id of an xy series.

The following cards are associated with points.

Card Type	**CCX**
Description	Used to signify refinement around a point. The first field of this card (basesize) is used to define the refinement size. The presence or absence of the card defines whether or not the mesh is to be refined around the point. See card descriptions in the Section 20.5.

Card Type	**WELLELEVS**		
Description	The elevation of the ground surface and the screened interval for the well.		
Required	NO		
Format	`WELLELEVS groundsurf topscreen botscreen`		
Sample	`WELLELEVS 390.0 345.0 340.0`		
Field	*Variable*	*Value*	*Description*
1	`groundsurf`	±	The elevation of the ground surface.
2	`topscreen`	±	The elevation of the top of the well screen
3	`botscreen`	±	The elevation of the bottom of the well screen

Card Type	**WELLFLUX**
Description	Used to specify flux rate at a well. The format is identical to the WELLFLUX card used in the *MODF/MT3D Local Source/Sink* coverage. See card descriptions in the Section 20.5.

Card Type	**CONC**		
Description	Concentration associated with a specified head or specified flux boundary condition.		
Required	NO. If not present, assume constant rate of zero.		
Format	`CONC constflag value/id`		
Sample	`CONC 1 138.6`		
Field	*Variable*	*Value*	*Description*
1	`constflag`	0,1	The type code: 0 = Transient attribute represented by xy series. 1 = Constant attribute represented by a single value.
2	`value/id`	±/+	The constant value or the id of an xy series.

20.11 Observation Coverage Attributes

The following cards are used to specify attributes associated with objects in an *Observation* coverage. The attributes should be placed in the file between the beginning and ending cards for the associated objects.

Card Type	**OBJECT**		
Description	Defines the object that the coverage is associated with.		
Required	YES		
Format	`OBJECT type`		
Sample	`OBJECT 3DGRID`		
Field	*Variable*	*Value*	*Description*
1	`type`	2DGRID 2DMESH 3DGRID 3DMESH	2D grid 2D mesh 3D grid 3D mesh

Card Type	**INTERP**
Description	Point interpolation options.
Required	YES
Format	`INTERP option`

Sample	`INTERP 0`		
Field	*Variable*	*Value*	*Description*
1	`option`	0 1 2	Interp from neighbor nodes/cells (within layer only) Interp from neighbor nodes/cells Use value from nearest node/cell

Card Type	**MTYPE**		
Description	Defines the measurement types associated with the coverage		
Required	NO		
Format	`MTYPE num` `id "name" trans`		

Sample	`MTYPE 2`
	`1 "head" 0`
	`2 "conc" 1`		

Field	*Variable*	*Value*	*Description*
1	`num`	+	The number of measurement types.
2	`id`	+	The id of the measurement type.
3	`name`	str	The name of the measurement type.
4	`trans`	0 1	steady state transient
`Repeat 2-4 num times`			

Card Type	**AMTYPE**		
Description	Defines the active measurement type		
Required	NO		
Format	`AMTYPE id`		
Sample	`AMTYPE id`		
Field	*Variable*	*Value*	*Description*
1	`id`	+	The id of the active measurement type.

Card Type	**OBS**		
Description	Observed values.		
Required	NO		
Format	`OBS setid observed val/id targtype int/stdev conf`		
Sample	`OBS 1 1 1 0 2.0 0.85`		
Field	*Variable*	*Value*	*Description*
1	`setid`	+	The id of the measurement type
2	`observed`	0 1	Not observed Observed
3	`val/id`	±/+	Either the value or the id of an xy series depending on the trans flag for the associated meas. type.
4	`targtype`	0 1	Specified confidence interval Standard deviation
5	`int/stdev`	±	Either the interval or the standard deviation
6	`conf`	0-1	The confidence in the interval.

Card Type	**OBSNAME**		
Description	Name of point		
Required	NO		
Format	`OBSNAME "name"`		
Sample	`OBSNAME "point #1"`		
Field	*Variable*	*Value*	*Description*
1	`name`	str	The name of the point

Card Type	**OBSCOLOR**		
Description	Color of point.		
Required	NO		
Format	`OBSCOLOR r g b`		
Sample	`OBSCOLOR 0 0 0`		
Field	*Variable*	*Value*	*Description*
1	r	0-255	Red component
2	g	0-255	Green component
3	b	0-255	Blue component

Card Type	**OBSLAYER**		
Description	Layer determination of point		
Required	NO (only required for 3D grid coverages)		
Format	`OBSLAYER layer`		
Sample	`OBSLAYER 0`		
Field	*Variable*	*Value*	*Description*
1	layer	0 +	Layer will be determined by z coordinate Layer id

20.12 Drawing Objects

The following cards are used to define rectangles, ovals, lines, and text strings.

Card Type	**RECT**
Description	The beginning of a set of cards defining a rectangle.
Required	NO

Card Type	**OVAL**
Description	The beginning of a set of cards defining an oval.
Required	NO

Card Type	**C# (C1, C2, C3, C4, for the four corner points)**		
Description	Corner point identifiers of rectangle or oval.		
Required	YES if a rectangle or oval has been defined.		
Format	`C1 x y z`		
Sample	`C1 10 10 0`		
Field	*Variable*	*Value*	*Description*
1-3	x, y, z	±	Coordinates of corner point.

Card Type	**THETA**		
Description	Viewing angle (bearing) that the rect or oval object was created in.		
Required	YES for rects and ovals.		
Format	`THETA theta`		
Sample	`THETA 0.0`		
Field	*Variable*	*Value*	*Description*
1	theta	±	The angle in degrees.

Card Type	**ALPHA**		
Description	Viewing angle (dip) that the rect or oval object was created in.		
Required	YES for rects and ovals.		
Format	`ALPHA alpha`		
Sample	`ALPHA 90.0`		
Field	*Variable*	*Value*	*Description*
1	`alpha`	±	The angle in degrees.

Card Type	**THICK**		
Description	Line thickness identifier.		
Required	YES if a line, rectangle or oval has been defined.		
Format	`THICK width`		
Sample	`THICK 1`		
Field	*Variable*	*Value*	*Description*
1	`width`	+	Line thickness in pixels.

Card Type	**STYLE**		
Description	Line style identifier.		
Required	YES if a line, rectangle or oval has been defined.		
Format	`STYLE style`		
Sample	`STYLE 0`		
Field	*Variable*	*Value*	*Description*
1	`style`	+	0 - Solid line style. 1 - Dashed line style.

Card Type	**LINECOL**		
Description	Line color identifier.		
Required	YES if a line, rectangle or oval has been defined.		
Format	`LINECOL r g b`		
Sample	`LINECOL 255 255 255`		
Field	*Variable*	*Value*	*Description*
1-3	`r, g, b`	0-255	Red, green and blue color components

Card Type	**FILLCOL**		
Description	Polygon fill color identifier.		
Required	YES if a rectangle or oval has been defined.		
Format	`FILLCOL r g b`		
Sample	`FILLCOL 255 255 255`		
Field	*Variable*	*Value*	*Description*
1-3	`r, g, b`	0-255	Red, green, blue color components

Card Type	**FILLPAT**		
Description	Polygon fill pattern identifier.		
Required	YES if a rectangle or oval has been defined.		
Format	`FILLPAT pattern`		
Sample	`FILLPAT 0`		
Field	*Variable*	*Value*	*Description*
1	`pattern`	+	Pattern index.

Card Type	**LINE**
Description	Beginning of a set of cards defining a line object.
Required	NO

Card Type	**VERTS**		
Description	Number of points in a line.		
Required	YES if a line has been defined.		
Format	`VERTS count`		
Sample	`VERTS 3`		
Field	*Variable*	*Value*	*Description*
1	count	+	Number of points in the line.

Card Type	**PT**		
Description	Defines a point on a line object		
Required	YES if a line has been defined.		
Format	`PT x y z`		
Sample	`PT 213.2. 523.2 0`		
Field	*Variable*	*Value*	*Description*
1-3	x y z	±	Coordinates of the point.

Card Type	**ARRHED**		
Description	Arrow head type identifier.		
Required	YES if a line has been defined.		
Format	`ARRHED style`		
Sample	`ARRHED 0`		
Field	*Variable*	*Value*	*Description*
1	style	0	No arrow head.
		1	Arrow head at beginning of line.
		2	Arrow head at end of line.
		3	Arrow heads at both ends of line.

Card Type	**HEDWID**		
Description	Arrow head base width identifier.		
Required	YES if a line has been defined.		
Format	`HEDWID width`		
Sample	`HEDWID 10`		
Field	*Variable*	*Value*	*Description*
1	width	+	Width of the base of the arrow head in pixels.

Card Type	**HEDLEN**		
Description	Arrow head length identifier.		
Required	YES if a line has been defined.		
Format	`HEDLEN length`		
Sample	`HEDLEN 25`		
Field	*Variable*	*Value*	*Description*
1	length	+	Length of the arrow head in pixels.

Card Type	**TEXT**

Description	Beginning of a set of cards defining a text object.
Required	NO

Card Type	**STRING**		
Description	Text string identifier.		
Required	YES if a text string has been defined.		
Format	`STRING "string"`		
Sample	`STRING "map title"`		
Field	*Variable*	*Value*	*Description*
1	`string`	str	Text string.

Card Type	**LOCAL**		
Description	Text string location identifier.		
Required	YES if a text string has been defined.		
Format	`LOCAL x y`		
Sample	`LOCAL 100 200`		
Field	*Variable*	*Value*	*Description*
1-2	`x, y`	±	Coordinates of the beginning of the text string.

Card Type	**PCFONT**		
Description	PC font identifier.		
Required	YES if a text string has been defined.		
Format	`PCFONT id`		
Sample	`PCFONT 2`		
Field	*Variable*	*Value*	*Description*
1	`id`	+	The id of the PC font.

Card Type	**UNIXFONT**		
Description	UNIX font identifier.		
Required	YES if a text string has been defined.		
Format	`UNIXFONT id`		
Sample	`UNIXFONT 2`		
Field	*Variable*	*Value*	*Description*
1	`id`	+	The id of the UNIX font.

21 ASCII Data Set Files

Data sets can be stored to either ASCII or binary files. The default format is binary. Data sets can be saved in ASCII format by selecting the *Export* command in the *Data Browser*. For both file formats, multiple data sets can be stored in a single file and both scalar and vector data sets can be saved to the same file. The file format is identical for 2D and 3D data sets. The ASCII data set format is shown in Figure 21.1. A sample data set file is shown in Figure 21.2.

For scalar data set files, one value is listed per vertex, cell, node, or scatter point. The points are listed sequentially in ascending order according to the ids of the nodes, points, vertices, or cells. For vector data set files, one set of XYZ vector components is listed per vertex, cell, node, or scatter point. If necessary, a set of status flags can be included in the file. If the status flag is false (0), the corresponding item (node, cell, etc.) is inactive. If status flags are not included in the file, it is assumed that all items are active.

```
DATASET   /* File type identifier */
OBJTYPE type     /* Type of object data set is associated with */
BEGSCL   /* Beginning of scalar data set */
OBJID id /* Object id */
ND numdata       /* Number of data values */
NC numcells      /* Number of cells or elements */
NAME "name"      /* Data set name */
ACTTS time       /* Marks the active time step */
MAPTS time       /* Marks the time step which is mapped as elevations */
TS istat time    /* Time step of the following data. */
stat1    /* Status flags */
stat2
.
.
statnumcells
val1     /* Scalar data values */
val2
.
.
valnumdata
/* Repeat TS card for each time step */
ENDDS    /* End of data set */
BEGVEC   /* Beginning of vector dataset */
VECTYPE type     /* Vector at node/gridnode or element/cell */
OBJID id /* Object id */
ND numdata       /* Number of data values */
NC numcells      /* Number of cells or elements */
NAME "name"      /* Data set name */
TS istat time    /* Time step of the following data. */
stat1    /* Status flags */
stat2
.
.
statnumcells
vx1 vy1 vz1
vx2 vy2 vz2
.
.
vnumdata vnumdata vnumdata
/* Repeat TS card for each time step */
ENDDS    /* End of data set */
/* Repeat BEGSCL and BEGVEC sequences for each data set */
```

Figure 21.1 ASCII Data Set File Format.

```
DATASET
OBJTYPE grid2d
BEGSCL
ACTTS 1.00000000e+00
ND 8
NC 8
NAME "trichloroethylene"
TS 1 1.00000000e+00
0
0
0
1
1
1
1
```

```
0
0.00000000e+00
0.00000000e+00
0.00000000e+00
3.24000000e+00
4.39000000e+00
2.96000000e+00
7.48000000e+00
0.00000000e+00
ENDDS
BEGVEC
VECTYPE 0
ND 8
NC 8
NAME "velocity"
TS 1 5.00000000e+00
0
0
0
1
1
1
1
0
1.60000000e+01 1.60000000e+01 3.20000000e+01
6.40000000e+01 6.40000000e+01 1.28000000e+02
1.44000000e+02 1.44000000e+02 2.88000000e+02
1.96000000e+02 1.96000000e+02 3.92000000e+02
2.25000000e+02 2.25000000e+02 4.50000000e+02
9.21600000e+03 9.21600000e+03 1.84320000e+04
9.60400000e+03 9.60400000e+03 1.92080000e+04
9.80100000e+03 9.80100000e+03 1.96020000e+04
ENDDS
```

Figure 21.2 Sample ASCII Data Set File.

If variograms have been defined for a data set or time step of a data set within GMS, the variograms are saved in the data set file. The variogram cards are not documented.

The card types used in the scalar data set file format are as follows:

Card Type	**DATASET**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**OBJTYPE**		
Description	Identifies the type of objects that the data sets in the file are associated with.		
Required	YES. If card does not exist, the file can only be read through the Data Browser. The data sets would then be assigned to the objects corresponding to the active module.		
Format	`OBJTYPE type`		
Sample	`OBJTYPE tin`		
Field	*Variable*	*Value*	*Description*
1	`type`	tin mesh2d grid2d scat2d mesh3d grid3d scat3d	TINs 2D meshes 2D grids 2D scatter points 3D meshes 3D grids 3D scatter points

Card Type	**BEGSCL**		
Description	Scalar data set file identifier. Marks beginning of scalar data set. No fields.		
Required	YES		

Card Type	**BEGVEC**		
Description	Vector data set file identifier. Marks beginning of vector data set. No fields.		
Required	YES		

Card Type	**ACTTS**		
Description	Used to mark the active data set. The card should be placed after the BEGSCL or BEGVEC card of the active data set and the active time step should be listed.		
Required	NO		
Format	`ACTTS time`		
Sample	`ACTTS 0.00`		
Field	*Variable*	*Value*	*Description*
1	`time`	±	The time corresponding to the active time step. Use 0.0 for steady state data sets.

Card Type	**MAPTS**		
Description	Used to mark the data set which is mapped to the object elevations. The card should be placed after the BEGSCL or BEGVEC card of the mapped data set and the mapped time step should be listed.		
Required	NO		
Format	`MAPTS time`		
Sample	`MAPTS 0.00`		
Field	*Variable*	*Value*	*Description*
1	`time`	±	The time corresponding to the mapped time step. Use 0.0 for steady state data sets.

Card Type	**OBJID**		
Description	The unique id of the object the data set is associated with.		
Required	This card is required for data sets associated with TINs and scatter point sets.		
Format	`OBJID id`		
Sample	`OBJID 2383`		
Field	*Variable*	*Value*	*Description*
1	`id`	+	The unique id of the object.

Card Type	**VECTYPE**			
Description	Identifies the type of vector data that will be read and where to apply it.			
Required	This card is only required if the vector data is associated with elements/cells. If this card is not present, it is assumed that the data are associated with nodes/gridnodes.			
Field	*Variable*	*Size*	*Value*	*Description*
1	type	4 byte int	0	The vectors will be applied to the nodes/gridnodes.
			1	The vectors will be applied to the elements/cells.

Card Type	**ND**		
Description	The number of data values that will be listed per time step. This number should correspond to the total number of vertices, nodes, cells centers (cell-centered grid), cell corners (mesh-centered grid), maximum node id (meshes) or scatter points.		
Required	YES		
Format	`ND numdata`		
Sample	`ND 10098`		
Field	*Variable*	*Value*	*Description*
1	numdata	+	The number of items. At each time step, numdata values are printed.

Card Type	**NC**		
Description	This number should correspond to the maximum element id (meshes) or the number of cells (grids).		
Required	YES		
Format	`NC numcells`		
Sample	`NC 3982`		
Field	*Variable*	*Value*	*Description*
1	numcells	+	The number of elements or cells.

Card Type	**NAME**		
Description	The name of the data set.		
Required	YES		
Format	`NAME "name"`		
Sample	`NAME "Total head"`		
Field	*Variable*	*Value*	*Description*
1	"name"	str	The name of the dataset in double quotes.

Card Type	**TS**		
Description	Marks the beginning of a new time step, indicates if stat flags are given, and defines the time step value, status flags, and scalar data values for each item.		
Required	YES		

Format	`TS istat time`
	`stat1`
	`stat2`
	`.`
	`.`
	`stat numcells`
	`val1`
	`val2`
	`.`
	`.`
	`valnumdata`		

Sample	`TS 1 12.5`
	`0`
	`1`
	`1`
	`1`
	`34.5`
	`74.3`
	`58.4`
	`72.9`		

Field	*Variable*	*Value*	*Description*
1	istat	0 1	Use status flags from previous time step. For first time step, this indicates that all cells are active. Status flags will be listed.

2	time	+	The time step value. If only one time step exists, time is not required
3 - (nd+2)	stat	0,1	The status of each item. If active, stat=1. If inactive stat=0. Omitted if i=0 on STAT card.
(nd+2) - (2nd+2)	val	±	The scalar data values of each item.

Card Type	**ENDDS**
Description	Marks the end of a scalar or vector data set. No fields.
Required	YES

21.1 Using Vector and Scalar Data with Grids

For meshes, TINs, and scatter point sets, the order that the values are listed in the file is simply the sequential order of the node, TIN, or scatter point ids. However, vector and scalar data can also be associated with the nodes or cells of a 2D or 3D grid. T For 2D grids, data values are ordered using a row-column (I-J) priority. For 3D grids, data values are ordered using a layer-row-column (K-I-J) priority.

The following C source code examples illustrate how a 2D or 3D array of scalar values corresponding to the nodes of a grid would be written to the main portion of an ASCII scalar file.

21.1.1 2D Grid Example:

```
for(i=0; i<nrow; i++){
  for(j=0; j<ncol; j++){
    fprintf(fp, "%f\n", scalar[i][j]);
  }
}
```

21.1.2 3D Grid Example:

```
for(k=0; k<nlay; k++){
  for(i=0; i<nrow; i++){
    for(j=0; j<ncol; j++){
      fprintf(fp, "%f\n", scalar[k][i][j]);
    }
  }
}
```

22 Binary Data Set Files

Data sets saved with a GMS project are saved in the binary format. The binary data set file format is shown in Figure 22.1. The binary format is patterned after the ASCII format in that the data are grouped into "cards". However, the cards are identified by a number rather than a card title. The card ids are four byte integers.

Card	Item	Size	Description
	version	4 byte integer	The GMS binary data set file format version. value = 3000.
100	objecttype	4 byte integer	Identifies the type of objects that the data sets in the file are associated with. Options are as follows: 1 TINs 2 Boreholes 3 2D meshes 4 2D grids 5 2D scatter points 6 3D meshes 7 3D grids 8 3D scatter points
110	SFLT	4 byte integer	The number of bytes that will be used in the remainder of the file for each floating point value (4, 8, or 16).
120	SFLG	4 byte integer	The number of bytes that will be used in the remainder of the file for status flags.
130 or 140	BEGSCL or BEGVEC		Marks the beginning of a set of cards defining a scalar or vector data set.
220	ACTTS	SFLT real	Marks the active data set and time step
230	MAPTS	SFLT real	Marks the mapped data set and time step
150	VECTYPE	4 byte integer	(0 or 1) In the case of vector data set files, indicates whether the vectors will be applied at the nodes/gridnodes or the elements/cells.
160	OBJID	4 byte integer	The id of the associated object. Value is ignored for grids and meshes.
170	NUMDATA	4 byte integer	The number of data values that will be listed per time step. This number should correspond to the number of vertices, nodes, cell centers (cell-centered grid), cell corners (mesh-centered grid) or scatter points.
180	NUMCELLS	4 byte integer	This number should correspond to the number of elements (meshes) or the number of cells (mesh-centered grids). Value is ignored for other object types.
190	NAME	40 bytes	The name of the dataset. Use one character per byte. Mark the end of the string with the '\0' character.
200	TS		Marks the beginning of a time step.
	ISTAT	SFLG integer	(0 or 1) Indicates whether or not status flags will be included in the file.
	TIME	SFLT real	Time corresponding to the time step.
	statflag1	SFLG integer	Status flag (0 or 1) for node 1
	statflag2	SFLG integer	Status flag (0 or 1) for node 2
	….		
	val1	SFLT real	Scalar value for item 1
	val2	SFLT real	Scalar value for item 2
	….		
			Repeat card 200 for each timestep in the data set.
210	ENDDS		Signal the end of a set of cards defining a data set.

Figure 22.1 The Binary Scalar or Vector Data Set File Format.

The cards in the binary data set file are as follows:

Card Type	**VERSION**			
Card ID	3000			
Description	File type identifier. No fields.			
Required	YES			

Card Type	**OBJTYPE**			
Card ID	100			
Description	Identifies the type of objects that the data sets in the file are associated with.			
Required	YES. If card does not exist, the file can only be read through the Data Browser. The data sets would then be assigned to the objects corresponding to the active module.			
Field	*Variable*	*Size*	*Value*	*Description*
1	id	4 byte int	1	TINs
			2	Boreholes
			3	2D meshes
			4	2D grids
			5	2D scatter points
			6	3D meshes
			7	3D grids
			8	3D scatter points

Card Type	**SFLT**			
Card ID	110			
Description	Identifies the number of bytes that will be used in the remainder of the file for each floating point value (4, 8, or 16).			
Required	YES			
Field	*Variable*	*Size*	*Value*	*Description*
1	sizefloat	4 byte int	4, 8, or 16	Number of bytes

Card Type	**SFLG**			
Card ID	120			
Description	Identifies the number of bytes that will be used in the remainder of the file for status flags (1, 2, or 4).			
Required	YES			
Field	*Variable*	*Size*	*Value*	*Description*
1	sizeflag	4 byte int	1, 2, or 4	Number of bytes

Card Type	**BEGSCL**
Card ID	130
Description	Marks the beginning of a set of cards defining a scalar data set.
Required	YES

Card Type	**BEGVEC**
Card ID	140
Description	Marks the beginning of a set of cards defining a vector data set.
Required	YES

Card Type	**ACTTS**		
Card ID	220		
Description	Used to mark the active data set. The card should be placed after the BEGSCL or BEGVEC card of the active data set and the active time step should be listed.		
Required	NO		
Field	*Variable*	*Value*	*Description*
1	time	±	The time corresponding to the active time step. Use 0.0 for steady state data sets.

Card Type	**MAPTS**		
Card ID	230		
Description	Used to mark the data set which is mapped to the object elevations. The card should be placed after the BEGSCL or BEGVEC card of the mapped data set and the mapped time step should be listed.		
Required	NO		
Field	*Variable*	*Value*	*Description*
1	time	±	The time corresponding to the mapped time step. Use 0.0 for steady state data sets.

Card Type	**VECTYPE**			
Card ID	150			
Description	Identifies the type of vector data that will be read and where to apply it.			
Required	This card is only required if the vector data is associated with elements/cells. If this card is not present, it is assumed that the data are associated with nodes/gridnodes.			
Field	*Variable*	*Size*	*Value*	*Description*
1	type	4 byte int	0	The vectors will be applied to the nodes/gridnodes.
			1	The vectors will be applied to the elements/cells.

Card Type	**OBJID**			
Card ID	160			
Description	The id of the associated object.			
Required	This card is required in the case of TINs, 2D scatter points, and 3D scatter points. With each of these objects, multiple objects may be defined at once. Hence the id is necessary to relate the data set to the proper object.			
Field	*Variable*	*Size*	*Value*	*Description*
1	id	4 byte int	+	The id of the object.

Card Type	**NUMDATA**			
Card ID	170			
Description	The number of data values that will be listed per time step. This number should correspond to the number of vertices, nodes, cell centers (cell-centered grid), cell corners (mesh-centered grid), maximum node id (meshes) or scatter points.			
Required	YES			
Field	*Variable*	*Size*	*Value*	*Description*
1	numdata	4 byte int	+	The number of items. At each timestep, numdata are listed.

Card Type	**NUMCELLS**			
Card ID	180			
Description	This number should correspond to the element id (meshes) or the number of cells (grids).			
Required	YES			
Field	*Variable*	*Size*	*Value*	*Description*
1	numcells	4 byte int	+	The number of elements or cells.

Card Type	**NAME**			
Card ID	190			
Description	The name of the data set.			
Required	YES			
Field	*Variable*	*Size*	*Value*	*Description*
1	name	40 bytes	str	The name of the data set. Use one character per byte. Mark the end of the string with the '\0' character.

Card Type	**TS**			
Card ID	200			
Description	Defines the set of scalar values associated with a timestep. Should be repeated for each time step.			
Required	YES			
Field	*Variable*	*Size*	*Value*	*Description*
1	istat	SFLG int	0 1	Use status flags from previous time step. For the first time step, this value indicates that all cells are active. Status flags will be listed.
2	time	SFLT int	+	The time step value. This number is ignored if ther is only one time step.
	stat	SFLG int	0 1	Inactive Active One status flag should be listed for each cell or element. These flags are included only when istat = 1.
	val	SFLT real	±	The scalar values

Card Type	**ENDDS**
Card ID	210
Description	Signals the end of a set of cards defining a data set
Required	YES

23 XY Series Files

The *XY Series Editor* is used in several places in GMS. The *XY Series Editor* is a general purpose editor for entering curves or pairs of lists of data. The *XY Series Editor* allows a curve to be imported from a file, created and edited graphically, or created and edited using two columns of edit fields in a spreadsheet-like interface.

XY series files can be used to prepare a set of curves for import to the *XY Series Editor*. XY series files are also used to export curves generated within the Editor for future use.

The format of the XY series file is shown in Figure 23.1. Curves are defined in an XY series file using one of three types of cards: XY1, XY2, or XY3. With the XY1 card, both the x and y values are listed for each point on the curve. There is no limit to the spacing or interval used between subsequent x values. The XY2 card is identical to the XY1 card except that the number of points and the x values are assumed to be static and cannot be altered by the user. With the XY3 card, the x values are defined by a beginning x value, an initial increment in x, and a percent change in x per increment. Only the y values are explicitly listed. A sample xy series file is shown in Figure 23.2.

The REFTIME card shown just after the XY* cards is an optional card that can be used to associate a date/time value with a time series. This represents the date/time corresponding to t=0 and can be used by GMS to convert scalar time values to a date/time format for display and data entry. The format for the REFTIME card is described in Section 24.

```
XY1 id n dx dy rep begc "name"  /* XY Series vers. #1 */
REFTIME bu td yr mo d hr min s
x1 y1    /* XY values */
x2 y2
.
.
xn yn
XY2 id n dx dy rep begc "name"  /* XY Series vers. #2 */
x1 y1    /* XY values */
x2 y2
.
.
xn yn
XY3 id n x1 incx pcx dx dy rep begc "name"     /* XY Series vers. #3 */
y1       /* Y values */
y2
.
.
yn
```

Figure 23.1 The XY Series File Format.

```
XY1 1 4 0 0 0 0 "concentration"
REFTIME d 0 1999 12 23 17 36 29
0.000 0.000
1.000 2.000
2.000.5.000
3.000.9.000
4.000 16.000
5.000 24.000
```

Figure 23.2 Sample XY Series File.

The card types used in the XY series file format are as follows:

Card Type	**XY1**		
Description	Defines a curve with a list of XY values. Any number of points and any x spacing between points may be used.		
Required	YES		

Format	`XY1 id n dx dy rep begc "name"`
	`REFTIME bu td yr mo d hr min s`
	x_1 y_1
	x_2 y_2
	.
	.
	x_n y_n		

Sample	`XY1 1 5 0 0 0 0 "head"`
	`REFTIME d 0 1999 12 23 17 36 29`
	`0.0 0.0`
	`1.0 2.0`
	`2.5 7.0`
	`3.0 8.0`
	`4.5 9.5`		

Field	*Variable*	*Value*	*Description*
1	id	+	The id of the XY series.
2	n	+	The number of point in the series.
3	dx	0,1	A flag defining whether the x values listed are to be interpreted as incremental (dx=1) or absolute (dx=0).
4	dy	0,1	A flag defining whether the y values listed are to be interpreted as incremental (dy=1) or absolute (dy=0).
5	rep	0,1	A flag defining whether the xy series is to be interpreted as cyclic or repeating (0=no, 1=yes)
6	begc	±	The x value in the series where the cyclic portion of the curve begins. Value is ignored if rep=0.
7	name	str	The name of the series.
8	reftime		Optional reference time card.
9+	x,y	±	The xy values of the points defining the curve. Repeat n times.

Card Type	**XY2**
Description	Defines a curve with a list of XY values. This card is identical to the XY1 card except that the number of points and the x values are assumed to be static and cannot be altered by the user.

Card Type	**XY3**
Description	Defines a curve with a list of Y values. The x values are defined by a beginning value, an increment, and a bias.
Required	YES

Format	`XY3 id n x1 incx biasx dx dy rep begc "name"`
	`y1`
	`y2`
	`.`
	`.`
	`yn`		

Sample	`XY3 1 10 0 1 0 0 0 0 0 "head"`
	`0.0`
	`2.0`
	`7.0`
	`8.0`
	`9.5`
	`9.1`		

Field	*Variable*	*Value*	*Description*
1	id	+	The id of the XY series.
2	n	+	The number of point in the series.
3	x1	±	The first x value.
4	incx	±	The increment in x used to compute the next x value.
5	pcx	+	The per cent change in x used to compute subsequent x values. Expressed as a decimal, i.e., 0.05 = 5%.
6	dx	0,1	A flag defining whether the x values listed are to be interpreted as incremental (dx=1) or absolute (dx=0).
7	dy	0,1	A flag defining whether the y values listed are to be interpreted as incremental (dy=1) or absolute (dy=0).
8	rep	0,1	A flag defining whether the xy series is to be interpreted as cyclic or repeating (0=no, 1=yes)
9	begc	±	The x value in the series where the cyclic portion of the curve begins. Value is ignored if rep=0.
10	name	str	The name of the series.
11	y	±	The y values of the points defining the curve. Repeat n times.

24 MODFLOW Super Files

A MODFLOW super file allows all of the packages involved in a MODFLOW simulation to be input at once. It is used by GMS and by the version of MODFLOW distributed with GMS. It is not an original MODFLOW file.

The format of the MODFLOW super file is shown in Figure 24.1 and a sample MODFLOW super file is shown in Figure 24.2.. The file contains a list of the package files used in the simulation. It also contains data needed by GMS which are not contained in any of the package files such as the layering and orientation of the 3D grid.

The format of the MODFLOW package files is described in McDonald & Harbaugh (1988).

```
MODSUP    /* File type identifier */
IJK ±idir ±jdir ±kdir    /* grid orientation */
LIST unit# "filename"    /* ASCII output file from MODFLOW */
BAS1 unit# "filename"    /* Basic package file */
BCF1 unit# "filename"    /* BCF1 package file */
BCF2 unit# "filename"    /* BCF2 package file */
BCF3 unit# "filename"    /* BCF3 package file */
OUT1 unit# "filename"    /* Output Control package file */
HEAD unit# "filename"    /* Binary Head file from MODFLOW*/
DRAW unit# "filename"    /* Binary Drawdown file from MODFLOW */
CCF  unit# "filename"    /* Cell-by-cell flow file from MODFLOW */
SIP1 unit# "filename"    /* SIP package file */
PCG2 unit# "filename"    /* PCG2 package file */
SOR1 unit# "filename"    /* SOR package file */
RIV1 unit# "filename"    /* River package file */
RCH1 unit# "filename"    /* Recharge package file */
WEL1 unit# "filename"    /* Well package file */
DRN1 unit# "filename"    /* Drain package file */
EVT1 unit# "filename"    /* Evapotranspiration package file */
GHB1 unit# "filename"    /* General Head Bound. package file */
STR1 unit# "filename"    /* The Stream package file */
HFB1 unit# "filename"    /* The Horiz. Flow Barrier package file */
CHD1 unit# "filename"    /* The Changing Head Boundary package file */
MT3D unit# "filename"    /* The MT3D head and flow file  */
LAY "filename"   /* Layer data file */
ORIG x y z       /* loc. of minx, miny, minz corner */
ROTZ angle       /* The angle of rotation of the grid */
DMAT id   /* The default material id */
MAT       /* Explicit definition of material ids */
id1 id2 … idn
LAYER flag const format /* Default layer thicknesses */
REFTIME bu td yr mo day hr min sec
UNITS l t m f c
```

Figure 24.1 *The* MODFLOW *Super File Format.*

```
MODSUP
IJK -y +x -z
LIST   26 "ding.out"
BAS1    1 "ding.bas"
BCF3   11 "ding.bcf"
OUT1   10 "ding.oc"
HEAD  -30 "ding.hed"
CCF   -40 "ding40.ccf"
PCG2   12 "ding.pcg"
WEL1   13 "ding.wel"
DRN1   14 "ding.drn"
RCH1   20 "ding.rch"
MT3D  -29 "ding.hff"
LAY  "ding.lyr"
ORIG  -691.175 199.653 -978
ROTZ 0
LAYER   0   1956.0000 (10G13.4)
DMAT 0
REFTIME d 0 1999 12 23 14 56 34
UNITS "m" "d" "mg" "N" "mg/l"
```

Figure 24.2 *Sample* MODFLOW *Super File.*

The card types used in the MODFLOW super file format are as follows:

Card Type	**MODSUP**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**IJK**

Description	Defines the orientation of the ijk indices. This card should be placed immediately after the MODSUP card. See IJK card described in Section 14
Required	YES

Card Type	**LIST**		
Description	Defines a unit number and file name for the ASCII solution file that will be written by MODFLOW. This card must be placed immediately after the IJK card.		
Required	YES		
Format	`LIST unit# "filename"`		
Sample	`LIST 26 "prov1.lst"`		
Field	*Variable*	*Value*	*Description*
1	`unit#`	+	The unit number associated with the file. It should be a unique integer
2	`filename`	str	The name of the listing file. When MODFLOW is executed, this file will appear in the same directory as the super file

Card Type	**BAS1**		
Description	Defines a unit number and file name for the MODFLOW Basic package. This card must be placed immediately after the LIST card.		
Required	YES		
Format	`BAS1 unit# "filename"`		
Sample	`BAS1 1 "prov1.bas"`		
Field	*Variable*	*Value*	*Description*
1	`unit#`	+	The unit number associated with the file. The unit number for the Basic package file in most cases should be 1.
2	`filename`	str	The name of the Basic package file. The file should be located in the same directory as the super file.

Card Type	**HEAD, DRAW, CCF**
Description	Defines a unit number and file name for the binary solution files that are written by MODFLOW.
Required	NO
Format	`Similar to LIST card.`

Card Type	**BCF1, BCF2, BCF3**
Description	Defines a unit number and file name for the MODFLOW block centered flow packages. Cards should appear after the BAS1 card. Only one of either BCF1, BCF2, or BCF3 cards should be used.
Required	YES
Format	`Similar to BAS1 card.`

Card Type	**OUT1, RIV1, RCH1, WEL1, DRN1, EVT1, GHB1, SIP1, PCG2, SOR1, STR1, CHD1, HFB1**
Description	Defines a unit number and file name for each of the MODFLOW packages. Cards should appear after the BAS1 card. Stress package files should appear after the BCFx card
Required	NO

Format	`Similar to BAS1 card.`

Card Type	**MT3D**
Description	Defines a unit number and file name for MT3D head and flow file.
Required	NO
Format	`Similar to BAS1 card.`

Card Type	**LAY**		
Description	Defines a file name for the layer data file. Presence or absence of the card indicates the status of the layer data entry flag. If card is present, the true layer option is active.		
Required	NO		
Format	`LAY "filename"`		
Sample	`LAY "prov1.lyr"`		
Field	*Variable*	*Value*	*Description*
1	filename	str	The name of the layer file.

Card Type	**ORIG, ROTZ, DMAT, MAT**
Description	These cards are described in Section 14.

Card Type	**LAYER**		
Description	Defines layer thicknesses. This card should come after one of the BCF cards. The layer thicknesses are not normally defined in MODFLOW. They are saved by GMS to ensure that the grid maintains the original geometry when displayed in oblique view. If this card is not defined, the layer thicknesses are defined using an average of the row and column widths.		
Required	NO		
Format	`LAYER flag const format` t_1 t_2 . . t_{nl}		
Sample	`LAYER 1 0 10F5.2` `10.0 10.0 12.0 12.0 15.0 15.0`		
Field	*Variable*	*Value*	*Description*
1	flag	0,1	Flag indicating how the thicknesses are defined. If flag=0, all thicknesses are the same and thicknesses will not be listed. If flag=1, all thickness will be listed.
2	const	±	The constant value assigned to all thicknesses if flag=0. Value is ignored if flag=1.
3	format	str	The FORTRAN format specifier for the thicknesses.
4 - (nl+3)	t	+	The thickness of each layer. Only listed if flag=1. Up to 10 values listed per line.

Card Type	**REFTIME**		
Description	Defines the reference time for the simulation.		
Required	NO		
Format	`REFTIME bu td td year mo day hr min sec`		
Sample	`REFTIME d 0 1999 12 30 15 45 23`		
Field	*Variable*	*Value*	*Description*

1	`bu`	 yr d h min s	Base unit: year day hour minute second
2	`td`	 0 1	Time display: Absolute Relative
3	`year`	+	Year
4	`mo`	1-12	Month
5	`day`	1-31	Day
6	`hr`	0-23	Hour
7	`min`	0-59	Minute
8	`sec`	0-59	Second

Card Type	**UNITS**		
Description	Defines the units for the simulation.		
Required	NO		
Format	`UNITS l t m f c`		
Sample	`UNITS "m" "d" "mg" "N" "mg/l"`		
Field	*Variable*	*Value*	*Description*
1	`l`	 ft in m cm	Length unit: feet inches meters centimeters
2	`t`	 yr d h min s	Time unit: year day hour minute second
3	`m`	 mg g kg slug	Mass unit: milligram gram kilogram slug
4	`f`	 lb N KN	Force unit: pound Newton Kilo-Newton
5	`c`	 g/l mg/l ppm ppb ppt	Concentration unit: grams/liter milligrams/liter parts per million parts per billion parts per thousand

25 MT3DMS Super Files

An MT3D super file allows all of the packages involved in a MT3D simulation to be input at once. The MT3D super file is used by both MT3D and RT3D.

The format of the MT3D super file is shown in Figure 25.1 and a sample MT3D super file is shown in Figure 25.2.. The file contains a list of the package files used in the simulation. It also contains data needed by GMS which are not contained in any of the package files such as the layering and orientation of the 3D grid.

The format of the MT3D package files is described in Zheng (1990).

```
MT3DSUP   /* File type identifier */
BTN "filename"   /* The Basic Transport package file */
ADV "filename"   /* The Advection package file */
DSP "filename"   /* The Dispersion package file */
SSM "filename"   /* The Source/Sink Mixing package file */
RCT "filename"   /* The Chemical Reaction package file */
FLO "filename"   /* The MODFLOW flow file */
CHK flag /* Flag for output of heads & flows (T/F) */
OUT "filename"   /* The standard ouput file */
CON "filename"   /* The concentration solution file */
DSS "filename"   /* Data set super file */
MAS "filename"   /* The mass balance file */
SPC "name" id mflag     /* Species flag */
STRSS flag       /* River vs. stream package flag */
ORIG x y z       /* The loc. of corner with min x, y, and z */
ROTZ angle       /* The angle of rotation */
DMAT id   /* The default material id */
MAT       /* Explicit definition of material ids */
id1 id2 … idn
LAYER flag const format /* Layer thicknesses */
t1 t2 . . tn     /* List of thicknesses */
DMAT n
HTOPEQ1 flag
UNITS l t m f c
```

Figure 25.1 *The* MT3D *Super File Format.*

```
MT3DSUP
BTN "model1.btn"
FLO "ding.hff"
ADV "model1.adv"
DSP "model1.dsp"
SSM "model1.ssm"
CHK Y
OUT "model1.out"
CON "model1.con"
MAS "model1 mas"
SPC "tce" 1 1
SPC "pce" 2 1
ORIG     -6.91175000000000e+02 1.99653000000000e+02 -9.78000000000000e+02
ROTZ 0
LAYER    0
   1956.0000
DMAT 0
HTOPEQ1 1
UNITS "m" "d" "kg" "N" "ppt"
```

Figure 25.2 *Sample* MT3D *Super File.*

The card types used in the MT3D super file format are as follows:

Card Type	**MT3DSUP, RT3DSUP**
Description	File type identifier. Also signifies which model is being used. Must be on first line of file. No fields.
Required	YES

Card Type	**BTN**		
Description	Defines a file name for the MT3D basic transport package file		
Required	YES		
Format	`BTN "filename"`		
Sample	`BTN "prob1.btn"`		
Field	*Variable*	*Value*	*Description*
1	`filename`	str	The name of the basic transport package file. The file should be located in the same directory as the super file.

Card Type	**ADV, DSP, SSM, RCT**		
Description	Defines a file name for the MT3D package files: advection, dispersion, source and sink mixing, and chemical reaction packages. Only those cards appropriate for the current simulation are required. The packages to be used are specified in the packages dialog in the Basic Transport Package dialog.		
Required	NO		
Format	`ADV "filename"`		
Sample	`ADV "prob1.adv"`		
Field	*Variable*	*Value*	*Description*
1	`filename`	str	The name of the package file. The file should be located in the same directory as the super file.

Card Type	**FLO**
Description	Defines a file name for the head and flow file which has been computed from a previous MODFLOW simulation.
Required	YES
Format	`Similar to BTN card.`

Card Type	**CHK**		
Description	Flag for output of heads and flow terms to the ASCII listing file		
Required	YES		
Format	`CHK flag`		
Sample	`CHK Y`		
Field	*Variable*	*Value*	*Description*
1	`flag`	Y/N	Yes or No depending on whether head and flow terms will be printed to the ASCII listing file

Card Type	**OUT**
Description	Defines a file name for the MT3D ASCII listing file. Results from the MT3D simulation will be printed to this file as per the specifications in the MT3D output control dialog.
Required	YES

Format	`Similar to BTN card.`

Card Type	**CON**
Description	Defines a default prefix for the MT3DMS solution concentration file. The concentrations for each specie are saved to a separate file. Each file is a GMS binary data set file.
Required	YES
Format	`Similar to BTN card.`

Card Type	**DSS**
Description	Defines a file name for the data set super file name. This file contains the names of all of the concentration solution files.
Required	YES
Format	`Similar to BTN card.`

Card Type	**MAS**
Description	Defines a file name for the mass balance file. The Save mass balance file option must have been selected in the MT3D output control dialog to use this option.
Required	NO
Format	`Similar to BTN card.`

Card Type	**STRSS**		
Description	Flag for identifying whether the stream or river package is being used.		
Required	NO. If card is not present, River package is assumed.		
Format	`STRSS flag`		
Sample	`STRSS 0`		
Field	*Variable*	*Value*	*Description*
1	`flag`	0 1	River package is being used Stream package is being used

Card Type	**SPC**		
Description	Species card.		
Required	YES		
Format	`SPC "name" id mflag`		
Sample	`SPC "tce" 1 1`		
Field	*Variable*	*Value*	*Description*
1	`name`	str	The name of the specie
2	`id`	+	The id of the specie
3	`mflag`	0 1	The specie is immobile The specie is mobile

Card Type	**RXNPARAM**		
Description	RT3D reaction parameter card.		
Required	YES (for RT3D)		
Format	`RXNPARAM "name" id cflag`		
Sample	`RXNPARAM "Kr" 2 0`		
Field	*Variable*	*Value*	*Description*
1	`name`	str	The name of the reaction parameter
2	`id`	+	The id of the reaction parameter

3	`mflag`	0 1	The parameter is constant The parameter is spatially variable

Card Type	**ORIG, ROTZ, DMAT, MAT**		
Description	These cards are described in Section 14.		

Card Type	**LAYER, UNITS**		
Description	These cards are described in Section 24.		

Card Type	**HTOPEQ1**		
Description	Flag for identifying whether the HTOP array should be set equal to the top of layer 1 or defined independently.		
Required	NO		
Format	`HTOPEQ1 flag`		
Sample	`HTOPEQ1 0`		
Field	*Variable*	*Value*	*Description*
1	`flag`	0 1	HTOP is NOT equal to top of layer 1. HTOP is equal to top of layer 1.

26 RT3D Super Files

The RT3D super file format is very similar to the MT3DMS super file format. The first line in the file is RT3DSUP. The file also contains the RXNPARAM card described in Section 20.7.

27 SEAM3D Super Files

The SEAM3D super file format is also very similar to the MT3DMS super file format. The first line in the file is SEAM3DSUP. The file also contains the S3DSPEC, S3DMCOL, S3DPROD cards described in Section 20.2.

28 SEEP2D Super Files

The format of the SEEP2D super file is shown in Figure 28.1and a sample SEEP2D super file is shown in Figure 28.2. The file primarily contains the names of the input and output files used in the simulation.

```
SEEPSUPER
SEEP     "filename"     /* Main SEEP2D input file */
ODAT     "filename"     /* Text output file */
OGEO     "filename"     /* Output geometry file */
DSET     "filename"     /* Data set solution file */
BASEMAT id       /* Base material for generating flow lines */
```

Figure 28.1 SEEP2D Super File Format.

```
SEEPSUPER
SEEP     sampcon.s2d
ODAT     sampcon.out
OGEO     sampcon.geo
DSET     sampcon.sol
BASEMAT 2
```

Figure 28.2 Sample SEEP2D Super File.

29 Image File

Image files are used in conjunction with TIFF files which have been previously imported to GMS and registered. They include the name of the TIFF file, the registration points, and the bounds of the clipping window. The format of the image file is shown in Figure 18.50 and a sample image file is shown in Figure 18.51.

```
IMAGE    /* File type identifier */
TIFF “filename”  /* Indicates the name of the tiff file used */
IMREGPTS
PT1 u1 v1 x1 y1
PT2 u2 v2 x2 y2
PT3 u2 v2 x2 y2
CLIPPOINT
x1 x2
y1 y2
```

Figure 29.1 The Image File Format.

```
IMAGE
TIFF "easttex.tif"
IMREGPTS
PT1 117 797 0.000000 10000.000000
PT2 117 88 0.000000 0.000000
PT3 1053 88 13220.000000 0.000000
CLIPPOINT
-1082.059503 13885.402536
-992.568818 8457.158566
```

Figure 29.2 Sample Image File.

The card types used in the Image file format are as follows:

Card Type	**IMAGE**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**TIFF**		
Description	Defines the name of the TIFF file to be displayed as an image.		
Required	YES		
Format	`TIFF "filename"`		
Sample	`TIFF "easttex.tif"`		
Field	*Variable*	*Value*	*Description*
1	`filename`	str	The name of the TIFF file.

Card Type	**PT1, PT2, PT3**		
Description	The three registration points used to define locations on a given image.		
Required	YES		
Format	PT1 tx_1 ty_1 wx_1 wy_1 PT2 tx_2 ty_2 wx_2 wy_2 PT3 tx_3 ty_3 wx_3 wy_3		
Sample	PT1 117 797 0.000000 10000.000000 PT2 117 88 0.000000 0.000000 PT3 1053 88 13220.0 0.000000		
Field	*Variable*	*Value*	*Description*
1-2 3-4	tx ty wx wy	± ±	Texture map coordinates. World coordinates.

Card Type	**CLIPPOINTS**		
Description	Defines the coordinates of the area in the TIFF file to be displayed as the image. (The area clipped and displayed from the TIFF file.		
Required	YES		
Format	CLIPPOINTS xmin xmax ymin ymax		
Sample	CLIPPOINTS -628.990382 14338.471657 -857.665608 8354.617436		
Field	*Variable*	*Value*	*Description*
1-2 3-4	xmin xmax ymin ymax	± ±	Min and max values in the x direction. Min and max values in the y direction.

30 GMS Pathline File

GMS includes a suite of tools for visualizing pathlines resulting from particle tracking. In addition to the MODPATH file formats, a generic GMS pathline file format is supported. This file format is a simple text format designed for users who wish to generate their own pathlines or import their pathlines from another application. The file format is shown in Figure 18.52 and a sample file is shown in Figure 18.53. The first part of the file is a set of headers defining the name of the pathline set, the number of independent paths, and the number and names of a series of optional data sets. The paths are then listed by listing the path id, coordinates, time, and data values. When listing the points on the paths, the paths can be mixed if desired, i.e., it is not necessary to complete one path before defining another path.

```
PATHLINE /* File type identifier */
NAME "name"      /* Pathline set name */
NUMPATHS n       /* The number of paths */
NUMDS nds /* Number of data sets */
DSNAMES "name1" "name2" etc.    /* Data set names */
PATHS    /* Marks beginning of paths */
i x y z t ds1 ds2 ... dsn       /* Point defining a path */
i x y z t ds1 ds2 ... dsn
.
.
.
(Repeat path pt records as many times as necessary.  The points from
different pathlines can be intermixed.)
```

Figure 30.1 *The* GMS *Pathline File Format.*

```
PATHLINE
NAME "sample pathlines"
NUMPATHS 2
NUMDS 2
DSNAMES "concentration" "velocity"
PATHS
1  27.4  34.5  10.0  0.0  100.0  0.01112
1  29.6  32.1  11.1  2.0  111.0  0.01234
1  30.4  29.7  12.3  4.0  109.0  0.01534
1  35.7  24.0  15.2  6.0   99.0  0.01023
2  78.2  11.4  50.1  0.0   55.0  0.00456
2  65.6  14.4  52.3  2.0   67.0  0.00464
2  60.2  17.4  55.5  4.0   34.0  0.00673
2  55.7  21.0  57.3  6.0   12.0  0.00981
```

Figure 30.2 Sample GMS *Pathline File.*

The cards used in the GMS pathline file format are as follows:

Card Type	**PATHLINE**
Description	File type identifier. Must be on first line of file. No fields.
Required	YES

Card Type	**NAME**		
Description	Defines the name of the pathline set.		
Required	YES		
Format	`NAME "name"`		
Sample	`NAME "path set #1"`		
Field	*Variable*	*Value*	*Description*
1	name	str	The name of the pathline set.

Card Type	**NUMPATHS**		
Description	Defines the number of distinct paths in the pathline set.		
Required	YES		
Format	`NUMPATHS "n"`		
Sample	`NUMPATHS "32"`		
Field	*Variable*	*Value*	*Description*
1	n	+	The number of paths.

Card Type	**NUMDS**		
Description	Defines the number of data sets associated with the pathlines.		
Required	YES		
Format	`NUMDS "nds"`		
Sample	`NUMDS "2"`		
Field	*Variable*	*Value*	*Description*
1	nds	+	The number of data sets.

Card Type	**DSNAMES**		
Description	Defines the names of the data sets associated with the pathlines.		
Required	YES		
Format	`DSNAMES "name1" "name2" …`		
Sample	`DSNAMES "concentration" "temp"`		
Field	*Variable*	*Value*	*Description*
1-nds	name	str	The names of the data sets.

Card Type	**PATHS**		
Description	Defines the points along the pathlines. Use as many lines as necessary.		
Required	YES		

Format	`PATHS`
	`i x y z t ds1 ds2 ... dsn`
	`i x y z t ds1 ds2 ... dsn`
	`.`
	`.`		

Sample	`PATHS`
	`1  27.4  34.5  10.0  0.0  100.0  0.01112`
	`1  29.6  32.1  11.1  2.0  111.0  0.01234`
	`.`
	`.`		

Field	*Variable*	*Value*	*Description*
1	i	+	The id of the path.
2-4	x,y,z	±	The coordinates of the point.
5	t	±	The time corresponding to the point
6-?	ds*	±	The values of the data sets.

Index