MODFLOW 6 – Description of Input and Output

MODFLOW 6 Development Team

Version 6.0.00—August 10, 2017

Contents

List of Figures

List of Tables

Introduction

MODFLOW 6 is a command line executable program that reads input from ASCII text files, and optionally from binary files. MODFLOW 6 writes simulation output to ASCII text and binary files. MODFLOW 6 itself, like its predecessors, does not provide any graphical output, though users may decide to adopt a Graphical User Interface (GUI) for preparing model input and visualizing model output. This document provides details on the format of the input files and the format of the output files. Details on the numerical methods and the underlying theory for MODFLOW 6 are described in separate reports (Hughes and others, 2017; Langevin and others, 2017; Provost and others, 2017). Instructions for preparing the input or visualizing the output is beyond the scope of this report.

Running a Simulation

MODFLOW 6 is run from the command line by entering the name of the MODFLOW 6 executable program. If the run is successful, it will conclude with a statement about normal termination.

```
                        MODFLOW Version 6
            U.S. GEOLOGICAL SURVEY MODULAR HYDROLOGIC MODEL
                      Version 0.9.00 04/07/2017

 Run start date and time (yyyy/mm/dd hh:mm:ss): 2017/03/31 14:10:29


 Writing simulation list file: mfsim.lst
 Using Simulation name file: mfsim.nam
 Solving:  Stress period:     1    Time step:     1
 Run end date and time (yyyy/mm/dd hh:mm:ss): 2017/03/31 14:10:30
 Elapsed run time:  0.655 Seconds


 Normal termination of simulation.
```

MODFLOW 6 requires that a simulation name file (described in a subsequent section titled "Simulation Name File") be present in the working directory. This simulation name file must be named "mfsim.nam". If the mfsim.nam file is not located in the present working directory, then MODFLOW 6 will terminate with the following error.

```
                        MODFLOW Version 6
            U.S. GEOLOGICAL SURVEY MODULAR HYDROLOGIC MODEL
                      Version 0.9.00 04/07/2017

 Run start date and time (yyyy/mm/dd hh:mm:ss): 2017/03/31 14:15:27

 Writing simulation list file: mfsim.lst

ERROR REPORT:

 *** ERROR OPENING FILE "mfsim.nam" ON UNIT 1001
      SPECIFIED FILE STATUS: OLD
      SPECIFIED FILE FORMAT: FORMATTED
      SPECIFIED FILE ACCESS: SEQUENTIAL
      SPECIFIED FILE ACTION: READ
        IOSTAT ERROR NUMBER: 29
  -- STOP EXECUTION (openfile)
 Stopping due to error(s)
```

During execution MODFLOW 6 creates a simulation output file, called a listing file, with the name "mfsim.lst". This file contains general simulation information, including information about exchanges between models, timing, and solver progress. Separate listing files are also written for each individual model. These listing files contains the details for the specific models.

In the event that MODFLOW 6 encounters an error, the error message is written to the command line window as well as to the simulation listing file. The error message will also contain the name of the file that was being read when the error occurred, if possible. This information can be used to diagnose potential causes of the error.

Form of Input Instructions

MODFLOW 6 differs from its predecessors in the form of the input. Whereas previous MODFLOW versions read numerical values, arrays, and lists in a highly structured form, MODFLOW 6 reads information in the form of blocks and keywords. MODFLOW 6 also reads arrays and lists of information, but these arrays and lists are tagged with identifying block names or keywords. MODFLOW 6 will terminate with an error if it detects an unrecognized block or keyword.

Block and Keyword Input

Input to MODFLOW 6 is provided within blocks. A block is a section of an ASCII input file that begins with a line that has "BEGIN" followed by the name of the block and ends with a line the begins with "END" followed by the name of the block. MODFLOW 6 will terminate with an error if blocks do not begin and end with the same name, or if a "BEGIN" or "END" line is missing. Information within a block differs depending on the part of MODFLOW 6 that reads the block. In general, keywords are used within blocks to turn options on or specify the type of information that follows the keyword. If an unrecognized keyword is encountered in a block, MODFLOW 6 will terminate with an error.

The keyword approach is adopted in MODFLOW 6 to improve readability of the MODFLOW 6 input files, enhance discovery of errors in input files, and improve support for backward compatibility by allowing the program to expand in functionality while allowing previously developed models to be run with newer versions of the program.

Within these user instructions, keywords are shown in capital letters to differentiate them from other input that is provided by the user. For example, "BEGIN" and "END" are recognized by MODFLOW 6 , and so they are capitalized. Also, line indentation is used within these user instructions to help with readability of the blocks. Typically, lines within a block are indented two spaces to accentuate that the lines are part of the block. This indentation is not enforced by the program, but users are encouraged to use it within their own input files to improve readability.

Unless stated otherwise in this user guide, information contained within a block can be listed in any order. If the same keyword is provided more than once, then the program will use the last information provided by that keyword.

Comment lines and blanks lines are also allowed within most blocks and within most input files. Valid comment characters include "#" "!", and "//". Comments can also be placed at the end of some input lines, after the required information. Comments are not allowed at the end of some lines if the program is required to read an arbitrary number of non-keyword items. Comments included at the end of the line must be separated from the rest of the line by at least one space.

Unless otherwise noted in the input instructions, multiple blocks of the same name cannot be specified in a single input file. The block order within the input file must follow the order presented in the input instructions. Each input file typically begins with an OPTIONS block, which is generally not required, followed by one or more data blocks.

The following is an example of how the input instructions for a block are presented in this document.

```
BEGIN OPTIONS
  [AUXILIARY <auxiliary(naux)>]
  [PRINT_INPUT]
  [MAXIMUM_ITERATION <maxsfrit>]
END OPTIONS
```

This example shows the items that may be specified with this OPTIONS block. Optional items are enclosed between "[" and "]" symbols, and these optional items can be nested as shown with the "[AUXILIARY <auxiliary(naux)>]" item. The "<" and ">" symbols indicate a variable that must be provided by the user. In this case, `auxiliary` is an array of size `naux`. Because there are bracket symbols around the entire item, the user it not required to specify anything for this item. Likewise, the user may or may not invoke the "PRINT_INPUT" option. Lastly, the user can specify "MAXIMUM_ITERATION" followed by a numeric value for "maxsfrit". If the user does not specify an optional item, then a default condition will apply. Behavior of the default condition is described in the input instructions for that item.

A valid user input block for OPTIONS might be:

```
#This is my options block
BEGIN OPTIONS
  AUXILIARY temperature salinity
  MAXIMUM_ITERATION 10
END OPTIONS
```

The following is another valid user input block for OPTIONS:

```
#This is an alternative options block
BEGIN OPTIONS
  # Assign two auxiliary variables
  AUXILIARY temperature salinity
  # Specify the maximum iteration
  MAXIMUM_ITERATION 10
  #specify the print input option
  PRINT_INPUT
END OPTIONS
#done with the options block
```

Specification of Block Information in OPEN/CLOSE File

For most blocks, information can be read from a separate text file. In this case, all of the information for the block must reside in the text file. The file name is specified using the OPEN/CLOSE keyword as the first and only entry in the block as follows:

```
#This is an alternative options block
BEGIN OPTIONS
  OPEN/CLOSE myoptblock.txt
END OPTIONS
```

When MODFLOW encounters the OPEN/CLOSE keyword, the program opens the specified file on unit 99 and continues processing the information in the file as if it were within the block itself. When the program reaches the end of the file, the file is closed, and the program returns to reading the original package file. The next line after the OPEN/CLOSE line must end the block.

Some blocks do not support the OPEN/CLOSE capability. A list of all of the blocks, organized by component and input file type, are listed in a table in appendix A. This table also indicates the blocks that do not support the OPEN/CLOSE capability.

File Name Input

Some blocks may require that a file name be entered. Although spaces within a file name are not generally recommended, they can be specified if the entire file name is enclosed within single quotes, which means that the file name itself cannot have a single quote within it. On Windows computers, file names are not case sensitive, and thus, "model.dis" can be referenced within the input files as "MODEL.DIS". On some other operating systems, however, file names are case sensitive and the case used in the input instructions must exactly reflect the case used to name the file.

Lengths of Character Variables

Character variables, which are used to store names of models, packages, observations and other objects, are limited in the number of characters that can be used. Table 1 lists the limit used for each type of character variable.

Table 1: Character variable maximum sizes

Size limit name	Size	Variable(s) affected
LENAUXNAME	16	Auxiliary variable names
LENBOUNDNAME	40	Boundary names
LENMODELNAME	16	Model names
LENOBSNAME	40	Observation names
LENPACKAGENAME	16	Package names
LENSOLUTIONNAME	16	Solution names
LENTIMESERIESNAME	24	Time-series and time-array-series names

Simulation Name File

The simulation name file is read from a file in the current working directory with the name "mf-sim.nam". Input within the simulation name file is provided through the following input blocks, which must be listed in the order shown below. The options block itself is optional. All other blocks are required.

Structure of Blocks

```
BEGIN OPTIONS
  [CONTINUE]
  [NOCHECK]
  [MEMORY_PRINT_OPTION <memory_print_option>]
END OPTIONS


BEGIN TIMING
  TDIS6 <tdis6>
END TIMING


BEGIN MODELS
  <mtype> <mfname> <mname>
  <mtype> <mfname> <mname>
  ...
END MODELS


BEGIN EXCHANGES
  <exgtype> <exgfile> <exgmnamea> <exgmnameb>
  <exgtype> <exgfile> <exgmnamea> <exgmnameb>
  ...
END EXCHANGES


BEGIN SOLUTIONGROUP <group_num>
  [MXITER <mxiter>]
  <slntype> <slnfname> <slnmnames(nslnmod)>
  <slntype> <slnfname> <slnmnames(nslnmod)>
  ...
END SOLUTIONGROUP
```

Explanation of Variables

CONTINUE—keyword flag to indicate that the simulation should continue even if one or more solutions do not converge.

NOCHECK—keyword flag to indicate that the model input check routines should not be called prior to each time step. Checks are performed by default.

memory_print_option—is a flag that controls printing of detailed memory manager usage to the end of the simulation list file. NONE means do not print detailed information. SUMMARY means print only the total memory for each simulation component. ALL means print information for each variable stored in the memory manager. NONE is default if memory_print_option is not specified.

tdis6—is the name of the Temporal Discretization (TDIS) Input File.

mtype—is the type of model to add to simulation.

mfname—is the file name of the model name file.

mname—is the user-assigned name of the model. The model name cannot exceed 20 characters. If the model name has spaces within it, then it must be enclosed in double quotation marks. The model name is case insensitive; any lowercase letters are converted and stored as upper case letters.

exgtype—is the exchange type.

exgfile—is the input file for the exchange.

exgmnamea—is the name of the first model that is part of this exchange.

exgmnameb—is the name of the second model that is part of this exchange.

group_num—is the group number of the solution group

mxiter—is the maximum number of outer iterations for this solution group. The default value is 1.

slntype—is the type of solution. The Integrated Model Solution (IMS6) is the only supported option in this version.

slnfname—name of file containing solution input.

slnmnames—is the array of model names to add to this solution.

Table 2: Model types available in Version 6.0.00

Mtype	Type of Model
GWF6	Groundwater Flow Model

Table 3: Exchange types available in Version 6.0.00

Exgtype	Type of Exchange
GWF6-GWF6	Exchange between two Groundwater Flow Models

Example Input File

```
# This block is optional
BEGIN OPTIONS
END OPTIONS

# Simulation timing information
BEGIN TIMING
  TDIS6 simulation.tdis
END TIMING

# List of models in the simulation
```

```
BEGIN MODELS
  #modeltype      namefile      modelname
        GWF6    model1.nam   GWF_Model_1
        GWF6    model2.nam   GWF_Model_2
END MODELS

# List of exchanges in the simulation
BEGIN EXCHANGES
  GWF6-GWF6 simulation.exg GWF_Model_1 GWF_Model_2
END EXCHANGES

# Models are part of the same numerical solution
BEGIN SOLUTIONGROUP 1
  IMS6 simulation.ims GWF_Model_1 GWF_Model_2
END SOLUTIONGROUP
```

Temporal Discretization (TDIS) Package

Timing for all models of the simulation is controlled by the Temporal Discretization (TDIS) Package. Input to the TDIS Package is read from the filename specified for TDIS in the TIMING input block of the simulation name file.

Structure of Blocks

```
BEGIN OPTIONS
  [TIME_UNITS <time_units>]
  [START_DATE_TIME <start_date_time>]
END OPTIONS

BEGIN DIMENSIONS
  NPER <nper>
END DIMENSIONS

BEGIN PERIODDATA
  <perlen> <nstp> <tsmult>
  <perlen> <nstp> <tsmult>
  ...
END PERIODDATA
```

Explanation of Variables

time_units—is the time units of the simulation. This is a text string that is used as a label within model output files. Values for time_units may be "unknown", "seconds", "minutes", "hours", "days", or "years". The default time unit is "unknown".

start_date_time—is the starting date and time of the simulation. This is a text string that is used as a label within the simulation list file. The value has no affect on the simulation. The recommended format for the starting date and time is described at https://www.w3.org/TR/NOTE-datetime.

nper—is the number of stress periods for the simulation.

perlen—is the length of a stress period.

nstp—is the number of time steps in a stress period.

tsmult—is the multiplier for the length of successive time steps. The length of a time step is calculated by multiplying the length of the previous time step by TSMULT. The length of the first time step, Δt_1, is related to PERLEN, NSTP, and TSMULT by the relation $\Delta t_1 = perlen \frac{tsmult-1}{tsmult^{nstp}-1}$.

Example Input File

```
# Comment for this TDIS input file

BEGIN OPTIONS
  TIME_UNITS DAYS
END OPTIONS
```

```
BEGIN DIMENSIONS
  NPER 2
END DIMENSIONS

BEGIN PERIODDATA
   365.00 1 1.0  Items: PERLEN NSTP TSMULT
   365.00 10 1.2 Items: PERLEN NSTP TSMULT
END PERIODDATA
```

Groundwater Flow (GWF) Model Input

This section describes the data files for a MODFLOW 6 Groundwater Flow (GWF) Model. A GWF Model is added to the simulation by including a GWF entry in the MODELS block of the simulation name file.

There are three types of spatial discretization approaches that can be used with the GWF Model. Input for a GWF Model may be entered in a structured form, like for previous MODFLOW versions, in that users specify cells using their layer, row, and column indices. Users may also work with a layered grid in which cells are defined using vertices. In this case, users specify cells using the layer number and the cell number. Lastly, GWF Models may be entered as fully unstructured models, in which cells are specified using only their cell number. Once a spatial discretization approach has been selected, then all input with cell indices must be entered accordingly.

The GWF Model is designed to permit input to be gathered, as it is needed, from many different files. Likewise, results from the model calculations can be written to a number of output files. The GWF Model Listing File is a key file to which the GWF model output is written. As MODFLOW 6 runs, information about the GWF Model is written to the GWF Model Listing File, including much of the input data (as a record of the simulation) and calculated results. Details about the files used by each package are provided in this section on the GWF Model Instructions.

MODFLOW 6 is further designed to allow the user to control the amount, type, and frequency of information to be output. Much of the output will be written to the Simulation and GWF Model Listing Files, but some model output can be written to other files. The Listing Files can become very large for common models. Text editors are useful for examining the Listing File. The GWF Model Listing File includes a summary of the input data read for all packages. In addition, the GWF Model Listing File optionally contains calculated head controlled by time step, and the overall volumetric budget controlled by time step. The Listing Files also contain information about solver convergence and error messages. Output to other files can include head and cell-by-cell flow terms for use in calculations external to the model or in user-supplied applications such as plotting programs.

The GWF Model reads a file called the Name File, which specifies most of the files that will be used in a simulation. Several files are always required whereas other files are optional depending on the simulation. The Output Control Package receives instructions from the user to control the amount and frequency of output. Details about the Name File and the Output Control Package are described in this section.

Information for Existing MODFLOW Users

MODFLOW 6 contains most of the functionality of MODFLOW-2005, MODFLOW-NWT, MODFLOW-USG, and MODFLOW-LGR. To the existing MODFLOW user, however, MODFLOW 6 will feel different from previous MODFLOW versions. Some packages have been divided, renamed, or removed, and some capabilities, which previously caused confusion or were implemented due to computer memory limitations, are no longer supported (for example, "quasi-3d confining units" are not supported in the GWF Model). The form of the input files for MODFLOW 6 is different from previous MODFLOW versions in that input files are now divided into blocks, and keywords are used to specify options and input variables. Extensive testing was used as part of the development process to ensure that MODFLOW 6 simulation results are identical to the results from previous MODFLOW versions. In some cases, it was not possible to exactly replicate the simulation results from previous MODFLOW versions. In those cases, the differences could be explained by an option that is no longer supported, or because of slight differences in the underlying formulation.

The following list, which is repeated from Langevin and others (2017), summarizes the major differences between the GWF Model in MODFLOW 6 and previous versions of MODFLOW. This list is intended for those with a general understanding of the capabilities in previous versions of MODFLOW.

1. The GWF Model in MODFLOW 6 supports three alternative input packages for specifying the grid used to discretize the groundwater system.

   - The Discretization (DIS) Package defines a grid based on layers, rows, and columns. In this report, this type of grid is referred to as a "regular MODFLOW grid" because it corresponds to traditional MODFLOW grids. An interior cell in a regular MODFLOW grid is connected to four adjacent cells in the same layer, to one overlying cell, and to one underlying cell.
   - The Discretization by Vertices (DISV) Package defines a grid using a list of (x, y) vertex pairs and the number of layers. A list of vertices is provided by the user to define a two-dimensional horizontal grid in plan view. This list of vertices may define a regular MODFLOW grid, or they may define more complex grids, such as grids consisting of triangles, hexagons, or Voronoi polygons, for example. This same two-dimensional horizontal grid applies to each layer in the model. Cells defined using the DISV Package are referenced by layer number and by the cell number within the horizontal grid. Within a layer, a cell may be horizontally connected to any number of surrounding cells in that layer. In the vertical direction a cell can be connected to only one overlying cell and only one underlying cell. Grids defined with the DISV Package are considered to be unstructured.
   - The unstructured Discretization (DISU) Package is the most flexible of the three packages and is patterned after the unstructured grid implemented in MODFLOW-USG. For each cell, the user specifies a list of connected cells and the connection properties. When the DISU Package is used, cells are referenced only by their cell number; unlike the MODFLOW-USG approach, there is no concept of a layer in the DISU Package in MODFLOW 6, but cells may still overlie or underlie one another.

2. For the layered grid types supported in the GWF Model (DIS and DISV), cells can be permanently excluded from the grid for the simulation. Input values (such as hydraulic conductivity) are still required for these excluded cells, and the program will write special codes or zero values for output, but the program does not allocate memory or store values for excluded cells during run time. In this case, the matrix equations are formulated for a reduced system in which only the included cells are numbered. Users can also mark excluded cells as "vertical pass-through cells." When these vertical pass-through cells are encountered, the program connects the cells overlying and underlying the pass-through cell. This capability allows "pinched" cells to be removed from the solution. These options to exclude cells or exclude them as pass-through cells are available for the DIS and DISV Packages through specification of the IDOMAIN array; the IDOMAIN capability is not available for the DISU Package.

3. There is no longer a Basic Package input file. Initial head values are specified using an Initial Conditions (IC) Package, and constant heads are specified using the Time Varying Specified Head (CHD) Package. Cells that are permanently excluded from the simulation can be eliminated using the IDOMAIN capability entered through the DIS or DISV Packages. For a cell that may transition from inactive ("dry") to active ("wet") during a simulation, the user can start the cell as inactive by assigning an initial head below the cell bottom.

4. The Newton-Raphson formulations and accompanying upstream weighting schemes implemented in MODFLOW-NWT and MODFLOW-USG for handling dry or nearly dry cells have been synthesized into a single formulation. The Newton-Raphson formulation in the GWF Model for

MODFLOW 6 remains an optional alternative to the standard formulation used in most previous MODFLOW versions. Much of this report is focused on systematically explaining standard and Newton-Raphson formulations for the GWF Model and its packages.

5. Information on temporal discretization, such as number of stress periods, period lengths, number of time steps, and time step multipliers, is specified at the simulation level, rather than for an individual model. This information is provided in the Timing Module, which controls the temporal discretization and applies to all models within a simulation. The Timing Module is part of the MODFLOW 6 framework and is described separately in Hughes and others (2017).

6. Aquifer properties used to calculate hydraulic conductance are specified in the Node Property Flow (NPF) Package. In MODFLOW 6, the NPF Package calculates intercell conductance values, manages cell wetting and drying, and adds Newton-Raphson terms for intercell flow expressions. The NPF Package allows individual cells to be designated as confined or convertible; this was not an option in previous MODFLOW versions as the designation was by layer. The NPF Package also has several options for simulating drainage problems and problems involving perched aquifers where an active cell overlies a partially saturated cell. The default NPF Package behavior (in which none of these options are set) is the most stable for typical groundwater problems. The default NPF Package behavior does not correspond to the default behavior for other MODFLOW internal flow packages. The NPF Package does not support quasi-3D confining units. The NPF Package replaces the Layer Property Flow (LPF), Block-Centered Flow (BCF), and Upstream Weighting (UPW) Packages from previous MODFLOW versions. Capabilities of the Hydrogeologic Unit Flow (HUF) Package (Anderman and Hill, 2000, 2003) are not supported in the GWF Model of MODFLOW 6 .

7. Aquifer storage properties are specified in the Storage (STO) Package. If the STO Package is excluded for a model, then the model represents steady-state conditions. If the STO Package is included, users can specify steady-state or transient conditions by stress period as needed. Compressible storage contributions are no longer approximated as zero for unconfined layers; contributions from pore drainage and compressible storage are separated in the model output.

8. The Horizontal Flow Barrier (HFB) Package (Hsieh and Freckleton, 1993; Harbaugh, 2005) in MODFLOW 6 allows barrier properties and locations to change by stress period. The capability to change barriers by stress period was not supported in previous MODFLOW versions.

9. The GWF Model in MODFLOW 6 allows multiple stress packages of the same type to be specified for a single GWF Model. This capability is also available in MODFLOW-CDSS (Banta, 2011). Package entries written to the budget file and budget terms in the listing file are written separately for each package.

10. Input of boundary conditions for simulation in multiple stress periods is entered differently than for previous MODFLOW versions. Boundary conditions are specified for a stress period in a "PERIOD" block. These boundary conditions remain active at their specified values until a subsequent "PERIOD" block is encountered or the end of the simulation is reached. Individual entries within the "PERIOD" block can be specified as a time-series entry. Values for these variables, which may correspond to a well pumping rate or a drain conductance, for example, are interpolated from a time-series dataset, for each time step, using several different interpolation options.

11. The Flow and Head Boundary (FHB) Package (Leake and Lilly, 1997; Harbaugh, 2005) is not supported in MODFLOW 6 ; however, its capabilities can be replicated using the WEL Package, the CHD Package, and the new time-series capability.

12. There is one Evapotranspiration (EVT) Package for MODFLOW 6 . The MODFLOW 6 EVT Package contains the functionality of the MODFLOW-2005 EVT Package, the Segmented Evapotranspiration (ETS) Package (Banta, 2000), and the Riparian Evapotranspiration (RIP-ET) Package (Maddock and others, 2012).

13. A new Multi-Aquifer Well (MAW) Package replaces the Multi-Node Well (MNW1 and MNW2) Packages (Halford and Hanson, 2002; Konikow and others, 2009). The new package does not contain all of the options available in MNW1 and MNW2, but it does contain the most commonly used ones. It also has new capabilities for simulating flowing wells. The MAW Package is solved as part of the matrix solution and is tightly coupled with the GWF Model. This tight coupling with the GWF Model may substantially improve convergence for simulations of groundwater flow to multi-aquifer wells.

14. Most capabilities of the Stream (STR) and Streamflow Routing (SFR) Packages (Prudic, 1989; Prudic and others, 2004; Niswonger and Prudic, 2005) are included in MODFLOW 6 as a new SFR Package. The new SFR Package contains all of the functionality of the SFR Package in MODFLOW-2005 with the following exceptions: (a) the concept of a "segment" has been eliminated, (b) only rectangular cross sections are supported for stream reaches, and (c) unsaturated zone flow beneath stream reaches cannot be simulated.

15. A new Lake (LAK) Package replaces the existing MODFLOW Lake Packages (Merritt and Konikow, 2000). In addition to being able to represent lakes that are incised into the model grid, the new LAK Package can also represent sub-grid scale lakes that are conceptualized as being on top of the model. The new package contains most of the capabilities available in previous LAK Packages. The LAK Package documented here does not represent unsaturated zone flow beneath a lake or support for the coalescing lake option described in Merritt and Konikow (2000).

16. A new Unsaturated Zone Flow (UZF) Package, based on the one described by Niswonger and others (2006), is included in the GWF Model of MODFLOW 6. The new UZF Package allows the UZF capabilities to be applied to only selected cells of the GWF model. The new UZF Package also supports a multi-layer option, which allows for vertical heterogeneity in unsaturated zone properties.

17. A new Water Mover (MVR) Package is included in MODFLOW 6. The MVR Package can be used to transfer water from individual "provider" features of selected packages (WEL, DRN, RIV, GHB, MAW, SFR, LAK, and UZF) to individual "receiver" features of the advanced packages (MAW, SFR, LAK, and UZF). Simple rules are used to determine how much of the available water is moved from the provider to the receiver, which allows management controls to be represented.

18. MODFLOW 6 contains a flexible new Observation (OBS) capability, which allows the user to define many different types of continuous-in-time or point-in-time observations. The new OBS capability replaces the Observation Process (Hill and others, 2000), the Gage Package, and the HYDMOD capability (Hanson and Leake, 1999) in previous MODFLOW versions. Flow, head, and drawdown observations can be obtained for the GWF Model. Flow and other package-specific observations, such as the head in a multi-aquifer well or lake stage, for example, can also be obtained. These observed values can be used subsequently with a parameter estimation program

or they can be used to make time-series plots of a wide range of simulated values. The new OBS capability does not support specification of field-measured observations, calculation of residuals, or interpolation within a grid, as was supported in previous versions of the MODFLOW OBS Process.

19. The GWF Model described in this report does not support the following list of packages and capabilities. Support for some of these capabilities may be added in future MODFLOW 6 versions.

- Interbed Storage Package (Leake and Prudic, 1991),
- Subsidence Package (Hoffmann and others, 2003),
- Subsidence and Aquifer-System Compaction Package for Water-Table Aquifers (Leake and Galloway, 2007),
- Drain with Return Flow Package (Banta, 2000)
- Reservoir Package (Fenske and others, 1996),
- Seawater Intrusion Package (Bakker and others, 2013),
- Surface-Water Routing Process (Hughes and others, 2012),
- Connected Linear Network Process (Panday and others, 2013),
- Parameter Value File (Harbaugh, 2005), and
- Link to the MT3DMS Contaminant Transport Model (Zheng and others, 2001).

In addition to this list of major differences, there are other differences between MODFLOW 6 and previous MODFLOW versions in terms of the input and output files and the way users interact with the program. These differences include:

1. The MODFLOW 6 program begins by reading a simulation name file. The simulation name file must be named "mfsim.nam."

2. All real variables in MODFLOW 6 are declared as double precision floating point numbers. Real variables written to binary output files are also written in double precision.

3. Unit numbers are no longer specified by the user. Unit numbers are determined automatically by MODFLOW 6 based upon user-provided file names.

4. The GWF Model name file contains a list of packages that are active for the model. Names for output files are not specified in the name file. Names for output files, such as the head and budget files are specified in the OC Package.

5. The EXTERNAL option for reading arrays and lists is no longer supported; however, the OPEN/-CLOSE option is still supported. The SFAC option for lists is no longer supported; however, many packages allow for specification of an auxiliary variable which can serve as a multiplier on a column of values in the list.

6. The CHD Package contains new flexibility. Cells can transition between constant-head cells and active cells during the simulation. This was not allowed in previous MODFLOW versions. Also, the CHD Packages no longer performs linear interpolation between a starting (shead) and ending head (ehead). Only a single head value is provided for each constant-head cell. The capability to linearly interpolate a head value for each time step within a stress period is available through the use of time series.

7. There are two different forms of input for the RCH and EVT Packages: array-based input and list-based input. For models that use DIS Package, the RCH and EVT input can be provided as arrays, which is consistent with previous MODFLOW versions. To use array input, the user must specify the READASARRAYS keyword in the options block. The READASARRAYS option can also be used for models that use the DISV Package. If the READASARRAYS option is not specified, then input to the RCH and EVT Packages is provided in list form. List-based input is the only option supported when the DISU Package is used.

   List-based input offers several advantages over the array-based input for specifying recharge and evapotranspiration. First, multiple list entries can be specified for a single cell. This makes it possible to divide a cell into multiple areas, and assign a different recharge or evapotranspiration rate for each area (perhaps based on land use or some other criteria). In this case, the user would likely specify an auxiliary variable to serve as a multiplier. This multiplier would be calculated by the user and provided in the input file as the fractional cell are for the individual recharge entries. Another advantage to using list-based input for specifying recharge is that "boundnames" can be specified. Boundnames work with the Observations capability and can be used to sum recharge or evapotranspiration rates for entries with the same boundname. A disadvantage of the list-based input is that one cannot easily assign recharge or evapotranspiration rates to the entire model without specifying a list of model cells. For this reason MODFLOW 6 also supports array-based input.

8. Calculation and reporting of drawdown for the model grid is no longer supported, as this calculation is easily performed as a postprocessing step. Calculation of drawdown is supported as an observation type by the OBS Package; drawdown is calculated as the difference between the starting head specified in the IC Package and the calculated head.

9. There are differences in the output files created by MODFLOW 6, such as:

   - A separate listing file is written for the simulation. This simulation listing file contains information about the simulation, including solver information. Separate listing files are written for each GWF Model that is part of the simulation.

   - Unformatted head files written by MODFLOW 6 are consistent with those written by previous MODFLOW versions; however, all real values are written in double precision.

   - The budget file written by the GWF Model is always written in "compact" form (as opposed to full three-dimensional arrays) and uses new method codes, which allow model and package names to be written to the file. Simulated intercell flows are always written in a compressed sparse row format, even for regular MODFLOW grids.

   - Information about the GWF Model grid is written to a separate file, called a "binary grid file" each time the model runs. The binary grid file can be used by postprocessing programs for subsequent analysis. The format of the binary grid file is described in a section on "Binary Output Files."

Array Input (READARRAY)

Some GWF Model packages require arrays of information to be provided by the user. This information is read using a generic READARRAY capability in MODFLOW 6. Within this user guide, variables that are read with READARRAY are marked accordingly, as shown in example input instructions for a DATA block.

```
BEGIN DATA
  ARRAY1
    <array1(nval)> -- READARRAY
END DATA
```

In this example, the uppercase ARRAY1 is a text string that is recognized by the program. While reading through the DATA block, the program would recognize ARRAY1, and would then use READARRAY to fill `array1` with `nval` values.

READARRAY Control Line

READARRAY works similar to the array readers in previous MODFLOW versions. It begins by reading a control line. The control line has one of three forms shown below, and is limited to a length of 999 characters.

```
1. CONSTANT <constant>
```

With CONSTANT, all values in the array are set equal to `constant`.

```
2. INTERNAL [FACTOR <factor>] [IPRN <iprn>]
```

With INTERNAL, the individual array elements will be read from the same file that contains the control line.

```
3. OPEN/CLOSE <fname> [FACTOR <factor>] [(BINARY)] [IPRN <iprn>]
```

With OPEN/CLOSE, the array will be read from the file whose name is specified by `fname`. This file will be opened just prior to reading the array and closed immediately after the array is read. A file that is read using this control line can contain only a single array.

READARRAY Variable Descriptions

<constant>—is a real number constant for real arrays and an integer constant for integer arrays. The `constant` value is assigned to the entire array.

FACTOR <factor>—are a keyword and a real number factor for real arrays and an integer factor for integer arrays. The individual elements of the array are multiplied by `factor` after they are read. If `factor` is specified as 0, then it is changed to 1.

(BINARY)—is an option that indicates the OPEN/CLOSE file contains array data in binary (unformatted) form. A binary file that can be read by MODFLOW may be created in only two ways. The first way is to use MODFLOW to create the file by saving heads in a binary file. This is commonly done when the user desires to use computed heads from one simulation as initial heads for a subsequent simulation. The other way to create a binary file is to write a special program that generates a binary file. If the program is written in Fortran, then the program should be compiled using a Fortran compiler that is compatible with the compiler used to compile MODFLOW. "(BINARY)" can be specified only when using U2DDBL or U2DINT, and only when the control line is OPEN/CLOSE.

IPRN <iprn>—are a keyword and a flag that indicates whether the array being read should be written to the Listing File after the array has been read and a code for indicating the format that should be used when the array is written. The format codes are the same as for MODFLOW-2005. IPRN is set to zero when the specified value exceeds those defined. If IPRN is less than zero or if the keyword and flag are omitted, the array will not be printed.

Table 4: IPRN Code and corresponding print formats for array readers. These print codes determine how the user-provided array is written to the list file

IPRN	Real	Integer
0	10G11.4	10I11
1	11G10.3	60I1
2	9G13.6	40I2
3	15F7.1	30I3
4	15F7.2	25I4
5	15F7.3	20I5
6	15F7.4	10I11
7	20F5.0	25I2
8	20F5.1	15I4
9	20F5.2	10I6
10	20F5.3	
11	20F5.4	
12	10G11.4	
13	10F6.0	
14	10F6.1	
15	10F6.2	
16	10F6.3	
17	10F6.4	
18	10F6.5	
19	5G12.5	
20	6G11.4	
21	7G9.2	

READARRAY Examples

The following examples use free-format control lines for reading an array. The example array is a real array consisting of 4 rows with 7 columns per row:

```
CONSTANT 5.7     This sets an entire array to the value "5.7".
INTERNAL FACTOR 1.0 IPRN 3        This reads the array values from the
 1.2 3.7 9.3 4.2 2.2 9.9 1.0   file that contains the control line.
 3.3 4.9 7.3 7.5 8.2 8.7 6.6   Thus, the values immediately follow the
 4.5 5.7 2.2 1.1 1.7 6.7 6.9   control line.
 7.4 3.5 7.8 8.5 7.4 6.8 8.8
OPEN/CLOSE inp.txt FACTOR 1.0 IPRN 3 Read array from formatted file "inp.dat".
OPEN/CLOSE inp.bin FACTOR 1.0 IPRN 3 Read array from binary file "inp.bin".
OPEN/CLOSE test.dat FACTOR 1.0 IPRN 3 Read array from file "test.dat".
```

Some arrays define information that is required for the entire model grid, or part of a model grid. This type of information is provided in a special type of data block called a "GRIDDATA" block. For example, hydraulic conductivity is required for every cell in the model grid. Hydraulic conductivity is read from a "GRIDDATA" block in the NPF Package input file. For GRIDDATA arrays with one value for every cell in the model grid, the arrays can optionally be read in a LAYERED format, in which an array is provided for each layer of the grid. Alternatively, the array can be read for the entire model grid. As an example, consider the GRIDDATA block for the IC Package shown below:

```
BEGIN GRIDDATA
```

```
  STRT [LAYERED]
    <strt(nodes)> -- READARRAY
END GRIDDATA
```

Here, the initial heads for the model are provided in the `strt` array. If the optional LAYERED keyword is present, then a separate array is provided for each layer. If the LAYERED keyword is not present, then the entire starting head array is read at once. The LAYERED keyword may be useful to discretization packages of type DIS and DISV, which support the concept of layers. Models defined with the DISU Package are not layered.

For a structured DIS model, the READARRAY utility is used to read arrays that are dimensioned to the full size of the grid (of size `nlay*nrow*ncol`). This utility first reads an array name, which associates the input to be read with the desired array. For these arrays, an optional keyword "LAYERED" can be located next to the array name. If "LAYERED" is detected, then a control line is provided for each layer and the array is filled with values for each model layer. If the "LAYERED" keyword is absent, then a single control line is used and the entire array is filled at once.

For example, the following block shows one way the starting head array (STRT) could be specified for a model with 4 layers. Following the array name and the "LAYERED" keyword are four control lines, one for each layer.

```
  STRT LAYERED
    CONSTANT 10.0 #layer 1
    CONSTANT 10.0 #layer 2
    CONSTANT 10.0 #layer 3
    CONSTANT 10.0 #layer 4
```

In this next example, the "LAYERED" keyword is absent. In this case, the control line applies to the entire `strt` array. One control line is required, and a constant value of 10.0 will be assigned to STRT for all cells in the model grid.

```
  STRT
    CONSTANT 10.0 #applies to all cells in the grid
```

List Input

Some items consist of several variables, such as layer, row, column, stage, and conductance, for example. List input refers to a block of data with a separate item on each line. For some common list types, the first set of variables is a cell identifier (denoted as `cellid` in this guide), such as layer, row, and column. With lists, the input data for each item must start on a new line. All variables for an item are assumed to be contained in a single line. Each input variable has a data type, which can be Double Precision, Integer, or Character. Integers are whole numbers and must not include a decimal point or exponent. Double Precision numbers can include a decimal point and an exponent. If no decimal point is included in the entered value, then the decimal point is assumed to be at the right side of the value. Any printable character is allowed for character variables.

Variables starting with the letters I-N are most commonly integers; however, in some instances, a character string may start with the letters I-N. Variables starting with the letters A-H and O-Z are primarily double precision numbers; however, these variable names may also be used for character data. In MODFLOW 6 all variables are explicitly declared within the source code, as opposed to the implicit type declaration in previous MODFLOW versions. This explicit declaration means that the variable type can be easily determined from the source code.

Free formatting is used throughout the input instructions. With free format, values are not required to occupy a fixed number of columns in a line. Each value can occupy one or more columns as required to represent the value; however, the values must still be included in the prescribed order. One or more

spaces, or a single comma optionally combined with spaces, must separate adjacent values. Also, a numeric value of zero must be explicitly represented with 0 and not by one or more spaces when free format is used, because detecting the difference between a space that represents 0 and a space that represents a value separator is not possible. Free format is similar to Fortran's list directed input.

Two capabilities included in Fortran's list-directed input are not included in the free-format input implemented in MODFLOW 6. Null values in which input values are left unchanged from their previous values are not allowed. In general, MODFLOW's input values are not defined prior to their input. A "/" cannot be used to terminate an input line without including values for all the variables; data values for all required input variables must be explicitly specified on an input line. For character data, MODFLOW's free format implementation is less stringent than the list-directed input of Fortran. Fortran requires character data to be delineated by apostrophes. MODFLOW does not require apostrophes unless a blank or a comma is part of a character variable.

As an example of a list, consider the PERIOD block for the GHB Package. The input format is shown below:

```
BEGIN PERIOD <iper>
  <cellid(ncelldim)> <bhead> <cond> [<aux(naux)>] [<boundname>]
  <cellid(ncelldim)> <bhead> <cond> [<aux(naux)>] [<boundname>]
  ...
END PERIOD
```

Each line represents a separate item, which consists of variables. In this case, the first variable of the item, `cellid` is an array of size `ncelldim`. The next two variables of the item are `bhead` and `cond`. Lastly, the item has two optional variables, `aux` and `boundname`. Three of the variables shown in the list are colored in blue. Variables that are colored in blue mean that they can be represented with a time series. The time series capability is described in the section on Time-Variable Input in this document.

The following is simple example of a PERIOD block for the GHB Package, which shows how a list is entered by the user.

```
BEGIN PERIOD 1
#      lay      row      col     stage     cond
         1       13        1     988.0    0.038
         1       14        9    1045.0    0.038
END PERIOD
```

As described earlier in the section on "Block and Keyword Input," block information can be read from a separate text file. To activate reading a list from separate text file, the first and only entry in the block must be a control line of the following form:

```
  OPEN/CLOSE <fname>
```

where `fname` is the name of the file containing the list. Lists for the stress packages (CHD, WEL, DRN, RIV, GHB, RCH, and EVT) have an additional BINARY option. The BINARY option is not supported for the advanced stress packages (LAK, MAW, SFR, UZF). The BINARY options is specified as follows:

```
  OPEN/CLOSE <fname> [(BINARY)]
```

If the (BINARY) keyword is found on the control line, then the file is opened as an unformatted file on unit 99, and the list is read. There are a number of requirements for using the (BINARY) option for lists. All stress package lists begin with integer values for the `cellid` (layer, row, and column, for example). These values must be represented as integer numbers in the unformatted file. Also, all auxiliary data must be included in the binary file; auxiliary data must be represented as double

precision numbers. Lastly, the (BINARY) option does not support entry of `boundname`, and so the BOUNDNAMES option should not be activated in the OPTIONS block for the package.

Units of Length and Time

The GWF Model formulates the groundwater flow equation without using prescribed length and time units. Any consistent units of length and time can be used when specifying the input data for a simulation. This capability gives a certain amount of freedom to the user, but care must be exercised to avoid mixing units. The program cannot detect the use of inconsistent units. For example, if hydraulic conductivity is entered in units of feet per day and pumpage as cubic meters per second, the program will run, but the results will be meaningless. Other processes generally are expected to work with consistent length and time units; however, other processes could conceivably place restrictions on which units are supported.

The user can set flags that specify the length and time units (see the input instructions for the Timing Module and Spatial Discretization Files), which may be useful in various parts of MODFLOW. For example, the program will label the table of simulation time with time units if the time units are specified by the optional TIME_UNITS label, which can be set in the TDIS Package. If the time units are not specified, the program still runs, but the table of simulation time does not indicate the time units. An optional LENGTH_UNITS label can be set in the Discretization Package. Situations in other processes may require that the length or time units be specified. In such situations, the input instructions will state the requirements. Remember that specifying the unit flags does not enforce consistent use of units. The user must insure that consistent units are used in all input data.

Steady-State Simulations

A steady-state simulation is represented by a single stress period having a single time step with the storage term set to zero. Setting the number and length of stress periods and time steps is the responsibility of the Timing Module of the MODFLOW 6 framework. The length of the stress period and time step will not affect the head solution because the time derivative is not calculated in a steady-state problem. Setting the storage term to zero is the responsibility of the Storage Package. Most other packages need not "know" that a simulation is steady state.

A GWF Model also can be mixed transient and steady state because each stress period can be designated transient or steady state. Thus, a GWF Model can start with a steady-state stress period and continue with one or more transient stress periods. The settings for controlling steady-state and transient options are in the Storage Package. If the Storage Package is not specified for a GWF Model, then the storage terms are zero and the GWF Model will be steady state.

Volumetric Budget

A summary of all inflows (sources) and outflows (sinks) of water is called a water budget. The water budget for the GWF Model is termed a volumetric budget because volumes of water and volumetric flow rates are involved; thus strictly speaking, a volumetric budget is not a mass balance, although this term has been used in other model reports. MODFLOW 6 calculates a water budget for the overall model as a check on the acceptability of the solution, and to provide a summary of the sources and sinks of water to the flow system. The water budget is printed to the GWF Model Listing File for selected time steps.

Numerical solution techniques for simultaneous equations do not always result in a correct answer; in particular, iterative solvers may stop iterating before a sufficiently close approximation to the solution is attained. A water budget provides an indication of the overall acceptability of the solution. The

system of equations solved by the model actually consists of a flow continuity statement for each model cell. Continuity should also exist for the total flows into and out of the model—that is, the difference between total inflow and total outflow should equal the total change in storage. In the model program, the water budget is calculated independently of the equation solution process, and in this sense may provide independent evidence of a valid solution.

The total budget as printed in the output does not include internal flows between model cells—only flows into or out of the model as a whole. For example, flow to or from rivers, flow to or from constant-head cells, and flow to or from wells are all included in the overall budget terms. Flow into and out of storage is also considered part of the overall budget inasmuch as accumulation in storage effectively removes water from the flow system and storage release effectively adds water to the flow—even though neither process, in itself, involves the transfer of water into or out of the ground-water regime. Each hydrologic package calculates its own contribution to the budget.

For every time step, the budget subroutine of each hydrologic package calculates the rate of flow into and out of the system due to the process simulated by the package. The inflows and outflows for each component of flow are stored separately. Most packages deal with only one such component of flow. In addition to flow, the volumes of water entering and leaving the model during the time step are calculated as the product of flow rate and time-step length. Cumulative volumes, from the beginning of the simulation, are then calculated and stored.

The GWF Model uses the inflows, outflows, and cumulative volumes to write the budget to the Listing File at the times requested by the model user. When a budget is written, the flow rates for the last time step and cumulative volumes from the beginning of simulation are written for each component of flow. Inflows are written separately from outflows. Following the convention indicated above, water entering storage is treated as an outflow (that is, as a loss of water from the flow system) while water released from storage is treated as an inflow (that is, a source of water to the flow system). In addition, total inflow and total outflow are written, as well as the difference between total inflow and outflow. The difference is then written as a percentage error, calculated using the formula:

$$D = \frac{100(IN - OUT)}{(IN + OUT)/2} \tag{1}$$

where D is the percentage error term, IN is the total inflow to the system, and OUT is the total outflow.

If the model equations are solved correctly, the percentage error should be small. In general, flow rates may be taken as an indication of solution validity for the time step to which they apply, while cumulative volumes are an indication of validity for the entire simulation up to the time of the output. The budget is written to the GWF Model Listing File at the end of each stress period whether requested or not.

Cell-By-Cell Flows

In some situations, calculating flow terms for various subregions of the model is useful. To facilitate such calculations, provision has been made to save flow terms for individual cells in a separate binary file so they can be used in computations external to the model itself. These individual cell flows are referred to here as "cell-by-cell" flow terms and are of four general types: (1) cell-by-cell stress flows, or flows into or from an individual cell caused by one of the external stresses represented in the model, such as evapotranspiration or recharge; (2) cell-by-cell storage terms, which give the rate of accumulation or depletion of storage in an individual cell; and (3) internal cell-by-cell flows, which are actually the flows across individual cell faces—that is, between adjacent model cells. These four kinds

of cell-by-cell flow terms are discussed further in subsequent paragraphs. To save any of these cell-by-cell terms, two flags in the model input must be set. The input to the Output Control file indicates the time steps for which cell-by-cell terms are to be saved. In addition, each hydrologic package includes an option called SAVE_FLOWS that must be set if the cell-by-cell terms computed by that package are to be saved. Thus, if the appropriate option in the Evapotranspiration Package input is set, cell-by-cell evapotranspiration terms will be saved for each time step for which the saving of cell-by-cell flow is requested through the Output Control Option. Only flow values are saved in the cell-by-cell files; neither water volumes nor cumulative water volumes are included. The flow dimensions are volume per unit time, where volume and time are in the same units used for all model input data. The cell-by-cell flow values are stored in unformatted form to make the most efficient use of disk space; see the Budget File section toward the end of this user guide for information on how the data are written to a file.

The cell-by-cell storage term gives the net flow to or from storage in a variable-head cell. The net storage for each cell in the grid is saved in transient simulations if the appropriate flags are set. Withdrawal from storage in the cell is considered positive, whereas accumulation in storage is considered negative.

The cell-by-cell constant-head flow term gives the flow into or out of an individual constant-head cell (specified with the CHD Package). This term is always associated with the constant-head cell itself, rather than with the surrounding cells that contribute or receive the flow. A constant-head cell may be surrounded by as many as six adjacent variable-head cells for a regular grid or any number of cells for the other grid types. The cell-by-cell calculation provides a single flow value for each constant-head cell, representing the algebraic sum of the flows between that cell and all of the adjacent variable-head cells. A positive value indicates that the net flow is away from the constant-head cell (into the variable-head part of the grid); a negative value indicates that the net flow is into the constant-head cell.

The internal cell-by-cell flow values represent flows across the individual faces of a model cell. Flows between cells are written in the compressed row storage format, whereby the flow between cell n and each one of its connecting m neighbor cells are contained in a single one-dimensional array. Flows are positive for the cell in question. Thus the flow reported for cell n and its connection with cell m is opposite in sign to the flow reported for cell m and its connection with cell n. These internal cell-by-cell flow values are useful in calculations of the groundwater flow into various subregions of the model, or in constructing flow vectors.

Cell-by-cell stress flows are flow rates into or out of the model, at a particular cell, owing to one particular external stress. For example, the cell-by-cell evapotranspiration term for cell n would give the flow out of the model by evapotranspiration from cell n. Cell-by-cell stress flows are considered positive if flow is into the cell, and negative if out of the cell.

GWF Model Name File

The GWF Model Name File specifies the options and packages that are active for a GWF model. The Name File contains two blocks: OPTIONS and PACKAGES. The length of each line must be 299 characters or less. The lines in each block can be in any order. Files listed in the PACKAGES block must exist when the program starts.

Comment lines are indicated when the first character in a line is one of the valid comment characters. Commented lines can be located anywhere in the file. Any text characters can follow the comment character. Comment lines have no effect on the simulation; their purpose is to allow users to provide documentation about a particular simulation.

Structure of Blocks

```
BEGIN OPTIONS
  [LIST <list>]
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [NEWTON [UNDER_RELAXATION]]
END OPTIONS
```

```
BEGIN PACKAGES
  <ftype> <fname> [<pname>]
  <ftype> <fname> [<pname>]
  ...
END PACKAGES
```

Explanation of Variables

list—is name of the listing file to create for this GWF model. If not specified, then the name of the list file will be the basename of the GWF model name file and the '.lst' extension. For example, if the GWF name file is called "my.model.nam" then the list file will be called "my.model.lst".

PRINT_INPUT—keyword to indicate that the list of all model stress package information will be written to the listing file immediately after it is read.

PRINT_FLOWS—keyword to indicate that the list of all model package flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and PRINT_FLOWS is specified, then flow rates are printed for the last time step of each stress period.

SAVE_FLOWS—keyword to indicate that all model package flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

NEWTON—keyword that activates the Newton-Raphson formulation for groundwater flow between connected, convertible groundwater cells and stress packages that support calculation of Newton-Raphson terms for groundwater exchanges. Cells will not dry when this option is used. By default, the Newton-Raphson formulation is not applied.

UNDER_RELAXATION—keyword that indicates whether the groundwater head in a cell will be under-relaxed when water levels fall below the bottom of the model below any given cell. By default, Newton-Raphson UNDER_RELAXATION is not applied.

ftype—is the file type, which must be one of the following character values shown in table 5. Ftype may be entered in any combination of uppercase and lowercase.

fname—is the name of the file containing the package input. The path to the file should be included if the file is not located in the folder where the program was run.

pname—is the user-defined name for the package. **Pname** is restricted to 16 characters. No spaces are allowed in **Pname**. **Pname** character values are read and stored by the program for stress packages only. These names may be useful for labeling purposes when multiple stress packages of the same type are located within a single GWF Model. If **Pname** is specified for a stress package, then **Pname** will be used in the flow budget table in the listing file; it will also be used for the text entry in the cell-by-cell budget file. **Pname** is case insensitive and is stored in all upper case letters.

Table 5: Ftype values described in this report. The **Pname** column indicates whether or not a package name can be provided in the name file

Ftype	Input File Description	Pname
LIST	Listing File	
DIS6	Rectilinear Discretization Input File	
DISV6	Discretization by Vertices Input File	
DISU6	Unstructured Discretization Input File	
IC6	Initial Conditions Package	
OC6	Output Control Option	
NPF6	Node Property Flow Package	
STO6	Storage Package	
HFB6	Horizontal Flow Barrier Package	
CHD6	Time-Variant Specified Head Option	*
WEL6	Well Package	*
DRN6	Drain Package	*
RIV6	River Package	*
GHB6	General-Head Boundary Package	*
RCH6	Recharge Package	*
EVT6	Evapotranspiration Package	*
MAW6	Multi-Aquifer Well Package	*
SFR6	Streamflow Routing Package	*
LAK6	Lake Package	*
UZF6	Unsaturated Zone Flow Package	*
MVR6	Water Mover Package	
GNC6	Ghost-Node Correction Package	
OBS6	Observations Option	

Example Input File

```
# This block is optional
BEGIN OPTIONS
  PRINT_INPUT
  PRINT_FLOWS
```

```
  SAVE_FLOWS
END OPTIONS


# List of packages. List can be listed in any order.
BEGIN PACKAGES
  IC6         bcf2ss.ic
  NPF6        bcf2ss.npf
  WEL6        bcf2ss.wel WEL-COUNTY
  RIV6        bcf2ss.riv
  RCH6        bcf2ss.rch
  OC6         bcf2ss.oc
  DIS6        bcf2ss.dis
END PACKAGES
```

Structured Discretization (DIS) Input File

Discretization information for structured grids is read from the file that is specified by "DIS6" as the file type. Only one discretization input file (DISU6, DISV6 or DIS6) can be specified for a model.

Structure of Blocks

```
BEGIN OPTIONS
  [LENGTH_UNITS <length_units>]
  [NOGRB]
  [XORIGIN <xorigin>]
  [YORIGIN <yorigin>]
  [ANGROT <angrot>]
END OPTIONS


BEGIN DIMENSIONS
  NLAY <nlay>
  NROW <nrow>
  NCOL <ncol>
END DIMENSIONS


BEGIN GRIDDATA
  DELR
    <delr(ncol)> -- U1DDBL
  DELC
    <delc(nrow)> -- U1DDBL
  TOP
    <top(ncol, nrow)> -- U2DDBL
  BOTM [LAYERED]
    <botm(ncol, nrow, nlay)> -- READARRAY
  [IDOMAIN [LAYERED]
    <idomain(ncol, nrow, nlay)> -- READARRAY]
END GRIDDATA
```

Explanation of Variables

length_units—is the length units used for this model. Values can be "FEET", "METERS", or "CENTIMETERS". If not specified, the default is "UNKNOWN".

NOGRB—keyword to deactivate writing of the binary grid file.

xorigin—x-position of the lower-left corner of the model grid. A default value of zero is assigned if not specified. The value for xorigin does not affect the model simulation, but it is written to the binary grid file so that postprocessors can locate the grid in space.

yorigin—y-position of the lower-left corner of the model grid. If not specified, then a default value equal to zero is used. The value for yorigin does not affect the model simulation, but it is written to the binary grid file so that postprocessors can locate the grid in space.

angrot—counter-clockwise rotation angle (in degrees) of the lower-left corner of the model grid. If not specified, then a default value of 0.0 is assigned. The value for angrot does not affect the model simulation, but it is written to the binary grid file so that postprocessors can locate the grid in space.

nlay—is the number of layers in the model grid.

nrow—is the number of rows in the model grid.

ncol—is the number of columns in the model grid.

delr—is the is the column spacing in the row direction.

delc—is the is the row spacing in the column direction.

top—is the top elevation for each cell in the top model layer.

botm—is the bottom elevation for each cell.

idomain—is an optional array that characterizes the existence status of a cell. If the idomain array is not specified, then all model cells exist within the solution. If the idomain value for a cell is 0, the cell does not exist in the simulation. Input and output values will be read and written for the cell, but internal to the program, the cell is excluded from the solution. If the idomain value for a cell is 1, the cell exists in the simulation. If the idomain value for a cell is -1, the cell does not exist in the simulation. Furthermore, the first existing cell above will be connected to the first existing cell below. This type of cell is referred to as a "vertical pass through" cell.

Example Input File

```
#The OPTIONS block is optional
BEGIN OPTIONS
  LENGTH_UNITS METERS
END OPTIONS

#The DIMENSIONS block is required
BEGIN DIMENSIONS
  NLAY 10
  NROW 1
  NCOL 21
END DIMENSIONS

#The GRIDDATA block is required
BEGIN GRIDDATA
  DELR
    INTERNAL FACTOR 1.
      .1 .1 .1 .1 .1 .1 .1 .1 .1 .1 .1 .1 .1 .1 .1 .1 .1 .1 .1 .1 0.01
  DELC
    CONSTANT 1.0
  TOP LAYERED
    CONSTANT 1.
  BOTM LAYERED
    CONSTANT 0.9
    CONSTANT 0.8
    CONSTANT 0.7
    CONSTANT 0.6
    CONSTANT 0.5
    CONSTANT 0.4
    CONSTANT 0.3
    CONSTANT 0.2
```

```
    CONSTANT 0.1
    CONSTANT 0.0
END GRIDDATA
```

Discretization with Vertices (DISV) Input File

Discretization information for DISV grids is read from the file that is specified by "DISV6" as the file type. Only one discretization input file (DISV6, DISU6 or DIS6) can be specified for a model.

The approach for numbering cell and cell vertices for the DISV Package is shown in figure 1. The list of vertices for a cell must be in clockwise order. Closing of the cell polygon by repeating the first vertex as the last vertex is not required in the present implementation. Internally within the program, however, the first vertex number is added to the end of the vertex list in order to close the polygon. Thus, users have the option for whether or not to close cell polygons.

A. Quad-based grid

Cell	*NVERT*	Vertices
1	4	[1 2 3 4]
2	5	[2 5 6 9 3]
3	5	[4 3 11 7 8]
4	4	[3 9 10 11]
5	4	[9 6 12 10]
6	4	[11 10 13 7]
7	4	[10 12 14 13]

B. Triangular grid

Cell	*NVERT*	Vertices
1	3	[1 5 4]
2	3	[1 2 5]
3	3	[2 6 5]
4	3	[2 3 6]
5	3	[3 7 6]
6	3	[4 5 8]
7	3	[5 9 8]
8	3	[5 6 9]
9	3	[6 10 9]
10	3	[6 7 10]

EXPLANATION

2 • Cell center and cell number
4 ○ Vertex and vertex number
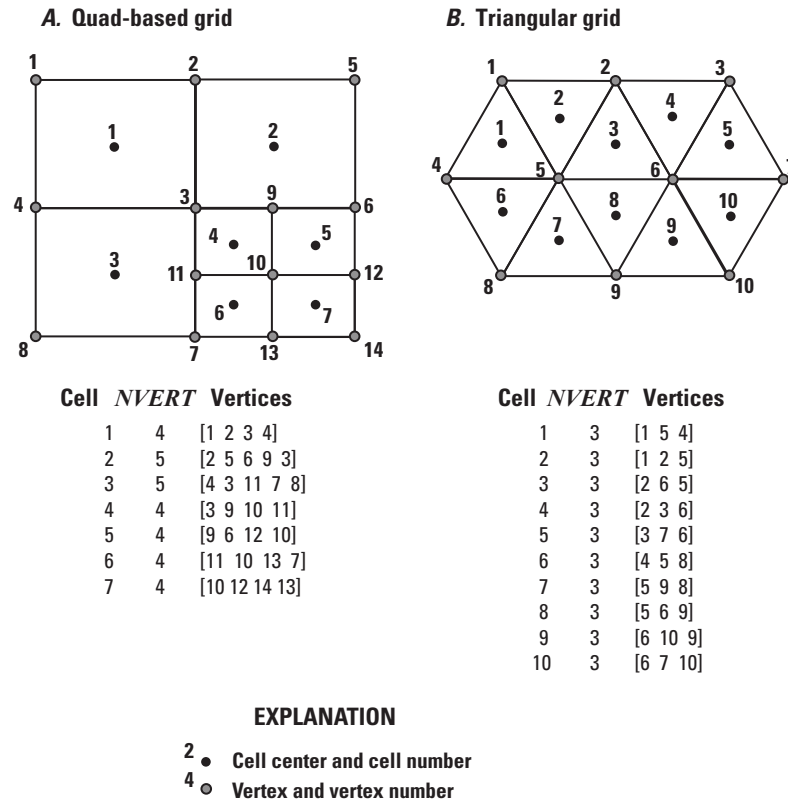
Figure 1: Schematic diagram showing the vertices and cells defined using the Discretization by Vertices Package. The list of vertices used to define each cell must be in clockwise order. From Langevin and others (2017)

Structure of Blocks

```
BEGIN OPTIONS
  [LENGTH_UNITS <length_units>]
  [NOGRB]
  [XORIGIN <xorigin>]
  [YORIGIN <yorigin>]
  [ANGROT <angrot>]
END OPTIONS
```

```
BEGIN DIMENSIONS
  NLAY <nlay>
  NCPL <ncpl>
  NVERT <nvert>
END DIMENSIONS

BEGIN GRIDDATA
  TOP
    <top(1, ncpl)> -- U2DDBL
  BOTM [LAYERED]
    <botm(nlay, ncpl)> -- READARRAY
  [IDOMAIN [LAYERED]
    <idomain(nlay, ncpl)> -- READARRAY]
END GRIDDATA

BEGIN VERTICES
  <iv> <xv> <yv>
  <iv> <xv> <yv>
  ...
END VERTICES

BEGIN CELL2D
  <icell2d> <xc> <yc> <ncvert> <icvert(ncvert)>
  <icell2d> <xc> <yc> <ncvert> <icvert(ncvert)>
  ...
END CELL2D
```

Explanation of Variables

length_units—is the length units used for this model. Values can be "FEET", "METERS", or "CENTIMETERS". If not specified, the default is "UNKNOWN".

NOGRB—keyword to deactivate writing of the binary grid file.

xorigin—x-position of the origin used for model grid vertices. This value should be provided in a real-world coordinate system. A default value of zero is assigned if not specified. The value for xorigin does not affect the model simulation, but it is written to the binary grid file so that postprocessors can locate the grid in space.

yorigin—y-position of the origin used for model grid vertices. This value should be provided in a real-world coordinate system. If not specified, then a default value equal to zero is used. The value for yorigin does not affect the model simulation, but it is written to the binary grid file so that postprocessors can locate the grid in space.

angrot—counter-clockwise rotation angle (in degrees) of the model grid coordinate system relative to a real-world coordinate system. If not specified, then a default value of 0.0 is assigned. The value for angrot does not affect the model simulation, but it is written to the binary grid file so that postprocessors can locate the grid in space.

nlay—is the number of layers in the model grid.

ncpl—is the number of cells per layer. This is a constant value for the grid and it applies to all layers.

nvert—is the total number of (x, y) vertex pairs used to characterize the horizontal configuration of the model grid.

top—is the top elevation for each cell in the top model layer.

botm—is the bottom elevation for each cell.

idomain—is an optional array that characterizes the existence status of a cell. If the idomain array is not specified, then all model cells exist within the solution. If the idomain value for a cell is 0, the cell does not exist in the simulation. Input and output values will be read and written for the cell, but internal to the program, the cell is excluded from the solution. If the idomain value for a cell is 1, the cell exists in the simulation. If the idomain value for a cell is -1, the cell does not exist in the simulation. Furthermore, the first existing cell above will be connected to the first existing cell below. This type of cell is referred to as a "vertical pass through" cell.

iv—is the vertex number. Records in the VERTICES block must be listed in consecutive order from 1 to nvert.

xv—is the x-coordinate for the vertex.

yv—is the y-coordinate for the vertex.

icell2d—is the cell2d number. Records in the CELL2D block must be listed in consecutive order from 1 to ncpl.

xc—is the x-coordinate for the cell center.

yc—is the y-coordinate for the cell center.

ncvert—is the number of vertices required to define the cell. There may be a different number of vertices for each cell.

icvert—is an array of integer values containing vertex numbers (in the VERTICES block) used to define the cell. Vertices must be listed in clockwise order. Cells that are connected must share vertices.

Example Input File

```
#The OPTIONS block is optional
BEGIN OPTIONS
  LENGTH_UNITS METERS
END OPTIONS

#The DIMENSIONS block is required
BEGIN DIMENSIONS
  NCPL 4
  NLAY 3
  NVERT 9
END DIMENSIONS

#The GRIDDATA block is required
BEGIN GRIDDATA
  TOP
    CONSTANT 3.0
  BOTM LAYERED
    CONSTANT 2.0
```

```
      CONSTANT 1.0
      CONSTANT 0.0
    IDOMAIN LAYERED
      INTERNAL FACTOR 1
        1 1 1 0
      CONSTANT 1
      CONSTANT 1
END GRIDDATA

#The VERTICES block is required
BEGIN VERTICES
  1 0. 1.
  2 .5 1.
  3 1. 1.
  4 0 .5
  5 .5 .5
  6 1. .5
  7 0. 0.
  8 .5 0.
  9 1. 0.
END VERTICES

BEGIN CELL2D
  1 .25 .75 4 1 2 5 4
  2 .75 .75 4 2 3 6 5
  3 .25 .25 4 4 5 8 7
  4 .75 .25 4 5 6 9 8
END CELL2D
```

Unstructured Discretization (DISU) Input File

Discretization information for unstructured grids is read from the file that is specified by "DISU6" as the file type. Only one discretization input file (DISU6, DISV6 or DIS6) can be specified for a model.

The shape and position of each cell can be defined using vertices. This information is optional and is only read if the number of vertices (NVERT) in the DIMENSIONS block is specified and is assigned a value larger than zero. If the vertices and two-dimensional cell information is provided in this file, then this information is also written to the binary grid file. Providing this information may be useful for other postprocessing programs that read the binary grid file.

Structure of Blocks

```
BEGIN OPTIONS
  [LENGTH_UNITS <length_units>]
  [NOGRB]
  [XORIGIN <xorigin>]
  [YORIGIN <yorigin>]
  [ANGROT <angrot>]
END OPTIONS
```

```
BEGIN DIMENSIONS
  NODES <nodes>
  NJA <nja>
  [NVERT <nvert>]
END DIMENSIONS
```

```
BEGIN GRIDDATA
  TOP
    <top(nodes)> -- U1DDBL
  BOT
    <bot(nodes)> -- U1DDBL
  AREA
    <area(nodes)> -- U1DDBL
END GRIDDATA
```

```
BEGIN CONNECTIONDATA
  IAC
    <iac(nodes)> -- U1DINT
  JA
    <ja(nja)> -- U1DINT
  IHC
    <ihc(nja)> -- U1DINT
  CL12
    <cl12(nja)> -- U1DDBL
  HWVA
    <hwva(nja)> -- U1DDBL
  [ANGLDEGX
    <angldegx(nja)> -- U1DDBL]
END CONNECTIONDATA
```

```
BEGIN VERTICES
  <iv> <xv> <yv>
  <iv> <xv> <yv>
  ...
END VERTICES
```

```
BEGIN CELL2D
  <icell2d> <xc> <yc> <ncvert> <icvert(ncvert)>
  <icell2d> <xc> <yc> <ncvert> <icvert(ncvert)>
  ...
END CELL2D
```

Explanation of Variables

length_units—is the length units used for this model. Values can be "FEET", "METERS", or "CENTIMETERS". If not specified, the default is "UNKNOWN".

NOGRB—keyword to deactivate writing of the binary grid file.

xorigin—x-position of the origin used for model grid vertices. This value should be provided in a real-world coordinate system. A default value of zero is assigned if not specified. The value for xorigin does not affect the model simulation, but it is written to the binary grid file so that postprocessors can locate the grid in space.

yorigin—y-position of the origin used for model grid vertices. This value should be provided in a real-world coordinate system. If not specified, then a default value equal to zero is used. The value for yorigin does not affect the model simulation, but it is written to the binary grid file so that postprocessors can locate the grid in space.

angrot—counter-clockwise rotation angle (in degrees) of the model grid coordinate system relative to a real-world coordinate system. If not specified, then a default value of 0.0 is assigned. The value for angrot does not affect the model simulation, but it is written to the binary grid file so that postprocessors can locate the grid in space.

nodes—is the number of cells in the model grid.

nja—is the sum of the number of connections and nodes. When calculating the total number of connections, the connection between cell n and cell m is considered to be different from the connection between cell m and cell n. Thus, nja is equal to the total number of connections, including n to m and m to n, and the total number of cells.

nvert—is the total number of (x, y) vertex pairs used to define the plan-view shape of each cell in the model grid. If nvert is not specified or is specified as zero, then the VERTICES and CELL2D blocks below are not read.

top—is the top elevation for each cell in the model grid.

bot—is the bottom elevation for each cell.

area—is the cell surface area (in plan view).

iac—is the number of connections (plus 1) for each cell. The sum of iac must be equal to nja.

ja—is a list of cell number (n) followed by its connecting cell numbers (m) for each of the m cells connected to cell n. The number of values to provide for cell n is iac(n). This list is sequentially provided for the first to the last cell. The first value in the list must be cell n itself, and the remaining cells must be listed in an increasing order (sorted from lowest number to highest). Note that the cell and its connections are only supplied for the GWF cells and their connections to the other GWF cells. Also note that the JA list input may be chopped up to have every node

number and its connectivity list on a separate line for ease in readability of the file. To further ease readability of the file, the node number of the cell whose connectivity is subsequently listed, may be expressed as a negative number the sign of which is subsequently corrected by the code.

ihc—is an index array indicating the direction between node n and all of its m connections. If $ihc = 0$ – cell n and cell m are connected in the vertical direction. Cell n overlies cell m if the cell number for n is less than m; cell m overlies cell n if the cell number for m is less than n. If $ihc = 1$ – cell n and cell m are connected in the horizontal direction. If $ihc = 2$ – cell n and cell m are connected in the horizontal direction, and the connection is vertically staggered. A vertically staggered connection is one in which a cell is horizontally connected to more than one cell in a horizontal connection.

cl12—is the array containing connection lengths between the center of cell n and the shared face with each adjacent m cell.

hwva—is a symmetric array of size nja. For horizontal connections, entries in hwva are the horizontal width perpendicular to flow. For vertical connections, entries in hwva are the vertical area for flow. Thus, values in the hwva array contain dimensions of both length and area. Entries in the hwva array have a one-to-one correspondence with the connections specified in the ja array. Likewise, there is a one-to-one correspondence between entries in the hwva array and entries in the ihc array, which specifies the connection type (horizontal or vertical). Entries in the hwva array must be symmetric; the program will terminate with an error if the value for hwva for an $n - m$ connection does not equal the value for hwva for the corresponding $m - n$ connection.

angldegx—is the angle (in degrees) between the horizontal x-axis and the outward normal to the face between a cell and its connecting cells (see figure 8 in the MODFLOW-USG documentation). The angle varies between zero and 360.0. angldegx is only needed if horizontal anisotropy is specified in the NPF Package. angldegx does not need to be specified if horizontal anisotropy is not used. angldegx is of size nja; values specified for vertical connections and for the diagonal position are not used. Note that angldegx is read in degrees, which is different from MODFLOW-USG, which reads a similar variable (anglex) in radians.

iv—is the vertex number. Records in the VERTICES block must be listed in consecutive order from 1 to nvert.

xv—is the x-coordinate for the vertex.

yv—is the y-coordinate for the vertex.

icell2d—is the cell2d number. Records in the CELL2D block must be listed in consecutive order from 1 to nodes.

xc—is the x-coordinate for the cell center.

yc—is the y-coordinate for the cell center.

ncvert—is the number of vertices required to define the cell. There may be a different number of vertices for each cell.

icvert—is an array of integer values containing vertex numbers (in the VERTICES block) used to define the cell. Vertices must be listed in clockwise order.

Example Input File

```
BEGIN OPTIONS
  LENGTH_UNITS METERS
END OPTIONS

BEGIN DIMENSIONS
  NODES 9
  NJA 33
END DIMENSIONS

BEGIN GRIDDATA
  TOP
    CONSTANT 0.
  BOT
    CONSTANT -10
  AREA
    INTERNAL FACTOR 1
      10000 10000 10000 10000 10000 10000 10000 10000 10000
END GRIDDATA

BEGIN CONNECTIONDATA
  IHC
    CONSTANT 1
  IAC
    INTERNAL FACTOR 1
    3 4 3 4 5 4 3 4 3
  JA
    INTERNAL FACTOR 1
    1 2 4
    2 1 3 5
    3 2 6
    4 1 5 7
    5 2 4 6 8
    6 3 5 9
    7 4 8
    8 5 7 9
    9 6 8
  CL12
    INTERNAL FACTOR 1
    0 50 50
    0 50 50 50
    0 50 50
    0 50 50 50
    0 50 50 50 50
    0 50 50 50
    0 50 50
    0 50 50 50
    0 50 50
  HWVA
    INTERNAL FACTOR 1
    0 100 100
    0 100 100 100
    0 100 100
    0 100 100 100
    0 100 100 100 100
    0 100 100 100
    0 100 100
    0 100 100 100
```

```
    0 100 100
END CONNECTIONDATA
```

Initial Conditions (IC) Package

Initial Conditions (IC) Package information is read from the file that is specified by "IC6" as the file type. Only one IC Package can be specified for a GWF model.

Structure of Blocks

```
BEGIN GRIDDATA
  STRT [LAYERED]
    <strt(nodes)> -- READARRAY
END GRIDDATA
```

Explanation of Variables

strt—is the initial (starting) head—that is, head at the beginning of the GWF Model simulation. strt must be specified for all simulations, including steady-state simulations. One value is read for every model cell. For simulations in which the first stress period is steady state, the values used for STRT generally do not affect the simulation (exceptions may occur if cells go dry and (or) rewet). The execution time, however, will be less if STRT includes hydraulic heads that are close to the steady-state solution. A head value lower than the cell bottom can be provided if a cell should start as dry.

Example Input File

```
#The OPTIONS block is optional
BEGIN OPTIONS
END OPTIONS

#The GRIDDATA block is required
BEGIN GRIDDATA
  STRT LAYERED
    CONSTANT  0.0  Initial Head layer 1
    CONSTANT  0.0  Initial Head layer 2
END GRIDDATA
```

Output Control (OC) Option

Input to the Output Control Option of the Groundwater Flow Model is read from the file that is specified as type "OC6" in the Name File. If no "OC6" file is specified, default output control is used. The Output Control Option determines how and when heads are printed to the listing file and/or written to a separate binary output file. Under the default, head and overall budget are written to the Listing File at the end of every stress period. The default printout format for head is 10G11.4.

Output Control data must be specified using words. The numeric codes supported in earlier MODFLOW versions can no longer be used.

All budget output is saved in the "COMPACT BUDGET" form. COMPACT BUDGET indicates that the cell-by-cell budget file(s) will be written in a more compact form than is used in the 1988 version of MODFLOW (McDonald and Harbaugh, 1988); however, programs that read these data in the form written by MODFLOW-88 will be unable to read the new compact file.

For the PRINT and SAVE options of heads, there is no longer an option to specify individual layers. Whenever one of these arrays is printed or saved, all layers are printed or saved.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [BUDGET FILEOUT <budgetfile>]
  [HEAD FILEOUT <headfile>]
  [HEAD PRINT_FORMAT COLUMNS <columns> WIDTH <width> DIGITS <digits> <format>]
END OPTIONS
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  [SAVE <rtype> <ocsetting>]
  [PRINT <rtype> <ocsetting>]
END PERIOD
```

Explanation of Variables

BUDGET—keyword to specify that record corresponds to the budget.

FILEOUT—keyword to specify that an output filename is expected next.

budgetfile—name of the output file to write budget information.

HEAD—keyword to specify that record corresponds to head.

headfile—name of the output file to write head information.

PRINT_FORMAT—keyword to specify format for printing to the listing file.

columns—number of columns for writing data.

width—width for writing each number.

`digits`—number of digits to use for writing a number.

`format`—write format can be EXPONENTIAL, FIXED, GENERAL, or SCIENTIFIC.

`iper`—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. `iper` must be less than `nper` in the TDIS Package and greater than zero. The `iper` value assigned to a stress period block must be greater than the `iper` value assigned for the previous block.

`SAVE`—keyword to indicate that information will be saved this stress period.

`PRINT`—keyword to indicate that information will be printed this stress period.

`rtype`—type of information to save or print. Can be BUDGET or HEAD.

`ocsetting`—specifies the steps for which the data will be saved.

```
ALL
FIRST
LAST
FREQUENCY <frequency>
STEPS <steps(nstp)>
```

`ALL`—keyword to indicate save for all time steps in period.

`FIRST`—keyword to indicate save for first step in period. This keyword may be used in conjunction with other keywords to print or save results for multiple time steps.

`LAST`—keyword to indicate save for last step in period. This keyword may be used in conjunction with other keywords to print or save results for multiple time steps.

`frequency`—save at the specified time step frequency. This keyword may be used in conjunction with other keywords to print or save results for multiple time steps.

`steps`—save for each step specified in `steps`. This keyword may be used in conjunction with other keywords to print or save results for multiple time steps.

Example Input File

```
BEGIN OPTIONS
  HEAD FILEOUT AdvGW_tidal.hds
  BUDGET FILEOUT AdvGW_tidal.cbc
  HEAD PRINT_FORMAT COLUMNS 100 WIDTH 15 DIGITS 4 GENERAL
END OPTIONS

BEGIN PERIOD 1
  PRINT HEAD FIRST
  PRINT HEAD LAST
  PRINT BUDGET LAST
  SAVE HEAD ALL
  SAVE BUDGET ALL
END PERIOD

# No output for stress periods 2 through 24
```

```
BEGIN PERIOD 2
END PERIOD

BEGIN PERIOD 25
  PRINT HEAD STEPS 6 12 23
  SAVE BUDGET FIRST
  SAVE BUDGET LAST
  SAVE BUDGET FREQUENCY 5
END PERIOD
```

Observation (OBS) Utility for a GWF Model

GWF Model observations include the simulated groundwater head (`head`), calculated drawdown (`drawdown`) at a node, and the flow between two connected nodes (`flow-ja-face`). The data required for each GWF Model observation type is defined in table 6. For `flow-ja-face` observation types, negative and positive values represent a loss from and gain to the `cellid` specified for ID, respectively.

Table 6: Available GWF model observation types

Model	Observation type	ID	ID2	Description
GWF	head	cellid	–	Head at a specified cell.
GWF	drawdown	cellid	–	Drawdown at a specified cell calculated as difference between starting head and simulated head for the time step.
GWF	flow-ja-face	cellid	cellid	Flow between two adjacent cells.

Example Observation Input File

An example GWF Model observation file is shown below.

```
begin options
  digits 10
  print_input
end options

begin continuous FILEOUT my_model.gwf.head.csv
# obsname  obstype  ID
  L1       head     1 51 51 # heads at lay 1 row 51 col 51
  L2       head     2 51 51 # heads at lay 2 row 51 col 51
end continuous

begin continuous FILEOUT my_model.gwf.ddn.csv
# obsname  obstype  ID
  L1ddn    drawdown 1 51 51 # heads at lay 1 row 51 col 51
  L2ddn    drawdown 2 51 51 # heads at lay 2 row 51 col 51
end continuous

begin continuous FILEOUT my_model.gwf.flow.csv
# obsname  obstype      ID      ID1
  L1rfflow flow-ja-face 1 51 51 1 51 52
  L2rfflow flow-ja-face 2 51 51 2 51 52
  L1-L2flow flow-ja-face 1 51 51 2 51 51
end continuous
```

Node Property Flow (NPF) Package

Input to the Node Property Flow (NPF) Package is read from the file that has type "NPF6" in the Name File. A single NPF Package is required for each GWF model.

Structure of Blocks

```
BEGIN OPTIONS
  [SAVE_FLOWS]
  [ALTERNATIVE_CELL_AVERAGING <alternative_cell_averaging>]
  [THICKSTRT]
  [VARIABLECV [DEWATERED]]
  [PERCHED]
  [REWET WETFCT <wetfct> IWETIT <iwetit> IHDWET <ihdwet>]
  [XT3D [RHS]]
END OPTIONS
```

```
BEGIN GRIDDATA
  ICELLTYPE [LAYERED]
    <icelltype(nodes)> -- READARRAY
  K [LAYERED]
    <k(nodes)> -- READARRAY
  [K22 [LAYERED]
    <k22(nodes)> -- READARRAY]
  [K33 [LAYERED]
    <k33(nodes)> -- READARRAY]
  [WETDRY [LAYERED]
    <wetdry(nodes)> -- READARRAY]
  [ANGLE1 [LAYERED]
    <angle1(nodes)> -- READARRAY]
  [ANGLE2 [LAYERED]
    <angle2(nodes)> -- READARRAY]
  [ANGLE3 [LAYERED]
    <angle3(nodes)> -- READARRAY]
END GRIDDATA
```

Explanation of Variables

SAVE_FLOWS—keyword to indicate that cell-by-cell flow terms will be written to the file specified with "BUDGET SAVE FILE" in Output Control.

alternative_cell_averaging—is a text keyword to indicate that an alternative method will be used for calculating the conductance for horizontal cell connections. The text value for alternative_cell_averaging can be "LOGARITHMIC", "AMT-LMK", or "AMT-HMK". "AMT-LMK" signifies that the conductance will be calculated using arithmetic-mean thickness and logarithmic-mean hydraulic conductivity. "AMT-HMK" signifies that the conductance will be calculated using arithmetic-mean thickness and harmonic-mean hydraulic conductivity. If the user does not specify a value for alternative_cell_averaging, then the harmonic-mean method will be used. This option cannot be used if the XT3D option is invoked.

THICKSTRT—indicates that cells having a negative icelltype are confined, and their cell thickness for conductance calculations will be computed as STRT-BOT rather than TOP-BOT.

VARIABLECV—keyword to indicate that the vertical conductance will be calculated using the saturated thickness and properties of the overlying cell and the thickness and properties of the underlying cell. If the DEWATERED keyword is also specified, then the vertical conductance is calculated using only the saturated thickness and properties of the overlying cell if the head in the underlying cell is below its top. If these keywords are not specified, then the default condition is to calculate the vertical conductance at the start of the simulation using the initial head and the cell properties. The vertical conductance remains constant for the entire simulation.

DEWATERED—If the DEWATERED keyword is specified, then the vertical conductance is calculated using only the saturated thickness and properties of the overlying cell if the head in the underlying cell is below its top.

PERCHED—keyword to indicate that when a cell is overlying a dewatered convertible cell, the head difference used in Darcy's Law is equal to the head in the overlying cell minus the bottom elevation of the overlying cell. If not specified, then the default is to use the head difference between the two cells.

REWET—activates model rewetting. Rewetting is off by default.

wetfct—is a keyword and factor that is included in the calculation of the head that is initially established at a cell when that cell is converted from dry to wet.

iwetit—is a keyword and iteration interval for attempting to wet cells. Wetting is attempted every iwetit iteration. This applies to outer iterations and not inner iterations. If iwetit is specified as zero or less, then the value is changed to 1.

ihdwet—is a keyword and integer flag that determines which equation is used to define the initial head at cells that become wet. If ihdwet is 0, $h = BOT + WETFCT(hm - BOT)$. If ihdwet is not 0, $h = BOT + WETFCT(THRESH)$.

XT3D—keyword indicating that the XT3D formulation will be used. If the RHS keyword is also included, then the XT3D additional terms will be added to the right-hand side. If the RHS keyword is excluded, then the XT3D terms will be put into the coefficient matrix. Use of XT3D will substantially increase the computational effort, but will result in improved accuracy for anisotropic conductivity fields and for unstructured grids in which the CVFD requirement is violated.

RHS—If the RHS keyword is also included, then the XT3D additional terms will be added to the right-hand side. If the RHS keyword is excluded, then the XT3D terms will be put into the coefficient matrix.

icelltype—flag for each cell that specifies how saturated thickness is treated. 0 means saturated thickness is held constant; >0 means saturated thickness varies with computed head when head is below the cell top; <0 means saturated thickness varies with computed head unless the THICKSTRT option is in effect. When THICKSTRT is in effect, a negative value of icelltype indicates that saturated thickness will be computed as STRT-BOT and held constant.

k—is the hydraulic conductivity. When anisotropy is applied, k corresponds to the K_{11} tensor component. All included cells (IDOMAIN > 0) must have a k value greater than zero.

k22—is the hydraulic conductivity of second ellipsoid axis; for an unrotated case this is the hydraulic conductivity in the y direction. If k22 is not included in the GRIDDATA block, then $k22$ is equal to k. For a regular grid in which no rotation angles are specified, k22 is the hydraulic conductivity

along columns in the y direction. For an unstructured grid, the user must assign principal x and y axes and provide the angle for each cell connection relative to the assumed x direction. All included cells (`IDOMAIN` > 0) must have a `k22` value greater than zero.

`k33`—is the hydraulic conductivity of third ellipsoid axis; for an unrotated case, this is the vertical hydraulic conductivity. When anisotropy is applied, `k33` corresponds to the K_{33} tensor component. All included cells (`IDOMAIN` > 0) must have a `k33` value greater than zero.

`wetdry`—is a combination of the wetting threshold and a flag to indicate which neighboring cells can cause a cell to become wet. If `wetdry` < 0, only a cell below a dry cell can cause the cell to become wet. If `wetdry` > 0, the cell below a dry cell and horizontally adjacent cells can cause a cell to become wet. If `wetdry` is 0, the cell cannot be wetted. The absolute value of `wetdry` is the wetting threshold. When the sum of BOT and the absolute value of `wetdry` at a dry cell is equaled or exceeded by the head at an adjacent cell, the cell is wetted. `wetdry` must be specified if "REWET" is specified in the OPTIONS block. If "REWET" is not specified in the options block, then `wetdry` can be entered, and memory will be allocated for it, even though it is not used.

`angle1`—is rotation angle of the hydraulic conductivity tensor in degrees. The angle represents the counter-clockwise rotation of the hydraulic conductivity ellipsoid within the $x - y$ plane. A value of zero, which is the default value if `angle1` is not specified, indicates that K_{11} is aligned with the x-axis of the model grid (the row direction for a regular grid) and K_{22} is aligned with the y-axis of the model grid (the column direction for a regular grid).

`angle2`—is rotation angle of the hydraulic conductivity tensor in degrees. The angle represents the tilt of the hydraulic conductivity ellipsoid up and out of the $x - y$ plane.

`angle3`—is rotation angle of the hydraulic conductivity tensor in degrees. The angle represents the rotation of the hydraulic conductivity tensor around its dominant axis.

Example Input File

```
BEGIN OPTIONS
  SAVE_FLOWS
END OPTIONS

BEGIN GRIDDATA
  #
  #icelltype(nodes) is 0:confined, 1:convertible
  ICELLTYPE
    constant 0
  #
  # horizontal hydraulic conductivity
  K
    constant 1.0
  #
  # vertical hydraulic conductivity
  K33
    constant 0.1
END GRIDDATA
```

Horizontal Flow Barrier (HFB) Package

Input to the Horizontal Flow Barrier (HFB) Package is read from the file that has type "HFB6" in the Name File. Only one HFB Package can be specified for a GWF model.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [PRINT_INPUT]
END OPTIONS
```

```
BEGIN DIMENSIONS
  MAXHFB <maxhfb>
END DIMENSIONS
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <cellid1(ncelldim)> <cellid2(ncelldim)> <hydchr>
  <cellid1(ncelldim)> <cellid2(ncelldim)> <hydchr>
  ...
END PERIOD
```

Explanation of Variables

PRINT_INPUT—keyword to indicate that the list of horizontal flow barriers will be written to the listing file immediately after it is read.

maxhfb—integer value specifying the maximum number of horizontal flow barriers that will be entered in this input file. The value of maxhfb is used to allocate memory for the horizontal flow barriers.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. iper must be less than nper in the TDIS Package and greater than zero. The iper value assigned to a stress period block must be greater than the iper value assigned for the previous block.

cellid1—identifier for the first cell. For a structured grid that uses the DIS input file, cellid1 is the layer, row, and column numbers of the cell. For a grid that uses the DISV input file, cellid1 is the layer number and cell2d number for the two cells. If the model uses the unstructured discretization (DISU) input file, then cellid1 is the node numbers for the cell. The barrier is located between cells designated as cellid1 and cellid2. For models that use the DIS and DISV grid types, the layer number for cellid1 and cellid2 must be the same. For all grid types, cells must be horizontally adjacent or the program will terminate with an error.

cellid2—identifier for the second cell. See cellid1 for description of how to specify.

hydchr—is the hydraulic characteristic of the horizontal-flow barrier. The hydraulic characteristic is the barrier hydraulic conductivity divided by the width of the horizontal-flow barrier. If hydraulic

characteristic is negative, then it acts as a multiplier to the conductance between the two model cells specified as containing a barrier. For example, if the value for `hydchr` was specified as 1.5, the conductance calculated for the two cells would be multiplied by 1.5.

Example Input File

```
BEGIN OPTIONS
  PRINT_INPUT
END OPTIONS

BEGIN DIMENSIONS
  MAXHFB 1
END DIMENSIONS

BEGIN PERIOD 1
  #L1 R1 C1 L2 R2 C2 HYDCHR
    1  1  4  1  1  5 0.1
END PERIOD 1
```

Storage (STO) Package

Input to the Storage (STO) Package is read from the file that has type "STO6" in the Name File. If the STO Package is not included for a model, then storage changes will not be calculated, and thus, the model will be steady state. Only one STO Package can be specified for a GWF model.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [SAVE_FLOWS]
  [STORAGECOEFFICIENT]
END OPTIONS

BEGIN GRIDDATA
  ICONVERT [LAYERED]
    <iconvert(nodes)> -- READARRAY
  SS [LAYERED]
    <ss(nodes)> -- READARRAY
  SY [LAYERED]
    <sy(nodes)> -- READARRAY
END GRIDDATA
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  [STEADY-STATE]
  [TRANSIENT]
END PERIOD
```

Explanation of Variables

SAVE_FLOWS—keyword to indicate that cell-by-cell flow terms will be written to the file specified with "BUDGET SAVE FILE" in Output Control.

STORAGECOEFFICIENT—keyword to indicate that the `ss` array is read as storage coefficient rather than specific storage.

iconvert—is a flag for each cell that specifies whether or not a cell is convertible for the storage calculation. 0 indicates confined storage is used. >0 indicates confined storage is used when head is above cell top and unconfined storage is used when head is below cell top. A mixed formulation is when when a cell converts from confined to unconfined (or vice versa) during a single time step.

ss—is specific storage (or the storage coefficient if STORAGECOEFFICIENT is specified as an option).

sy—is specific yield.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. `iper` must be less than `nper` in the TDIS Package and greater than zero.

The `iper` value assigned to a stress period block must be greater than the `iper` value assigned for the previous block.

`STEADY-STATE`—keyword to indicate that stress-period `iper` is steady-state. Steady-state conditions will apply until the `TRANSIENT` keyword is specified in a subsequent `BEGIN PERIOD` block.

`TRANSIENT`—keyword to indicate that stress-period `iper` is transient. Transient conditions will apply until the `STEADY-STATE` keyword is specified in a subsequent `BEGIN PERIOD` block.

Example Input File

```
BEGIN OPTIONS
  SAVE_FLOWS
END OPTIONS

BEGIN GRIDDATA
  #cell storage conversion 0:confined, 1:convertible
  ICONVERT
    constant 1
  #specific storage (for all model cells)
  SS
    constant 1.e-5
  #specific yield (specified by layer because of LAYERED keyword)
  SY LAYERED
    constant 0.2
    constant 0.15
    constant 0.15
END GRIDDATA

BEGIN PERIOD 1
  STEADY-STATE
END PERIOD

BEGIN PERIOD 2
  TRANSIENT
END PERIOD

#stress period 3 will be transient because
#a BEGIN PERIOD block is not provided.

BEGIN PERIOD 4
  STEADY-STATE
END PERIOD
```

Constant-Head (CHD) Package

Input to the Constant-Head (CHD) Package is read from the file that has type "CHD6" in the Name File. Any number of CHD Packages can be specified for a single groundwater flow model; however, an error will occur if a CHD Package attempts to make a GWF cell a constant-head cell when that cell has already been designated as a constant-head cell either within the present CHD Package or within another CHD Package. All single valued variables are free format.

In previous MODFLOW versions, it was not possible to convert a constant-head cell to an active cell. Once a cell was designated as a constant-head cell, it remained a constant-head cell until the end of the end of the simulation. In MODFLOW 6 a constant-head cell will become active again if it is not included as a constant-head cell in subsequent stress periods.

Previous MODFLOW versions allowed specification of SHEAD and EHEAD, which were the starting and ending prescribed heads for a stress period. Linear interpolation was used to calculate a value for each time step. In MODFLOW 6 only a single head value can be specified for any constant-head cell in any stress period. The time-series functionality must be used in order to interpolate values to individual time steps.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [AUXILIARY <auxiliary(naux)>]
  [AUXMULTNAME <auxmultname>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
END OPTIONS
```

```
BEGIN DIMENSIONS
  MAXBOUND <maxbound>
END DIMENSIONS
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <cellid(ncelldim)> <head> [<aux(naux)>] [<boundname>]
  <cellid(ncelldim)> <head> [<aux(naux)>] [<boundname>]
  ...
END PERIOD
```

Explanation of Variables

auxiliary—defines an array of one or more auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided on this line; however, lists of information provided in subsequent blocks must have a column of data for each auxiliary variable name defined here. The number of auxiliary variables detected on this line determines the value for naux. Comments

cannot be provided anywhere on this line as they will be interpreted as auxiliary variable names. Auxiliary variables may not be used by the package, but they will be available for use by other parts of the program. The program will terminate with an error if auxiliary variables are specified on more than one line in the options block.

auxmultname—name of auxiliary variable to be used as multiplier of CHD head value.

BOUNDNAMES—keyword to indicate that boundary names may be provided with the list of constant-head cells.

PRINT_INPUT—keyword to indicate that the list of constant-head information will be written to the listing file immediately after it is read.

PRINT_FLOWS—keyword to indicate that the list of constant-head flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and PRINT_FLOWS is specified, then flow rates are printed for the last time step of each stress period.

SAVE_FLOWS—keyword to indicate that constant-head flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

TS6—keyword to specify that record corresponds to a time-series file.

FILEIN—keyword to specify that an input filename is expected next.

ts6_filename—defines a time-series file defining time series that can be used to assign time-varying values. See the "Time-Variable Input" section for instructions on using the time-series capability.

OBS6—keyword to specify that record corresponds to an observations file.

obs6_filename—name of input file to define observations for the constant-head package. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the constant-head package.

maxbound—integer value specifying the maximum number of constant-head cells that will be specified for use during any stress period.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. iper must be less than nper in the TDIS Package and greater than zero. The iper value assigned to a stress period block must be greater than the iper value assigned for the previous block.

cellid—is the cell identifier, and depends on the type of grid that is used for the simulation. For a structured grid that uses the DIS input file, cellid is the layer, row, and column. For a grid that uses the DISV input file, cellid is the layer and cell2d number. If the model uses the unstructured discretization (DISU) input file, then cellid is the node number for the cell.

head—is the head at the boundary.

aux—represents the values of the auxiliary variables for each constant head. The values of auxiliary variables must be present for each constant head. The values must be specified in the order of the auxiliary variables specified in the OPTIONS block. If the package supports time series and the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section),

values can be obtained from a time series by entering the time-series name in place of a numeric value.

boundname—name of the constant head boundary cell. `boundname` is an ASCII character variable that can contain as many as 40 characters. If `boundname` contains spaces in it, then the entire name must be enclosed within single quotes.

Example Input File

```
#The OPTIONS block is optional
BEGIN OPTIONS
  AUXILIARY temperature
  BOUNDNAMES
  PRINT_INPUT
  PRINT_FLOWS
  SAVE_FLOWS
END OPTIONS

#The DIMENSIONS block is required
BEGIN DIMENSIONS
  MAXBOUND 2
END DIMENSIONS

#The following block of constant-head cells will be activated
#for stress period 1. This block will remain active throughout
#the simulation.

BEGIN PERIOD 1
#l r c head  temperature boundname
 1 1 2 100.  20.5        chd_1_2
 1 1 3 100.  20.4        chd_1_3
END PERIOD 1
```

Available observation types

CHD Package observations are limited to the simulated constant head flow rate (`chd`). The data required for the CHD Package observation type is defined in table 7. Negative and positive values for an observation represent a loss from and gain to the GWF model, respectively.

Table 7: Available CHD Package observation types

Model	Observation type	ID	ID2	Description
CHD	chd	cellid or boundname	–	Flow between the groundwater system and a constant-head boundary or a group of cells with constant-head boundaries.

Example Observation Input File

```
BEGIN OPTIONS
  DIGITS 8
  PRINT_INPUT
END OPTIONS

BEGIN SINGLE FILEOUT my_model.chd01.csv
# obsname obstype  ID
  chd_2_1 CHD       1 1 2
  chd_2_2 CHD       1 2 2
  chd_2_3 CHD       1 3 2
  chd_2_4 CHD       1 4 2
END SINGLE

BEGIN CONTINUOUS FILEOUT my_model.chd02.csv
# obsname    obstype  ID
  chd_3_flow CHD       CHD_1_3
END CONTINUOUS
```

Well (WEL) Package

Input to the Well (WEL) Package is read from the file that has type "WEL6" in the Name File. Any number of WEL Packages can be specified for a single groundwater flow model. All single valued variables are free format.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [AUXILIARY <auxiliary(naux)>]
  [AUXMULTNAME <auxmultname>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [AUTO_FLOW_REDUCE <auto_flow_reduce>]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
  [MOVER]
END OPTIONS

BEGIN DIMENSIONS
  MAXBOUND <maxbound>
END DIMENSIONS
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <cellid(ncelldim)> <q> [<aux(naux)>] [<boundname>]
  <cellid(ncelldim)> <q> [<aux(naux)>] [<boundname>]
  ...
END PERIOD
```

Explanation of Variables

auxiliary—defines an array of one or more auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided on this line; however, lists of information provided in subsequent blocks must have a column of data for each auxiliary variable name defined here. The number of auxiliary variables detected on this line determines the value for naux. Comments cannot be provided anywhere on this line as they will be interpreted as auxiliary variable names. Auxiliary variables may not be used by the package, but they will be available for use by other parts of the program. The program will terminate with an error if auxiliary variables are specified on more than one line in the options block.

auxmultname—name of auxiliary variable to be used as multiplier of well flow rate.

BOUNDNAMES—keyword to indicate that boundary names may be provided with the list of well cells.

PRINT_INPUT—keyword to indicate that the list of well information will be written to the listing file immediately after it is read.

PRINT_FLOWS—keyword to indicate that the list of well flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and PRINT_FLOWS is specified, then flow rates are printed for the last time step of each stress period.

SAVE_FLOWS—keyword to indicate that well flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

auto_flow_reduce—keyword and real value that defines the fraction of the cell thickness used as an interval for smoothly adjusting negative pumping rates to 0 in cells with head values less than or equal to the bottom of the cell. Negative pumping rates are adjusted to 0 or a smaller negative value when the head in the cell is equal to or less than the calculated interval above the cell bottom. auto_flow_reduce is set to 0.1 if the specified value is less than or equal to zero. By default, negative pumping rates are not reduced during a simulation.

TS6—keyword to specify that record corresponds to a time-series file.

FILEIN—keyword to specify that an input filename is expected next.

ts6_filename—defines a time-series file defining time series that can be used to assign time-varying values. See the "Time-Variable Input" section for instructions on using the time-series capability.

OBS6—keyword to specify that record corresponds to an observations file.

obs6_filename—name of input file to define observations for the Well package. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the Well package.

MOVER—keyword to indicate that this instance of the Well Package can be used with the Water Mover (MVR) Package. When the MOVER option is specified, additional memory is allocated within the package to store the available, provided, and received water.

maxbound—integer value specifying the maximum number of wells cells that will be specified for use during any stress period.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. iper must be less than nper in the TDIS Package and greater than zero. The iper value assigned to a stress period block must be greater than the iper value assigned for the previous block.

cellid—is the cell identifier, and depends on the type of grid that is used for the simulation. For a structured grid that uses the DIS input file, cellid is the layer, row, and column. For a grid that uses the DISV input file, cellid is the layer and cell2d number. If the model uses the unstructured discretization (DISU) input file, then cellid is the node number for the cell.

q—is the volumetric well rate. A positive value indicates recharge (injection) and a negative value indicates discharge (extraction). If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

aux—represents the values of the auxiliary variables for each well. The values of auxiliary variables must be present for each well. The values must be specified in the order of the auxiliary variables

specified in the OPTIONS block. If the package supports time series and the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

boundname—name of the well cell. `boundname` is an ASCII character variable that can contain as many as 40 characters. If `boundname` contains spaces in it, then the entire name must be enclosed within single quotes.

Example Input File

```
#The OPTIONS block is optional
BEGIN OPTIONS
  AUXILIARY depth screen_length
  BOUNDNAMES
  PRINT_INPUT
  PRINT_FLOWS
  SAVE_FLOWS
END OPTIONS

#The DIMENSIONS block is required
BEGIN DIMENSIONS
  MAXBOUND 5
END DIMENSIONS

#The following block of wells will be activated for stress periods
#2 and 3. No wells are present in stress period 1 due to an
#absence of a block for that period.
BEGIN PERIOD 2
    #layer row  col     Q    depth  screen_length boundname

         #wells 1 and 2
        7  102   17 -19000    275.9              17.6      CW_1
        9  192   44 -13000    280.0              24.0      CW_2

         #wells 3 through 5
        9  109   67 -24000    295.1              12.1      CW_3
       10   43   17 -12000    301.3               9.6      CW_4
       11   12   17 -17000    315.0              18.6      CW_5

END PERIOD

#Turn off all wells for stress period 4
BEGIN PERIOD 4
  #An empty block indicates that there are no wells.
END PERIOD

#For stress period 5, turn on wells 1 and 4,
#and add three wells that are grouped in a well field
BEGIN PERIOD 5
    #layer row  col     Q    depth  screen_length boundname
        7  102   17 -19000    275.9              17.6      CW_1
       10   43   17 -12000    301.3               9.6      CW_4

    #wells in well field
        5   27   50 -11000    190.0              20.0 well_field
```

```
      5   27   51  -10000    185.0              20.0 well_field
      5   28   50  -12000    187.3              15.0 well_field
END PERIOD

#Use a list of wells in file wells.dat for stress period 6 until
#the end of the simulation. Use the SFAC variable to multiply
#all pumping rates by 1000.
BEGIN PERIOD 6
  OPEN/CLOSE wells.dat SFAC 1000.
END PERIOD
```

Available observation types

Well Package observations include the simulated well rates (`wel`) and the well discharge that is available for the MVR package (`to-mvr`). The data required for each WEL Package observation type is defined in table 8. The sum of `wel` and `to-mvr` is equal to the simulated well discharge rate, which may be less than the specified `q` if the `AUTO_FLOW_REDUCE` option is enabled. Negative and positive values for an observation represent a loss from and gain to the GWF model, respectively.

Table 8: Available WEL Package observation types

Stress Package	Observation type	ID	ID2	Description
WEL	wel	cellid or boundname	–	Flow between the groundwater system and a well boundary or a group of well boundaries.
WEL	to-mvr	cellid or boundname	–	Well boundary discharge that is available for the MVR package for a well boundary or a group of well boundaries.

Example Observation Input File

```
BEGIN OPTIONS
  DIGITS 7
  PRINT_INPUT
END OPTIONS

BEGIN CONTINUOUS FILEOUT my_model.wel.obs.csv
# obsname         obstype ID
  wel-7-102-17    WEL     7  102  17
  wel-7-102-17    WEL     CW_1
  well-field      WEL     well_field
END CONTINUOUS
```

Drain (DRN) Package

Input to the Drain (DRN) Package is read from the file that has type "DRN6" in the Name File. Any number of DRN Packages can be specified for a single groundwater flow model. All single valued variables are free format.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [AUXILIARY <auxiliary(naux)>]
  [AUXMULTNAME <auxmultname>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
  [MOVER]
END OPTIONS

BEGIN DIMENSIONS
  MAXBOUND <maxbound>
END DIMENSIONS
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <cellid(ncelldim)> <elev> <cond> [<aux(naux)>] [<boundname>]
  <cellid(ncelldim)> <elev> <cond> [<aux(naux)>] [<boundname>]
  ...
END PERIOD
```

Explanation of Variables

auxiliary—defines an array of one or more auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided on this line; however, lists of information provided in subsequent blocks must have a column of data for each auxiliary variable name defined here. The number of auxiliary variables detected on this line determines the value for naux. Comments cannot be provided anywhere on this line as they will be interpreted as auxiliary variable names. Auxiliary variables may not be used by the package, but they will be available for use by other parts of the program. The program will terminate with an error if auxiliary variables are specified on more than one line in the options block.

auxmultname—name of auxiliary variable to be used as multiplier of drain conductance.

BOUNDNAMES—keyword to indicate that boundary names may be provided with the list of drain cells.

PRINT_INPUT—keyword to indicate that the list of drain information will be written to the listing file immediately after it is read.

PRINT_FLOWS—keyword to indicate that the list of drain flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and PRINT_FLOWS is specified, then flow rates are printed for the last time step of each stress period.

SAVE_FLOWS—keyword to indicate that drain flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

TS6—keyword to specify that record corresponds to a time-series file.

FILEIN—keyword to specify that an input filename is expected next.

ts6_filename—defines a time-series file defining time series that can be used to assign time-varying values. See the "Time-Variable Input" section for instructions on using the time-series capability.

OBS6—keyword to specify that record corresponds to an observations file.

obs6_filename—name of input file to define observations for the Drain package. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the Drain package.

MOVER—keyword to indicate that this instance of the Drain Package can be used with the Water Mover (MVR) Package. When the MOVER option is specified, additional memory is allocated within the package to store the available, provided, and received water.

maxbound—integer value specifying the maximum number of drains cells that will be specified for use during any stress period.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. iper must be less than nper in the TDIS Package and greater than zero. The iper value assigned to a stress period block must be greater than the iper value assigned for the previous block.

cellid—is the cell identifier, and depends on the type of grid that is used for the simulation. For a structured grid that uses the DIS input file, cellid is the layer, row, and column. For a grid that uses the DISV input file, cellid is the layer and cell2d number. If the model uses the unstructured discretization (DISU) input file, then cellid is the node number for the cell.

elev—is the elevation of the drain. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

cond—is the hydraulic conductance of the interface between the aquifer and the drain. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

aux—represents the values of the auxiliary variables for each drain. The values of auxiliary variables must be present for each drain. The values must be specified in the order of the auxiliary variables specified in the OPTIONS block. If the package supports time series and the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

boundname—name of the drain cell. `boundname` is an ASCII character variable that can contain as many as 40 characters. If `boundname` contains spaces in it, then the entire name must be enclosed within single quotes.

Example Input File

```
#The OPTIONS block is optional
BEGIN OPTIONS
  BOUNDNAMES
  PRINT_INPUT
  PRINT_FLOWS
  SAVE_FLOWS
END OPTIONS


#The DIMENSIONS block is required
BEGIN DIMENSIONS
  MAXBOUND 5
END DIMENSIONS


#The following block of drains will be activated for for the entire stress period
BEGIN PERIOD 1
  #node elevation conductance boundname
        73        10.2           1000.        my_drn
        76        10.2           1000.        my_drn
        79        10.2           1000.        my_drn
        80        10.2           1000.        my_drn
        81        10.2           1000.        my_drn
END PERIOD
```

Available observation types

Drain Package observations include the simulated drain rates (`drn`) and the drain discharge that is available for the MVR package (`to-mvr`). The data required for each DRN Package observation type is defined in table 9. The sum of `drn` and `to-mvr` is equal to the simulated drain discharge rate for a drain boundary or group of drain boundaries.

Table 9: Available DRN Package observation types

Model	Observation type	ID	ID2	Description
DRN	drn	cellid or boundname	–	Flow between the groundwater system and a drain boundary or group of drain boundaries.
DRN	to-mvr	cellid or boundname	–	Drain boundary discharge that is available for the MVR package for a drain boundary or a group of drain boundaries.

Example Observation Input File

```
BEGIN OPTIONS
  DIGITS 8
  PRINT_INPUT
END OPTIONS

BEGIN CONTINUOUS FILEOUT my_model.drn01.csv
# obsname     obstype  ID
  drn_73      DRN      73
  drn_79      DRN      79
END CONTINUOUS

BEGIN CONTINUOUS FILEOUT my_model.drn02.csv
# obsname     obstype  ID
  drn_80      DRN      80
  drn_all     DRN      my_drn
END CONTINUOUS
```

River (RIV) Package

Input to the River (RIV) Package is read from the file that has type "RIV6" in the Name File. Any number of RIV Packages can be specified for a single groundwater flow model. All single valued variables are free format.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [AUXILIARY <auxiliary(naux)>]
  [AUXMULTNAME <auxmultname>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
  [MOVER]
END OPTIONS

BEGIN DIMENSIONS
  MAXBOUND <maxbound>
END DIMENSIONS
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <cellid(ncelldim)> <stage> <cond> <rbot> [<aux(naux)>] [<boundname>]
  <cellid(ncelldim)> <stage> <cond> <rbot> [<aux(naux)>] [<boundname>]
  ...
END PERIOD
```

Explanation of Variables

auxiliary—defines an array of one or more auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided on this line; however, lists of information provided in subsequent blocks must have a column of data for each auxiliary variable name defined here. The number of auxiliary variables detected on this line determines the value for naux. Comments cannot be provided anywhere on this line as they will be interpreted as auxiliary variable names. Auxiliary variables may not be used by the package, but they will be available for use by other parts of the program. The program will terminate with an error if auxiliary variables are specified on more than one line in the options block.

auxmultname—name of auxiliary variable to be used as multiplier of riverbed conductance.

BOUNDNAMES—keyword to indicate that boundary names may be provided with the list of river cells.

PRINT_INPUT—keyword to indicate that the list of river information will be written to the listing file immediately after it is read.

PRINT FLOWS—keyword to indicate that the list of river flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and **PRINT FLOWS** is specified, then flow rates are printed for the last time step of each stress period.

SAVE FLOWS—keyword to indicate that river flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

TS6—keyword to specify that record corresponds to a time-series file.

FILEIN—keyword to specify that an input filename is expected next.

ts6 filename—defines a time-series file defining time series that can be used to assign time-varying values. See the "Time-Variable Input" section for instructions on using the time-series capability.

OBS6—keyword to specify that record corresponds to an observations file.

obs6 filename—name of input file to define observations for the River package. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the River package.

MOVER—keyword to indicate that this instance of the River Package can be used with the Water Mover (MVR) Package. When the **MOVER** option is specified, additional memory is allocated within the package to store the available, provided, and received water.

maxbound—integer value specifying the maximum number of rivers cells that will be specified for use during any stress period.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. **iper** must be less than **nper** in the TDIS Package and greater than zero. The **iper** value assigned to a stress period block must be greater than the **iper** value assigned for the previous block.

cellid—is the cell identifier, and depends on the type of grid that is used for the simulation. For a structured grid that uses the DIS input file, **cellid** is the layer, row, and column. For a grid that uses the DISV input file, **cellid** is the layer and cell2d number. If the model uses the unstructured discretization (DISU) input file, then **cellid** is the node number for the cell.

stage—is the head in the river. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

cond—is the riverbed hydraulic conductance. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

rbot—is the elevation of the bottom of the riverbed. If the Options block includes a TIMESERIES-FILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

aux—represents the values of the auxiliary variables for each river. The values of auxiliary variables must be present for each river. The values must be specified in the order of the auxiliary variables specified in the OPTIONS block. If the package supports time series and the Options block includes

a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

boundname—name of the river cell. `boundname` is an ASCII character variable that can contain as many as 40 characters. If `boundname` contains spaces in it, then the entire name must be enclosed within single quotes.

Example Input File

```
BEGIN OPTIONS
  PRINT_INPUT
  PRINT_FLOWS
  SAVE_FLOWS
  BOUNDNAMES
  TS8 FILEIN river_stages.ts
END OPTIONS

begin dimensions
  MAXBOUND 20
end dimensions

BEGIN PERIOD 1
# layer   row    col    stage         cond    rbot  BoundName
     1     3     1   river_stage_1   1001.   35.9
     1     4     2   river_stage_1   1002.   35.8
     1     5     3   river_stage_1   1003.   35.7
     1     5     4   river_stage_1   1004.   35.6
     1     6     5   river_stage_1   1005.   35.5
     1     6     6   river_stage_1   1006.   35.4 riv1_c6
     1     6     7   river_stage_1   1007.   35.3 riv1_c7
     1     5     8   river_stage_1   1008.   35.2
     1     5     9   river_stage_1   1009.   35.1
     1     5    10   river_stage_1   1010.   35.0
     1    10     1   river_stage_2   1001.   36.9 riv2_upper
     1     9     2   river_stage_2   1002.   36.8 riv2_upper
     1     8     3   river_stage_2   1003.   36.7 riv2_upper
     1     7     4   river_stage_2   1004.   36.6
     1     7     5   river_stage_2   1005.   36.5
     1     6     6   river_stage_2   1006.   36.4 riv2_c6
     1     6     7   river_stage_2   1007.   36.3 riv2_c7
     1     7     8   river_stage_2   1008.   36.2
     1     7     9   river_stage_2   1009.   36.1
     1     7    10   river_stage_2   1010.   36.0
END PERIOD
```

Available observation types

River Package observations include the simulated river flow rates (`riv`) and the river discharge that is available for the MVR package (`to-mvr`). The data required for each RIV Package observation type is defined in table 10. The sum of `riv` and `to-mvr` is equal to the simulated river flow rate. Negative and positive values for an observation represent a loss from and gain to the GWF model, respectively.

Table 10: Available RIV Package observation types

Stress Package	Observation type	ID	ID2	Description
RIV	riv	cellid or boundname	–	Flow between the groundwater system and a river boundary.
RIV	to-mvr	cellid or boundname	–	River boundary discharge that is available for the MVR package.

Example Observation Input File

```
BEGIN OPTIONS
  DIGITS 7
  PRINT_INPUT
END OPTIONS

BEGIN CONTINUOUS FILEOUT my_model.riv.csv
# obsname      type ID
  rv1-5-4      RIV  1    5    4
  rv1-6-5      RIV  1    6    5
  rv1-c7       RIV  riv1_c7     # flow at boundary "riv1_c7"
  rv2-7-4      RIV  1    7    4
  rv2-8-5      RIV  1    7    5
  rv2-9-6      RIV  1    6    6
END CONTINUOUS

BEGIN CONTINUOUS FILEOUT my_model.riv.flows.csv
# obsname      type ID
  rv1-3-1      RIV  1    3    1
  rv1-4-2      RIV  1    4    2
  rv1-5-3      RIV  1    5    3
  rv1-c6       RIV  riv1_c6
  rv2-upper    RIV  riv2_upper
END CONTINUOUS
```

General-Head Boundary (GHB) Package

Input to the General-Head Boundary (GHB) Package is read from the file that has type "GHB6" in the Name File. Any number of GHB Packages can be specified for a single groundwater flow model. All single valued variables are free format.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [AUXILIARY <auxiliary(naux)>]
  [AUXMULTNAME <auxmultname>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
  [MOVER]
END OPTIONS

BEGIN DIMENSIONS
  MAXBOUND <maxbound>
END DIMENSIONS
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <cellid(ncelldim)> <bhead> <cond> [<aux(naux)>] [<boundname>]
  <cellid(ncelldim)> <bhead> <cond> [<aux(naux)>] [<boundname>]
  ...
END PERIOD
```

Explanation of Variables

    `auxiliary`—defines an array of one or more auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided on this line; however, lists of information provided in subsequent blocks must have a column of data for each auxiliary variable name defined here. The number of auxiliary variables detected on this line determines the value for naux. Comments cannot be provided anywhere on this line as they will be interpreted as auxiliary variable names. Auxiliary variables may not be used by the package, but they will be available for use by other parts of the program. The program will terminate with an error if auxiliary variables are specified on more than one line in the options block.

    `auxmultname`—name of auxiliary variable to be used as multiplier of general-head boundary conductance.

    `BOUNDNAMES`—keyword to indicate that boundary names may be provided with the list of general-head boundary cells.

PRINT_INPUT—keyword to indicate that the list of general-head boundary information will be written to the listing file immediately after it is read.

PRINT_FLOWS—keyword to indicate that the list of general-head boundary flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and PRINT_FLOWS is specified, then flow rates are printed for the last time step of each stress period.

SAVE_FLOWS—keyword to indicate that general-head boundary flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

TS6—keyword to specify that record corresponds to a time-series file.

FILEIN—keyword to specify that an input filename is expected next.

ts6_filename—defines a time-series file defining time series that can be used to assign time-varying values. See the "Time-Variable Input" section for instructions on using the time-series capability.

OBS6—keyword to specify that record corresponds to an observations file.

obs6_filename—name of input file to define observations for the General-Head Boundary package. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the General-Head Boundary package.

MOVER—keyword to indicate that this instance of the General-Head Boundary Package can be used with the Water Mover (MVR) Package. When the MOVER option is specified, additional memory is allocated within the package to store the available, provided, and received water.

maxbound—integer value specifying the maximum number of general-head boundary cells that will be specified for use during any stress period.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. iper must be less than nper in the TDIS Package and greater than zero. The iper value assigned to a stress period block must be greater than the iper value assigned for the previous block.

cellid—is the cell identifier, and depends on the type of grid that is used for the simulation. For a structured grid that uses the DIS input file, cellid is the layer, row, and column. For a grid that uses the DISV input file, cellid is the layer and cell2d number. If the model uses the unstructured discretization (DISU) input file, then cellid is the node number for the cell.

bhead—is the boundary head. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

cond—is the hydraulic conductance of the interface between the aquifer cell and the boundary. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

aux—represents the values of the auxiliary variables for each general-head boundary. The values of auxiliary variables must be present for each general-head boundary. The values must be specified in the order of the auxiliary variables specified in the OPTIONS block. If the package supports

> time series and the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

boundname—name of the general-head boundary cell. `boundname` is an ASCII character variable that can contain as many as 40 characters. If `boundname` contains spaces in it, then the entire name must be enclosed within single quotes.

Example Input File

```
BEGIN OPTIONS
  PRINT_INPUT (echo input to listing file)
  PRINT_FLOWS (print the flows to the listing file)
  TS8 FILEIN tides.ts
  BOUNDNAMES
END OPTIONS

# Dimensions block
BEGIN DIMENSIONS
  MAXBOUND 15
END DIMENSIONS

# Stress-period block(s)
BEGIN PERIOD 1
#Lay Row Col  Bhead Cond   boundname
  2   1   10   tides  15.0   Estuary-L2
  2   2   10   tides  15.0   Estuary-L2
  2   3   10   tides  15.0   Estuary-L2
  2   4   10   tides  15.0   Estuary-L2
  2   5   10   tides  15.0   Estuary-L2
  2   6   10   tides  15.0   Estuary-L2
  2   7   10   tides  15.0   Estuary-L2
  2   8   10   tides  15.0   Estuary-L2
  2   9   10   tides  15.0   Estuary-L2
  2  10   10   tides  15.0   Estuary-L2
  2  11   10   tides  15.0   Estuary-L2
  2  12   10   tides  15.0   Estuary-L2
  2  13   10   tides  15.0   Estuary-L2
  2  14   10   tides  15.0   Estuary-L2
  2  15   10   tides  15.0   Estuary-L2
END PERIOD
```

Available observation types

General-Head Boundary Package observations include the simulated general-head boundary flow rates (`ghb`) and the general-head boundary discharge that is available for the MVR package (`to-mvr`). The data required for each GHB Package observation type is defined in table 11. The sum of `ghb` and `to-mvr` is equal to the simulated general-head boundary flow rate. Negative and positive values for an observation represent a loss from and gain to the GWF model, respectively.

Table 11: Available GHB Package observation types

Stress Package	Observation type	ID	ID2	Description
GHB	ghb	cellid or boundname	–	Flow between the groundwater system and a general-head boundary or group of general-head boundaries.
GHB	to-mvr	cellid or boundname	–	General-head boundary discharge that is available for the MVR package from a general-head boundary or group of general-head boundaries.

Example Observation Input File

```
BEGIN OPTIONS
  DIGITS 7
  PRINT_INPUT
END OPTIONS

BEGIN CONTINUOUS FILEOUT my_model.ghb.obs.csv
# obsname     obstype ID
  ghb-2-6-10  GHB     2   6   10
  ghb-2-7-10  GHB     2   7   10
END CONTINUOUS

BEGIN CONTINUOUS FILEOUT my_model.ghb.flows.csv
# obsname       obstype ID
  Estuary2      GHB     Estuary-L2
END CONTINUOUS
```

Recharge (RCH) Package – List-Based Input

Input to the Recharge (RCH) Package is read from the file that has type "RCH6" in the Name File. Any number of RCH Packages can be specified for a single groundwater flow model. All single valued variables are free format.

Recharge input can be specified using lists or arrays, unless the DISU Package is used. List-based input must be used if discretization is specified using the DISU Package. List-based input for recharge is the default, and is described here. Instructions for specifying array-based recharge are described in the next section.

List-based input offers several advantages over the array-based input for specifying recharge. First, multiple list entries can be specified for a single cell. This makes it possible to divide a cell into multiple areas, and assign a different recharge rate for each area (perhaps based on land use or some other criteria). In this case, the user would likely specify an auxiliary variable to serve as a multiplier. This multiplier would be calculated by the user and provided in the input file as the fractional cell are for the individual recharge entries. Another advantage to using list-based input for specifying recharge is that boundnames can be specified. Boundnames work with the Observations capability and can be used to sum recharge rates for entries with the same boundname. A disadvantage of the list-based input is that one cannot easily assign recharge to the entire model without specifying a list of model cells. For this reason MODFLOW 6 also supports array-based input for recharge.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [FIXED_CELL]
  [AUXILIARY <auxiliary(naux)>]
  [AUXMULTNAME <auxmultname>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
END OPTIONS


BEGIN DIMENSIONS
  MAXBOUND <maxbound>
END DIMENSIONS
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <cellid(ncelldim)> <recharge> [<aux(naux)>] [<boundname>]
  <cellid(ncelldim)> <recharge> [<aux(naux)>] [<boundname>]
  ...
END PERIOD
```

Explanation of Variables

FIXED_CELL—indicates that recharge will not be reassigned to a cell underlying the cell specified in the list if the specified cell is inactive.

auxiliary—defines an array of one or more auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided on this line; however, lists of information provided in subsequent blocks must have a column of data for each auxiliary variable name defined here. The number of auxiliary variables detected on this line determines the value for naux. Comments cannot be provided anywhere on this line as they will be interpreted as auxiliary variable names. Auxiliary variables may not be used by the package, but they will be available for use by other parts of the program. The program will terminate with an error if auxiliary variables are specified on more than one line in the options block.

auxmultname—name of auxiliary variable to be used as multiplier of recharge.

BOUNDNAMES—keyword to indicate that boundary names may be provided with the list of recharge cells.

PRINT_INPUT—keyword to indicate that the list of recharge information will be written to the listing file immediately after it is read.

PRINT_FLOWS—keyword to indicate that the list of recharge flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and PRINT_FLOWS is specified, then flow rates are printed for the last time step of each stress period.

SAVE_FLOWS—keyword to indicate that recharge flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

TS6—keyword to specify that record corresponds to a time-series file.

FILEIN—keyword to specify that an input filename is expected next.

ts6_filename—defines a time-series file defining time series that can be used to assign time-varying values. See the "Time-Variable Input" section for instructions on using the time-series capability.

OBS6—keyword to specify that record corresponds to an observations file.

obs6_filename—name of input file to define observations for the Recharge package. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the Recharge package.

maxbound—integer value specifying the maximum number of recharge cells cells that will be specified for use during any stress period.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. iper must be less than nper in the TDIS Package and greater than zero. The iper value assigned to a stress period block must be greater than the iper value assigned for the previous block.

cellid—is the cell identifier, and depends on the type of grid that is used for the simulation. For a structured grid that uses the DIS input file, `cellid` is the layer, row, and column. For a grid that uses the DISV input file, `cellid` is the layer and cell2d number. If the model uses the unstructured discretization (DISU) input file, then `cellid` is the node number for the cell.

recharge—is the recharge flux rate (LT^{-1}). This rate is multiplied inside the program by the surface area of the cell to calculate the volumetric recharge rate. A time-series name may be specified.

aux—represents the values of the auxiliary variables for each recharge. The values of auxiliary variables must be present for each recharge. The values must be specified in the order of the auxiliary variables specified in the OPTIONS block. If the package supports time series and the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

boundname—name of the recharge cell. `boundname` is an ASCII character variable that can contain as many as 40 characters. If `boundname` contains spaces in it, then the entire name must be enclosed within single quotes.

Example Input File

```
BEGIN OPTIONS
  AUXILIARY var1 var2 mult
  PRINT_INPUT
  PRINT_FLOWS
  SAVE_FLOWS
  BOUNDNAMES
  TS8 FILEIN recharge_rates.ts
  # Note: Time-series file recharge_rates.ts defines time series rch_1
  AUXMULTNAME mult
END OPTIONS

BEGIN DIMENSIONS
  MAXBOUND 10
END DIMENSIONS

BEGIN PERIOD 1
  RECHARGE
  # Lay Row Col  Rate   Var1  Var2   mult  BoundName
     1   1   1   rch_1  1.0   2.0    1.0      Rch-1-1
     1   1   2   rch_1  1.1   2.1    1.0      Rch-1-2
     1   1   3   rch_1  1.2   2.2    0.5
     1   2   1   rch_1  1.3   2.3    1.0      Rch-2-1
     1   2   2   rch_1  1.4   2.4    1.0      Rch-2-2
     1   2   3   rch_1  1.5   2.5    1.0
     1   2   4   rch_1  1.6   2.6    0.5
     1   3   1   rch_1  1.7   2.7    1.0
     1   3   2   rch_1  1.8   2.8    1.0
     1   3   3   rch_1  1.9   2.9    1.0
END PERIOD
```

Available observation types

Recharge Package observations are limited to the simulated recharge flow rates (`rch`). The data required for the RCH Package observation type is defined in table 12. Negative and positive values for an observation represent a loss from and gain to the GWF model, respectively.

Table 12: Available RCH Package observation types

Stress Package	Observation type	ID	ID2	Description
RCH	rch	cellid or boundname	–	Flow to the groundwater system through a recharge boundary or a group of recharge boundaries.

Example Observation Input File

```
BEGIN OPTIONS
  PRINT_INPUT
END OPTIONS

BEGIN CONTINUOUS FILEOUT my_model.rch.csv
  rch1-1   RCH   Rch-1-1
  rch1-2   RCH   Rch-1-2
  rch2-1   RCH   Rch-2-1
  rch2-2   RCH   Rch-2-2
  rch2-3   RCH   1  2  3
END CONTINUOUS
```

Recharge (RCH) Package – Array-Based Input

Input to the Recharge (RCH) Package is read from the file that has type "RCH6" in the Name File. Any number of RCH Packages can be specified for a single groundwater flow model. All single valued variables are free format.

Recharge input can be specified using lists or arrays. Array-based input for recharge is activated by providing READASARRAYS within the OPTIONS block. Instructions for specifying list-based recharge is described in the previous section. Array-based input for recharge provides a similar approach for providing recharge rates as previous MODFLOW versions. Array-based input for recharge can be used only with the DIS and DISV Packages. Array-based input for recharge cannot be used with the DISU Package.

When array-based input is used for recharge, the DIMENSIONS block should not be specified. The array size is determined from the model grid.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  READASARRAYS
  [FIXED_CELL]
  [AUXILIARY <auxiliary(naux)>]
  [AUXMULTNAME <auxmultname>]
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [TAS6 FILEIN <tas6_filename>]
  [OBS6 FILEIN <obs6_filename>]
END OPTIONS
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  [IRCH
    <irch(ncol*nrow; ncpl)> -- U1DDBL]
  RECHARGE
    <recharge(ncol*nrow; ncpl)> -- U1DDBL
  [AUX
    <aux(ncol*nrow; ncpl)> -- U1DDBL]
END PERIOD
```

Explanation of Variables

READASARRAYS—indicates that array-based input will be used for the Recharge Package. This keyword must be specified to use array-based input.

FIXED_CELL—indicates that recharge will not be reassigned to a cell underlying the cell specified in the list if the specified cell is inactive.

auxiliary—defines an array of one or more auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided on this line; however, lists of information provided in

subsequent blocks must have a column of data for each auxiliary variable name defined here. The number of auxiliary variables detected on this line determines the value for naux. Comments cannot be provided anywhere on this line as they will be interpreted as auxiliary variable names. Auxiliary variables may not be used by the package, but they will be available for use by other parts of the program. The program will terminate with an error if auxiliary variables are specified on more than one line in the options block.

`auxmultname`—name of auxiliary variable to be used as multiplier of recharge.

`PRINT_INPUT`—keyword to indicate that the list of recharge information will be written to the listing file immediately after it is read.

`PRINT_FLOWS`—keyword to indicate that the list of recharge flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and `PRINT_FLOWS` is specified, then flow rates are printed for the last time step of each stress period.

`SAVE_FLOWS`—keyword to indicate that recharge flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

`TAS6`—keyword to specify that record corresponds to a time-array-series file.

`FILEIN`—keyword to specify that an input filename is expected next.

`tas6_filename`—defines a time-array-series file defining a time-array series that can be used to assign time-varying values. See the Time-Variable Input section for instructions on using the time-array series capability.

`OBS6`—keyword to specify that record corresponds to an observations file.

`obs6_filename`—name of input file to define observations for the Recharge package. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the Recharge package.

`iper`—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. `iper` must be less than `nper` in the TDIS Package and greater than zero. The `iper` value assigned to a stress period block must be greater than the `iper` value assigned for the previous block.

`irch`—`irch` is the layer number that defines the layer in each vertical column where recharge is applied. If `irch` is omitted, recharge by default is applied to cells in layer 1. `irch` can only be used if READASARRAYS is specified in the OPTIONS block.

`recharge`—is the recharge flux rate (LT^{-1}). This rate is multiplied inside the program by the surface area of the cell to calculate the volumetric recharge rate. The recharge array may be defined by a time-array series (see the "Using Time-Array Series in a Package" section).

`aux`—is an array of values for auxiliary variable aux(iaux), where iaux is a value from 1 to naux, and aux(iaux) must be listed as part of the auxiliary variables. A separate array can be specified for each auxiliary variable. If an array is not specified for an auxiliary variable, then a value of zero is assigned. If the value specified here for the auxiliary variable is the same as auxmultname, then the recharge array will be multiplied by this array.

Example Input File

```
BEGIN OPTIONS
  AUXILIARY var1 var2 mymult
  READASARRAYS
  HIGHEST_ACTIVE
  PRINT_INPUT
  PRINT_FLOWS
  SAVE_FLOWS
  AUXMULTNAME mymult
END OPTIONS

BEGIN PERIOD 1

  # For this model, the absence of an IRCH array causes
  # recharge to apply to model layer 1. To make recharge
  # apply to layer 2 instead, the following lines
  # (uncommented) could be used:
  # IRCH
  #   constant 2

  # recharge rate
  RECHARGE
    constant 0.0040

  # auxiliary variable (var1) array
  var1
    constant 100.

  # auxiliary variable (var2) array
  var2
    constant 0.

  # auxiliary variable (mymult) array
  # Because ``AUXMULTNAME mymult'' was specified in the
  # options block, the MYMULT array will be used to multiply
  # the values in the RECHARGE array
  MYMULT
    INTERNAL FACTOR 1.0
      0.5  1.0  1.0  0.5
      1.0  1.0  1.0  1.0
      0.5  1.0  1.0  0.5

END PERIOD
```

Evapotranspiration (EVT) Package – List-Based Input

Input to the Evapotranspiration (EVT) Package is read from the file that has type "EVT6" in the Name File. Any number of EVT Packages can be specified for a single groundwater flow model. All single-valued variables are free format.

Evapotranspiration input can be specified using lists or arrays, unless the DISU Package is used. List-based input must be used if discretization is specified using the DISU Package. List-based input for recharge is the default, and is described here. Instructions for specifying array-based recharge are described in the next section.

List-based input offers several advantages over the array-based input for specifying evapotranspiration. First, multiple list entries can be specified for a single cell. This makes it possible to divide a cell into multiple areas, and assign a different evapotranspiration rate or extinction depth for each area (perhaps based on vegetation type or some other criteria). In this case, the user would likely specify an auxiliary variable to serve as a multiplier. This multiplier would be calculated by the user and provided in the input file as the fractional cell are for the individual evapotranspiration entries. Another advantage to using list-based input for specifying evapotranspiration is that boundnames can be specified. Boundnames work with the Observations capability and can be used to sum evapotranspiration rates for entries with the same boundname. A disadvantage of the list-based input is that one cannot easily assign evapotranspiration to the entire model without specifying a list of model cells. For this reason MODFLOW 6 also supports array-based input for evapotranspiration.

ET input is read in list form, as shown in the PERIOD block below. Each line in the PERIOD block defines all input for one cell. Entries following `cellid`, in order, define the ET surface (`etss`), maximum ET flux rate (`etsr`), extinction depth (`etsx`), all (`netseg` – 1) `pxdp` values, all (`netseg` – 1) `petm` values, all auxiliary variables (if AUXILIARY option is specified), and boundary name (if BOUNDNAMES option is specified).

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [FIXED_CELL]
  [AUXILIARY <auxiliary(naux)>]
  [AUXMULTNAME <auxmultname>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
  [SURF_RATE_SPECIFIED]
END OPTIONS
```

```
BEGIN DIMENSIONS
  MAXBOUND <maxbound>
  NSEG <nseg>
END DIMENSIONS
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <cellid(ncelldim)> <surface> <rate> <depth> <pxdp(nseg-1)> <petm(nseg-1)> [<petm0>] [<aux(naux)>] [<boundname>]
  <cellid(ncelldim)> <surface> <rate> <depth> <pxdp(nseg-1)> <petm(nseg-1)> [<petm0>] [<aux(naux)>] [<boundname>]
  ...
END PERIOD
```

Explanation of Variables

FIXED_CELL—indicates that evapotranspiration will not be reassigned to a cell underlying the cell specified in the list if the specified cell is inactive.

auxiliary—defines an array of one or more auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided on this line; however, lists of information provided in subsequent blocks must have a column of data for each auxiliary variable name defined here. The number of auxiliary variables detected on this line determines the value for naux. Comments cannot be provided anywhere on this line as they will be interpreted as auxiliary variable names. Auxiliary variables may not be used by the package, but they will be available for use by other parts of the program. The program will terminate with an error if auxiliary variables are specified on more than one line in the options block.

auxmultname—name of auxiliary variable to be used as multiplier of evapotranspiration rate.

BOUNDNAMES—keyword to indicate that boundary names may be provided with the list of evapotranspiration cells.

PRINT_INPUT—keyword to indicate that the list of evapotranspiration information will be written to the listing file immediately after it is read.

PRINT_FLOWS—keyword to indicate that the list of evapotranspiration flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and PRINT_FLOWS is specified, then flow rates are printed for the last time step of each stress period.

SAVE_FLOWS—keyword to indicate that evapotranspiration flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

TS6—keyword to specify that record corresponds to a time-series file.

FILEIN—keyword to specify that an input filename is expected next.

ts6_filename—defines a time-series file defining time series that can be used to assign time-varying values. See the "Time-Variable Input" section for instructions on using the time-series capability.

OBS6—keyword to specify that record corresponds to an observations file.

obs6_filename—name of input file to define observations for the Evapotranspiration package. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the Evapotranspiration package.

SURF_RATE_SPECIFIED—indicates that the evapotranspiration rate at the ET surface will be specified as PETM0 in list input.

maxbound—integer value specifying the maximum number of evapotranspiration cells cells that will be specified for use during any stress period.

nseg—number of ET segments. Default is one. When nseg is greater than 1, pxdp and petm arrays must be specified nseg-1 times each, in order from the uppermost segment down. pxdp defines the extinction-depth proportion at the bottom of a segment. petm defines the proportion of the maximum ET flux rate at the bottom of a segment.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. iper must be less than nper in the TDIS Package and greater than zero. The iper value assigned to a stress period block must be greater than the iper value assigned for the previous block.

cellid—is the cell identifier, and depends on the type of grid that is used for the simulation. For a structured grid that uses the DIS input file, cellid is the layer, row, and column. For a grid that uses the DISV input file, cellid is the layer and cell2d number. If the model uses the unstructured discretization (DISU) input file, then cellid is the node number for the cell.

surface—is the elevation of the ET surface (L). A time-series name may be specified.

rate—is the maximum ET flux rate (LT^{-1}). A time-series name may be specified.

depth—is the ET extinction depth (L). A time-series name may be specified.

pxdp—is the proportion of the ET extinction depth at the bottom of a segment (dimensionless). A time-series name may be specified.

petm—is the proportion of the maximum ET flux rate at the bottom of a segment (dimensionless). A time-series name may be specified.

petm0—is the proportion of the maximum ET flux rate that will apply when head is at or above the ET surface (dimensionless). PETM0 is read only when the SURF_RATE_SPECIFIED option is used. A time-series name may be specified.

aux—represents the values of the auxiliary variables for each evapotranspiration. The values of auxiliary variables must be present for each evapotranspiration. The values must be specified in the order of the auxiliary variables specified in the OPTIONS block. If the package supports time series and the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

boundname—name of the evapotranspiration cell. boundname is an ASCII character variable that can contain as many as 40 characters. If boundname contains spaces in it, then the entire name must be enclosed within single quotes.

Example Input File

```
# Example for structured model with list-based input
BEGIN OPTIONS
  AUXNAMES Mult
  BOUNDNAMES
  TS8 FILEIN EtRate.ts
  # Note: Time-series file EtRate.ts defines time series et_rate
  AUXMULTNAME Mult
PRINT_INPUT

BEGIN DIMENSIONS
  MAXBOUND 10
END DIMENSIONS

BEGIN PERIOD 1
# Lay Row Col SURFACE RATE DEPTH PXPD1 PXPD2 PETM1 PETM2 Mult Name
    1   1  13  110.0  et_rate 10.0 0.2   0.5   0.3   0.1   0.2  ET-1
    1   2  13  110.0  et_rate 10.0 0.2   0.5   0.3   0.1   0.4  ET-2
    1   3  13  110.0  et_rate 10.0 0.2   0.5   0.3   0.1   0.6  ET-3
    1   4  13  110.0  et_rate 10.0 0.2   0.5   0.3   0.1   0.8  ET-4
    1   5  13  110.0  2.e-2   10.0 0.2   0.5   0.3   0.1   1.0  ET-5
    1   6  13  110.0  2.e-2   10.0 0.2   0.5   0.3   0.1   1.0  ET-6
    1   7  13  110.0  2.e-2   10.0 0.2   0.5   0.3   0.1   0.7  ET-7
    1   8  13  110.0  2.e-2   10.0 0.2   0.5   0.3   0.1   0.5  ET-8
    1   9  13  110.0  2.e-2   10.0 0.2   0.5   0.3   0.1   0.3  ET-9
    1  10  13  110.0  et_rate 10.0 0.2   0.5   0.3   0.1   0.1  ET-10
END PERIOD
```

Available observation types

Evapotranspiration Package observations are limited to the simulated evapotranspiration flow rates (`evt`). The data required for the EVT Package observation type is defined in table 13. Negative and positive values for an observation represent a loss from and gain to the GWF model, respectively.

Table 13: Available EVT Package observation types

Stress Package	Observation type	ID	ID2	Description
EVT	evt	cellid or boundname	–	Flow from the groundwater system through an evapotranspiration boundary or group of evapotranspiration boundaries.

Example Observation Input File

```
BEGIN OPTIONS
  PRINT_INPUT
END OPTIONS

BEGIN CONTINUOUS FILEOUT my_model.evt.csv
  et1-1   EVT   1 1 1
  et1-2   EVT   1 1 2
  et2-1   EVT   1 2 1
```

```
  et2-2    EVT   1  2  2
  et2-3    EVT   1  2  3
END CONTINUOUS
```

Evapotranspiration (EVT) Package – Array-Based Input

Input to the Evapotranspiration (EVT) Package is read from the file that has type "EVT6" in the Name File. Any number of EVT Packages can be specified for a single groundwater flow model. All single-valued variables are free format.

Evapotranspiration input can be specified using lists or arrays. Array-based input for evapotranspiration is activated by providing READASARRAYS within the OPTIONS block. Instructions for specifying list-based evapotranspiration is described in the previous section. Array-based input for evapotranspiration provides a similar approach for providing evapotranspiration rates as previous MODFLOW versions. Array-based input for evapotranspiration can be used only with the DIS and DISV Packages. Array-based input for evapotranspiration cannot be used with the DISU Package.

When array-based input is used for evapotranspiration, the DIMENSIONS block should not be specified. The array size is determined from the model grid. Segmented evapotranspiration cannot be used with the READASARRAYS option.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  READASARRAYS
  [FIXED_CELL]
  [AUXILIARY <auxiliary(naux)>]
  [AUXMULTNAME <auxmultname>]
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [TAS6 FILEIN <tas6_filename>]
  [OBS6 FILEIN <obs6_filename>]
END OPTIONS
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  [IEVT
    <ievt(ncol*nrow; ncpl)> -- U1DDBL]
  SURFACE
    <surface(ncol*nrow; ncpl)> -- U1DDBL
  RATE
    <rate(ncol*nrow; ncpl)> -- U1DDBL
  DEPTH
    <depth(ncol*nrow; ncpl)> -- U1DDBL
  AUX(IAUX)
    <aux(iaux)(ncol*nrow; ncpl)> -- U1DDBL
END PERIOD
```

Explanation of Variables

READASARRAYS—indicates that array-based input will be used for the Evapotranspiration Package. This keyword must be specified to use array-based input.

FIXED_CELL—indicates that evapotranspiration will not be reassigned to a cell underlying the cell specified in the list if the specified cell is inactive.

auxiliary—defines an array of one or more auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided on this line; however, lists of information provided in subsequent blocks must have a column of data for each auxiliary variable name defined here. The number of auxiliary variables detected on this line determines the value for naux. Comments cannot be provided anywhere on this line as they will be interpreted as auxiliary variable names. Auxiliary variables may not be used by the package, but they will be available for use by other parts of the program. The program will terminate with an error if auxiliary variables are specified on more than one line in the options block.

auxmultname—name of auxiliary variable to be used as multiplier of evapotranspiration rate.

PRINT_INPUT—keyword to indicate that the list of evapotranspiration information will be written to the listing file immediately after it is read.

PRINT_FLOWS—keyword to indicate that the list of evapotranspiration flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and PRINT_FLOWS is specified, then flow rates are printed for the last time step of each stress period.

SAVE_FLOWS—keyword to indicate that evapotranspiration flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

TAS6—keyword to specify that record corresponds to a time-array-series file.

FILEIN—keyword to specify that an input filename is expected next.

tas6_filename—defines a time-array-series file defining a time-array series that can be used to assign time-varying values. See the Time-Variable Input section for instructions on using the time-array series capability.

OBS6—keyword to specify that record corresponds to an observations file.

obs6_filename—name of input file to define observations for the Evapotranspiration package. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the Evapotranspiration package.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. iper must be less than nper in the TDIS Package and greater than zero. The iper value assigned to a stress period block must be greater than the iper value assigned for the previous block.

ievt—ievt is the layer number that defines the layer in each vertical column where evapotranspiration is applied. If ievt is omitted, evapotranspiration by default is applied to cells in layer 1.

surface—is the elevation of the ET surface (L).

rate—is the maximum ET flux rate (LT^{-1}).

depth—is the ET extinction depth (L).

aux(iaux)—is an array of values for auxiliary variable aux(iaux), where iaux is a value from 1 to naux, and aux(iaux) must be listed as part of the auxiliary variables. A separate array can be specified for each auxiliary variable. If an array is not specified for an auxiliary variable, then a value of zero is assigned. If the value specified here for the auxiliary variable is the same as auxmultname, then the evapotranspiration rate will be multiplied by this array.

Example Input File

```
BEGIN OPTIONS
  READASARRAYS
  AUXILIARY var1 var2
  HIGHEST_ACTIVE
  PRINT_INPUT
  PRINT_FLOWS
  SAVE_FLOWS
END OPTIONS

BEGIN PERIOD 1
  #For a structured grid, IEVT defaults to model
  # layer 1, so no need to enter IEVT here.

  #ET surface elevation
  SURFACE
    constant 150.0
  #Maximum ET rate
  RATE
    constant 0.007
  #ET extinction depth
  DEPTH
    constant 15.0
  #auxiliary variable (var1) array
  var1
    constant 100.0
  #auxiliary variable (var2) array
  var2
    constant 0.0
END PERIOD
```

Multi-Aquifer Well (MAW) Package

Input to the Multi-Aquifer Well (MAW) Package is read from the file that has type "MAW6" in the Name File. Any number of MAW Packages can be specified for a single groundwater flow model. All single valued variables are free format.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [AUXILIARY <auxiliary(naux)>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_HEAD]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [HEAD FILEOUT <headfile>]
  [BUDGET FILEOUT <budgetfile>]
  [NO_WELL_STORAGE]
  [FLOWING_WELLS]
  [SHUTDOWN_THETA <shutdown_theta>]
  [SHUTDOWN_KAPPA <shutdown_kappa>]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
  [MOVER]
END OPTIONS
```

```
BEGIN DIMENSIONS
  NMAWWELLS <nmawwells>
END DIMENSIONS
```

```
BEGIN PACKAGEDATA
  <wellno> <radius> <bottom> <strt> <condeqn> <ngwfnodes> [<aux(naux)>] [<boundname>]
  <wellno> <radius> <bottom> <strt> <condeqn> <ngwfnodes> [<aux(naux)>] [<boundname>]
  ...
END PACKAGEDATA
```

```
BEGIN CONNECTIONDATA
  <wellno> <icon> <cellid(ncelldim)> <scrn_top> <scrn_bot> <hk_skin> <radius_skin>
  <wellno> <icon> <cellid(ncelldim)> <scrn_top> <scrn_bot> <hk_skin> <radius_skin>
  ...
END CONNECTIONDATA
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <wellno> <mawsetting>
  <wellno> <mawsetting>
  ...
END PERIOD
```

Explanation of Variables

auxiliary—defines an array of one or more auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided on this line; however, lists of information provided in subsequent blocks must have a column of data for each auxiliary variable name defined here. The number of auxiliary variables detected on this line determines the value for naux. Comments cannot be provided anywhere on this line as they will be interpreted as auxiliary variable names. Auxiliary variables may not be used by the package, but they will be available for use by other parts of the program. The program will terminate with an error if auxiliary variables are specified on more than one line in the options block.

BOUNDNAMES—keyword to indicate that boundary names may be provided with the list of multi-aquifer well cells.

PRINT_INPUT—keyword to indicate that the list of multi-aquifer well information will be written to the listing file immediately after it is read.

PRINT_HEAD—keyword to indicate that the list of multi-aquifer well heads will be printed to the listing file for every stress period in which "HEAD PRINT" is specified in Output Control. If there is no Output Control option and **PRINT_HEAD** is specified, then heads are printed for the last time step of each stress period.

PRINT_FLOWS—keyword to indicate that the list of multi-aquifer well flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and **PRINT_FLOWS** is specified, then flow rates are printed for the last time step of each stress period.

SAVE_FLOWS—keyword to indicate that multi-aquifer well flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

HEAD—keyword to specify that record corresponds to head.

headfile—name of the binary output file to write stage information.

BUDGET—keyword to specify that record corresponds to the budget.

FILEOUT—keyword to specify that an output filename is expected next.

budgetfile—name of the binary output file to write budget information.

NO_WELL_STORAGE—keyword that deactivates inclusion of well storage contributions to the multi-aquifer well package continuity equation.

FLOWING_WELLS—keyword that activates the flowing wells option for the multi-aquifer well package.

shutdown_theta—value that defines the weight applied to discharge rate for wells that limit the water level in a discharging well (defined using the **HEAD_LIMIT** keyword in the stress period data). **SHUTDOWN_THETA** is used to control discharge rate oscillations when the flow rate from the aquifer is less than the specified flow rate from the aquifer to the well. Values range between 0.0 and 1.0, and larger values increase the weight (decrease under-relaxation) applied to the well discharge rate. The **head_limit** option has been included to facilitate backward compatibility with previous versions of MODFLOW but use of the **rate_scaling** option instead of the **head_limit** option is recommended. By default, **SHUTDOWN_THETA** is 0.7.

shutdown kappa—value that defines the weight applied to discharge rate for wells that limit the water level in a discharging well (defined using the HEAD_LIMIT keyword in the stress period data). SHUTDOWN_KAPPA is used to control discharge rate oscillations when the flow rate from the aquifer is less than the specified flow rate from the aquifer to the well. Values range between 0.0 and 1.0, and larger values increase the weight applied to the well discharge rate. The head limit option has been included to facilitate backward compatibility with previous versions of MODFLOW but use of the rate_scaling option instead of the head_limit option is recommended. By default, SHUTDOWN_KAPPA is 0.0001.

TS6—keyword to specify that record corresponds to a time-series file.

FILEIN—keyword to specify that an input filename is expected next.

ts6_filename—defines a time-series file defining time series that can be used to assign time-varying values. See the "Time-Variable Input" section for instructions on using the time-series capability.

OBS6—keyword to specify that record corresponds to an observations file.

obs6_filename—name of input file to define observations for the MAW package. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the MAW package.

MOVER—keyword to indicate that this instance of the MAW Package can be used with the Water Mover (MVR) Package. When the MOVER option is specified, additional memory is allocated within the package to store the available, provided, and received water.

nmawwells—integer value specifying the number of multi-aquifer wells that will be simulated for all stress periods.

wellno—integer value that defines the well number for this multi-aquifer well entry. wellno must be greater than zero and less than or equal to nmawwells.

radius—radius for the multi-aquifer well.

bottom—bottom elevation of the multi-aquifer well.

strt—starting head for the multi-aquifer well.

condeqn—character string that defines the conductance equation that is used to calculate the saturated conductance for the multi-aquifer well. Possible multi-aquifer well condeqn strings include: SPECIFIED–character keyword to indicate the multi-aquifer well saturated conductance will be specified. THEIM–character keyword to indicate the multi-aquifer well saturated conductance will be calculated using the Theim equation. SKIN–character keyword to indicate that the multi-aquifer well saturated conductance will be calculated using the screen top and bottom, screen hydraulic conductivity, and skin radius. CUMULATIVE–character keyword to indicate that the multi-aquifer well saturated conductance will be calculated using a combination of the Theim equation and the screen top and bottom, screen hydraulic conductivity, and skin radius. MEAN–character keyword to indicate the multi-aquifer well saturated conductance will be calculated using using the aquifer and screen top and bottom, aquifer and screen hydraulic conductivity, and well and skin radius.

ngwfnodes—integer value that defines the number of GWF nodes connected to this (wellno) multi-aquifer well. One or more screened intervals can be connected to the same GWF node. ngwfnodes must be greater than zero.

aux—represents the values of the auxiliary variables for each multi-aquifer well. The values of auxiliary variables must be present for each multi-aquifer well. The values must be specified in the order of the auxiliary variables specified in the OPTIONS block. If the package supports time series and the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

boundname—name of the multi-aquifer well cell. `boundname` is an ASCII character variable that can contain as many as 40 characters. If `boundname` contains spaces in it, then the entire name must be enclosed within single quotes.

wellno—integer value that defines the well number for this multi-aquifer well entry. `wellno` must be greater than zero and less than or equal to `nmawwells`.

icon—integer value that defines the GWF connection number for this multi-aquifer well connection entry. `iconn` must be greater than zero and less than or equal to `ngwfnodes` for multi-aquifer well `wellno`.

cellid—is the cell identifier, and depends on the type of grid that is used for the simulation. For a structured grid that uses the DIS input file, `cellid` is the layer, row, and column. For a grid that uses the DISV input file, `cellid` is the layer and cell2d number. If the model uses the unstructured discretization (DISU) input file, then `cellid` is the node number for the cell.

scrn_top—value that defines the top elevation of the screen for the multi-aquifer well connection. `scrn_top` can be any value if `condeqn` is SPECIFIED or THEIM. If the specified `scrn_top` is greater than the top of the GWF cell it is set equal to the top of the cell.

scrn_bot—value that defines the bottom elevation of the screen for the multi-aquifer well connection. `scrn_bot` can be any value if `condeqn` is SPECIFIED or THEIM. If the specified `scrn_bot` is less than the bottom of the GWF cell it is set equal to the bottom of the cell.

hk_skin—value that defines the skin (filter pack) hydraulic conductivity (if `condeqn` for the multi-aquifer well is SKIN, CUMULATIVE, or MEAN) or conductance (if `condeqn` for the multi-aquifer well is SPECIFIED) for each GWF node connected to the multi-aquifer well (`ngwfnodes`). `hk_skin` can be any value if `condeqn` is THEIM.

radius_skin—real value that defines the skin radius (filter pack radius) for the multi-aquifer well. `radius_skin` can be any value if `condeqn` is SPECIFIED or THEIM. Otherwise, `radius_skin` must be greater than `radius` for the multi-aquifer well.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. `iper` must be less than `nper` in the TDIS Package and greater than zero. The `iper` value assigned to a stress period block must be greater than the `iper` value assigned for the previous block.

wellno—integer value that defines the well number for this multi-aquifer well entry. `wellno` must be greater than zero and less than or equal to `nmawwells`.

mawsetting—line of information that is parsed into a keyword and values. Keyword values that can be used to start the `mawsetting` string include: STATUS, FLOWING_WELL, RATE, WELL_HEAD, HEAD_LIMIT, SHUT_OFF, RATE_SCALING, and AUXILIARY.

```
STATUS <status>
FLOWING_WELL <fwelev> <fwcond>
RATE <rate>
WELL_HEAD <well_head>
HEAD_LIMIT <head_limit>
SHUT_OFF <minrate> <maxrate>
RATE_SCALING <pump_elevation> <scaling_length>
AUXILIARY <auxname> <auxval>
```

status—keyword option to define well status. status can be ACTIVE, INACTIVE, or CONSTANT. By default, status is ACTIVE.

FLOWING_WELL—keyword to indicate the well is a flowing well. The flowing_well option can be used to simulate flowing wells when the simulated well head exceeds the specified drainage elevation.

fwelev—elevation used to determine whether or not the well is flowing.

fwcond—conductance used to calculate the discharge of a free flowing well. Flow occurs when the head in the well is above the well top elevation (fwelev).

rate—is the volumetric pumping rate for the multi-aquifer well. A positive value indicates recharge and a negative value indicates discharge (pumping). rate only applies to active (IBOUND> 0) multi-aquifer wells. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value. By default, the rate for each multi-aquifer well is zero.

well_head—is the head in the multi-aquifer well. well_head is only applied to constant head (STATUS is CONSTANT) and inactive (STATUS is INACTIVE) multi-aquifer wells. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

head_limit—is the limiting water level (head) in the well, which is the minimum of the well rate or the well inflow rate from the aquifer. head_limit is only applied to discharging wells (rate< 0). head_limit can be deactivated by specifying the text string 'off'. The head_limit option is based on the head_limit functionality available in the MNW2 (Konikow and others, 2009) package for MODFLOW-2005. The head_limit option has been included to facilitate backward compatibility with previous versions of MODFLOW but use of the rate_scaling option instead of the head_limit option is recommended. By default, head_limit is 'off'.

SHUT_OFF—keyword for activating well shut off capability. Subsequent values define the minimum and maximum pumping rate that a well must exceed to shutoff or reactivate a well, respectively, during a stress period. shut_off is only applied to discharging wells (rate< 0) and if head_limit is specified (not set to 'off'). If head_limit is specified, shut_off can be deactivated by specifying a minimum value equal to zero. The shut_off option is based on the shut_off functionality available in the MNW2 (Konikow and others, 2009) package for MODFLOW-2005. The shut_off option has been included to facilitate backward compatibility with previous versions of MODFLOW but use of the rate_scaling option instead of the shut_off option is recommended. By default, shut_off is not used.

minrate—is the minimum rate that a well must exceed to shutoff a well during a stress period. The well will shut down during a time step if the flow rate to the well from the aquifer is less than minrate. If a well is shut down during a time step, reactivation of the well cannot occur until the next time step to reduce oscillations. minrate must be less than maxrate.

maxrate—is the maximum rate that a well must exceed to reactivate a well during a stress period. The well will reactivate during a timestep if the well was shutdown during the previous time step and the flow rate to the well from the aquifer exceeds `maxrate`. Reactivation of the well cannot occur until the next time step if a well is shutdown to reduce oscillations. `maxrate` must be greater than `minrate`.

RATE_SCALING—activate rate scaling. If `rate_scaling` is specified, both `pump_elevation` and `scaling_length` must be specified. `rate_scaling` cannot be used with `head_limit`.

pump_elevation—is the elevation of the multi-aquifer well pump (`pump_elevation`). `pump_elevation` cannot be less than the bottom elevation (`bottom`) of the multi-aquifer well. By default, `pump_elevation` is set equal to the bottom of the largest `GWF` node number connected to a MAW well.

scaling_length—height above the pump elevation (`scaling_length`) below which the pumping rate is reduced. The default value for `scaling_length` is the well radius.

AUXILIARY—keyword for specifying auxiliary variable.

auxname—name for the auxiliary variable to be assigned `auxval`. `auxname` must match one of the auxiliary variable names defined in the `OPTIONS` block. If `auxname` does not match one of the auxiliary variable names defined in the `OPTIONS` block the data are ignored.

auxval—value for the auxiliary variable. If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

Example Input File – Conductance Calculated using Thiem Equation

```
begin options
  print_input
  print_head
  print_flows
  boundnames
  newton
  head fileout maw-1.head.bin
  budget fileout maw-1.cbc
end options

begin dimensions
  nmawwells 2
end dimensions

begin packagedata
#  wellno radius bottom strt condeqn ngwnodes name
       1    0.15 -100.0 9.14 theim       2 pwell
       2    0.25 -100.0 9.14 theim       1 iwell
end packagedata

begin connectiondata
# wellno conn l r c  stop sbot k  rskin
      1    1 1 51 51   0   0 0     0
```

```
     1    2 2 51 51    0    0  0      0
     2    1 2 2  2     0    0  0      0
end connectiondata

begin period 1
  1 rate_scaling -90. 5.
  1 rate -1767.
  2 status inactive
end period

begin period 100
  2 status active
  2 rate 529.
  1 rate -2767.
end period
```

Example Input File – Conductance Calculated using Screen Geometry

```
begin options
  print_input
  print_head
  print_flows
  boundnames
  newton
end options

begin dimensions
  nmawwells 2
end dimensions

begin packagedata
#  wellno radius bottom strt condeqn ngwnodes name
        1   0.15 -100.0 9.14    mean       2 pwell
        2   0.25 -100.0 9.14    mean       1 iwell
end packagedata

begin connectiondata
# wellno conn l r c  stop  sbot  k  rskin
        1    1 1 51 51   0. -100. 361. .25
        1    2 2 51 51   0. -100. 361. .25
        2    1 2 2  2  -50. -100. 361  .50
end connectiondata

begin period 1
  1 rate_scaling -90. 5.
  1 rate -1767.
  2 status inactive
end period

begin period 100
  2 status active
  2 rate 529.
  1 rate -2767.
end period
```

Example Input File – Flowing Well with Conductance Specified

```
begin options
  print_input
```

```
  print_head
  print_flows
  boundnames
  flowing_wells
  newton
end options

begin dimensions
  nmawwells 1
end dimensions

begin packagedata
#  wellno radius bottom strt condeqn ngwnodes name
      1   0.15 -514.9 9.14 specified    2 ntwell
end packagedata

begin connectiondata
# wellno conn l r c  stop   sbot         k  rskin
      1    1 1 51 51  -50 -514.9 111.3763    0
      1    2 2 51 51  -50 -514.9 445.9849    0
end connectiondata

begin period 1
  1 rate 0
  1 flowing_well 0. 7500. 0.5
end period
```

Available observation types

Multi-Aquifer Well Package observations include well head and all of the terms that contribute to the continuity equation for each multi-aquifer well. Additional LAK Package observations include the conductance for a well-aquifer connection conductance (`conductance`) and the calculated flowing well-aquifer connection conductance (`fw-conductance`). The data required for each MAW Package observation type is defined in table 14. Negative and positive values for `maw` observations represent a loss from and gain to the GWF model, respectively. For all other flow terms, negative and positive values represent a loss from and gain from the MAW package, respectively.

Table 14: Available MAW Package observation types

Stress Package	Observation type	ID	ID2	Description
MAW	head	wellno or boundname	–	Head in a multi-aquifer well. If boundname is specified, boundname must be unique for each multi-aquifer well.
MAW	from-mvr	wellno or boundname	–	Simulated inflow to a well from the MVR package for a multi-aquifer well or a group of multi-aquifer wells.

Table 14.   Available MAW Package observation types.—Continued

Stress Package	Observation type	ID	ID2	Description
MAW	maw	wellno or boundname	icon or –	Simulated flow rate for a multi-aquifer well or a group of multi-aquifer wells and its aquifer connection(s). If boundname is not specified for ID, then the simulated multi-aquifer well-aquifer flow rate at a specific multi-aquifer well connection is observed. In this case, ID2 must be specified and is the connection number icon.
MAW	rate	wellno or boundname	–	Simulated pumping rate for a multi-aquifer well or a group of multi-aquifer wells.
MAW	rate-to-mvr	wellno or boundname	–	Simulated well discharge that is available for the MVR package for a multi-aquifer well or a group of multi-aquifer wells.
MAW	fw-rate	wellno or boundname	–	Simulated flowing well flow rate for a multi-aquifer well or a group of multi-aquifer wells.
MAW	fw-to-mvr	wellno or boundname	–	Simulated flowing well discharge rate that is available for the MVR package for a multi-aquifer well or a group of multi-aquifer wells.
MAW	storage	wellno or boundname	–	Simulated storage flow rate for a multi-aquifer well or a group of multi-aquifer wells.
MAW	constant	wellno or boundname	–	Simulated constant-flow rate for a multi-aquifer well or a group of multi-aquifer wells.
MAW	conductance	wellno or boundname	icon or –	Simulated well conductance for a multi-aquifer well or a group of multi-aquifer wells and its aquifer connection(s). If boundname is not specified for ID, then the simulated multi-aquifer well conductance at a specific multi-aquifer well connection is observed. In this case, ID2 must be specified and is the connection number icon.
MAW	fw-conductance	wellno or boundname	–	Simulated flowing well conductance for a multi-aquifer well or a group of multi-aquifer wells.

Example Observation Input File

```
BEGIN OPTIONS
  PRECISION SINGLE
  DIGITS 10
  PRINT_INPUT
END OPTIONS

BEGIN CONTINUOUS FILEOUT my_model.maw.csv
  m1head    head  1
  m1rate01 maw   1 1
  m1rate02 maw   1 2
  m1rate    maw   well-1
  m2rate01 maw   well-2
END CONTINUOUS
```

Streamflow Routing (SFR) Package

Input to the Streamflow Routing (SFR) Package is read from the file that has type "SFR6" in the Name File. Any number of SFR Packages can be specified for a single groundwater flow model; however, water cannot be routed between reaches in separate packages except in cases where the MVR Package is used to route water between separate packages. All single valued variables are free format.

Reach connectivity must be explicitly specified for this version of the SFR Package, unlike the abbreviated SFR Package segment connectivity specified in previous versions of MODFLOW. Explicit specification of reach connectivity has been adopted to facilitate better validation of stream network connectivity by the program. Explicit reach connectivity means that a reach must be specified as an upstream connection for all downstream connections to the reach. Downstream connections for a reach are denoted with a negative reach number. Flow in a reach is unidirectional, always flowing from the upstream end to the downstream end of a reach. An example of the reach connectivity for a hypothetical stream network is shown in figure 2.
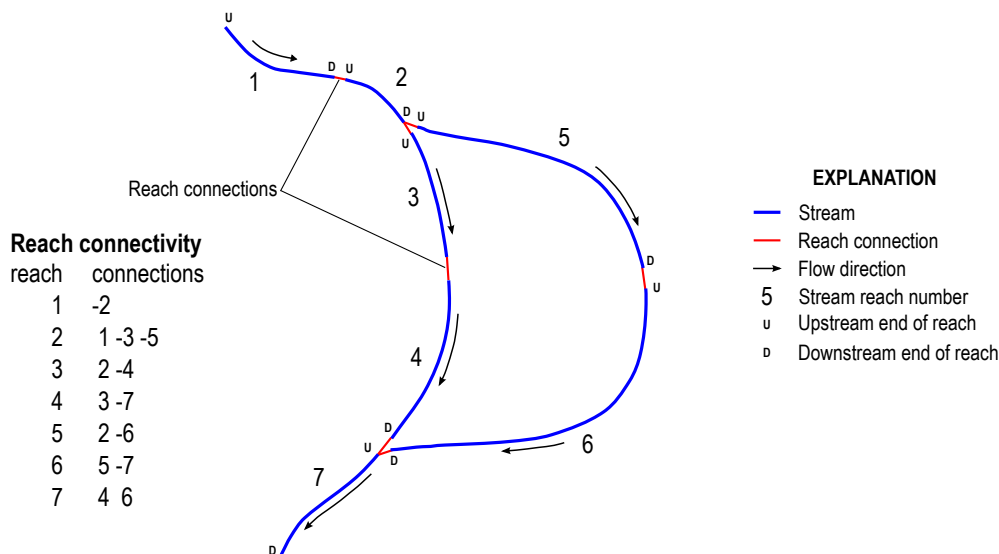
Figure 2: Simple stream network having seven reaches with a junction having two reaches, a confluence of two reaches, and the resulting reach connectivity. Downstream connections for a reach must include the reach as an upstream connection for all downstream connections to the reach. Downstream connections for a reach are denoted with a negative reach number

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [AUXILIARY <auxiliary(naux)>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_STAGE]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
```

```
  [STAGE FILEOUT <stagefile>]
  [BUDGET FILEOUT <budgetfile>]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
  [MOVER]
  [MAXIMUM_ITERATIONS <maximum_iterations>]
  [MAXIMUM_DEPTH_CHANGE <maximum_depth_change>]
  [UNIT_CONVERSION <unit_conversion>]
END OPTIONS

BEGIN DIMENSIONS
  MAXBOUND <maxbound>
END DIMENSIONS

BEGIN PACKAGEDATA
  <rno> <cellid(ncelldim)> <rlen> <rwid> <rgrd> <rtp> <rbth> <rhk> <man> <ncon> <ustrf> <ndv> [<aux(naux)>] [<bound
  <rno> <cellid(ncelldim)> <rlen> <rwid> <rgrd> <rtp> <rbth> <rhk> <man> <ncon> <ustrf> <ndv> [<aux(naux)>] [<bound
  ...
END PACKAGEDATA

BEGIN CONNECTIONDATA
  <rno> <ic(ncon(rno))>
  <rno> <ic(ncon(rno))>
  ...
END CONNECTIONDATA
```

IF ndv IS GREATER THAN ZERO FOR ANY REACH

```
BEGIN DIVERSIONS
  <rno> <idv> <iconr> <cprior>
  <rno> <idv> <iconr> <cprior>
  ...
END DIVERSIONS
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <rno> <sfrsetting>
  <rno> <sfrsetting>
  ...
END PERIOD
```

Explanation of Variables

auxiliary—defines an array of one or more auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided on this line; however, lists of information provided in subsequent blocks must have a column of data for each auxiliary variable name defined here. The number of auxiliary variables detected on this line determines the value for naux. Comments cannot be provided anywhere on this line as they will be interpreted as auxiliary variable names. Auxiliary variables may not be used by the package, but they will be available for use by other parts of the program. The program will terminate with an error if auxiliary variables are specified on more than one line in the options block.

BOUNDNAMES—keyword to indicate that boundary names may be provided with the list of stream reach cells.

PRINT_INPUT—keyword to indicate that the list of stream reach information will be written to the listing file immediately after it is read.

PRINT_STAGE—keyword to indicate that the list of stream reach stages will be printed to the listing file for every stress period in which "HEAD PRINT" is specified in Output Control. If there is no Output Control option and PRINT_STAGE is specified, then stages are printed for the last time step of each stress period.

PRINT_FLOWS—keyword to indicate that the list of stream reach flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and PRINT_FLOWS is specified, then flow rates are printed for the last time step of each stress period.

SAVE_FLOWS—keyword to indicate that stream reach flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

STAGE—keyword to specify that record corresponds to stage.

stagefile—name of the binary output file to write stage information.

BUDGET—keyword to specify that record corresponds to the budget.

FILEOUT—keyword to specify that an output filename is expected next.

budgetfile—name of the binary output file to write budget information.

TS6—keyword to specify that record corresponds to a time-series file.

FILEIN—keyword to specify that an input filename is expected next.

ts6_filename—defines a time-series file defining time series that can be used to assign time-varying values. See the "Time-Variable Input" section for instructions on using the time-series capability.

OBS6—keyword to specify that record corresponds to an observations file.

obs6_filename—name of input file to define observations for the SFR package. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the SFR package.

MOVER—keyword to indicate that this instance of the SFR Package can be used with the Water Mover (MVR) Package. When the MOVER option is specified, additional memory is allocated within the package to store the available, provided, and received water.

maximum_iterations—value that defines an maximum number of Streamflow Routing Newton-Raphson iterations allowed for a reach. By default, maxsfrit is equal to 100.

maximum_depth_change—value that defines the depth closure tolerance. By default, dmaxchg is equal to 1×10^{-5}.

unit_conversion—value (or conversion factor) that is used in calculating stream depth for stream reach. A constant of 1.486 is used for flow units of cubic feet per second, and a constant of 1.0 is used for units of cubic meters per second. The constant must be multiplied by 86,400 when using time units of days in the simulation.

maxbound—integer value specifying the maximum number of stream reach cells that will be specified for use during any stress period.

rno—integer value that defines the reach number associated with the specified data on the line. rno must be greater than zero and less than or equal to MAXBOUND.

cellid—The keyword 'none' must be specified for reaches that are not connected to an underlying GWF cell. The keyword 'none' is used for reaches that are in cells that have IDOMAIN values less than one or are in areas not covered by the GWF model grid. Reach-aquifer flow is not calculated if the keyword 'none' is specified.

rlen—real value that defines the reach length. rlen must be greater than zero.

rwid—real value that defines the reach width. rwid must be greater than zero.

rgrd—real value that defines the stream gradient (slope) across the reach. rgrd must be greater than zero.

rtp—real value that defines the top elevation of the reach streambed.

rbth—real value that defines the thickness of the reach streambed. rbth can be any value if cellid is 'none'. Otherwise, rbth must be greater than zero.

rhk—real value that defines the hydraulic conductivity of the reach streambed. rhk can be any positive value if cellid is 'none'. Otherwise, rhk must be greater than zero.

man—real or character value that defines the Manning's roughness coefficient for the reach. man must be greater than zero. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

ncon—integer value that defines the number of reaches connected to the reach.

ustrf—real value that defines the fraction of upstream flow from each upstream reach that is applied as upstream inflow to the reach. The sum of all ustrf values for all reaches connected to the same upstream reach must be equal to one and ustrf must be greater than or equal to zero.

ndv—integer value that defines the number of downstream diversions for the reach.

aux—represents the values of the auxiliary variables for each stream reach. The values of auxiliary variables must be present for each stream reach. The values must be specified in the order of the auxiliary variables specified in the OPTIONS block. If the package supports time series and the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

boundname—name of the stream reach cell. boundname is an ASCII character variable that can contain as many as 40 characters. If boundname contains spaces in it, then the entire name must be enclosed within single quotes.

rno—integer value that defines the reach number associated with the specified data on the line. rno must be greater than zero and less than or equal to MAXBOUND.

ic—integer value that defines the reach number of the reach connected to the current reach and whether it is connected to the upstream or downstream end of the reach. Negative `ic` numbers indicate connected reaches are connected to the downstream end of the current reach. Positive `ic` numbers indicate connected reaches are connected to the upstream end of the current reach. The absolute value of `ic` must be greater than zero and less than or equal to `MAXBOUND`.

rno—integer value that defines the reach number associated with the specified data on the line. `rno` must be greater than zero and less than or equal to `MAXBOUND`.

idv—integer value that defines the downstream diversion number for the diversion for reach `rno`. `idv` must be greater than zero and less than or equal to `ndv` for reach `rno`.

iconr—integer value that defines the downstream reach that will receive the diverted water. `idv` must be greater than zero and less than or equal to `MAXBOUND`. Furthermore, reach `iconr` must be a downstream connection for reach `rno`.

cprior—character string value that defines the the prioritization system for the diversion, such as when insufficient water is available to meet all diversion stipulations, and is used in conjunction with the value of `flow` value specified in the `STRESS_PERIOD_DATA` section. Available diversion options include: (1) `cprior` = 'FRACTION', then the amount of the diversion is computed as a fraction of the streamflow leaving reach `rno` (Q_{DS}); in this case, $0.0 \leq$ `divflow` ≤ 1.0. (2) `cprior` = 'EXCESS', a diversion is made only if Q_{DS} for reach `rno` exceeds the value of `divflow`. If this occurs, then the quantity of water diverted is the excess flow ($Q_{DS}-$ `divflow`) and Q_{DS} from reach `rno` is set equal to `divflow`. This represents a flood-control type of diversion, as described by Danskin and Hanson (2002). (3) `cprior` = 'THRESHOLD', then if Q_{DS} in reach `rno` is less than the specified diversion flow (`divflow`), no water is diverted from reach `rno`. If Q_{DS} in reach `rno` is greater than or equal to (`divflow`), (`divflow`) is diverted and Q_{DS} is set to the remainder ($Q_{DS}-$ `divflow`)). This approach assumes that once flow in the stream is sufficiently low, diversions from the stream cease, and is the 'priority' algorithm that originally was programmed into the STR1 Package (Prudic, 1989). (4) `cprior` = 'UPTO' – if Q_{DS} in reach `rno` is greater than or equal to the specified diversion flow (`divflow`), Q_{DS} is reduced by `divflow`. If Q_{DS} in reach `rno` is less than (`divflow`), `divflow` is set to Q_{DS} and there will be no flow available for reaches connected to downstream end of reach `rno`.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. `iper` must be less than `nper` in the TDIS Package and greater than zero. The `iper` value assigned to a stress period block must be greater than the `iper` value assigned for the previous block.

rno—integer value that defines the reach number associated with the specified data on the line. `rno` must be greater than zero and less than or equal to `MAXBOUND`.

sfrsetting—line of information that is parsed into a keyword and values. Keyword values that can be used to start the `sfrsetting` string include: `STATUS`, `MANNING`, `STAGE`, `INFLOW`, `RAINFALL`, `EVAPORATION`, `RUNOFF`, `DIVERSION`, `UPSTREAM_FRACTION`, and `AUXILIARY`.

```
STATUS <status>
MANNING <manning>
STAGE <stage>
INFLOW <inflow>
RAINFALL <rainfall>
EVAPORATION <evaporation>
```

```
RUNOFF <runoff>
DIVERSION <idv> <divrate>
UPSTREAM_FRACTION <upstream_fraction>
AUXILIARY <auxname> <auxval>
```

status—keyword option to define stream reach status. `status` can be `ACTIVE`, `INACTIVE`, or `SIMPLE`. The `SIMPLE status` option simulates streamflow using a user-specified stage for a reach or a stage set to the top of the reach (depth = 0). In cases where the simulated leakage calculated using the specified stage exceeds the sum of inflows to the reach, the stage is set to the top of the reach and leakage is set equal to the sum of inflows. Upstream factions should be changed using the `UPSTREAM_FRACTION sfrsetting` if the status for one or more reaches is changed to `ACTIVE` or `INACTIVE`. For example, if one of two downstream connections for a reach is inactivated, the upstream fraction for the active and inactive downstream reach should be changed to 1.0 and 0.0, respectively, to ensure that the active reach receives all of the downstream outflow from the upstream reach. By default, `status` is `ACTIVE`.

manning—real or character value that defines the Manning's roughness coefficient for the reach. `manning` must be greater than zero. If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

stage—real or character value that defines the stage for the reach. The specified `stage` is only applied if the reach uses the simple routing option. If `STAGE` is not specified for reaches that use the simple routing option, the specified stage is set to the top of the reach. If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

inflow—real or character value that defines the volumetric inflow rate for the streamflow routing reach. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value. By default, inflow rates are zero for each reach.

rainfall—real or character value that defines the volumetric rate per unit area of water added by precipitation directly on the streamflow routing reach. If the Options block includes a TIME-SERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value. By default, rainfall rates are zero for each reach.

evaporation—real or character value that defines the volumetric rate per unit area of water subtracted by evaporation from the streamflow routing reach. A positive evaporation rate should be provided. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value. By default, evaporation rates are zero for each reach.

runoff—real or character value that defines the volumetric rate of diffuse overland runoff that enters the streamflow routing reach. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value. By default, runoff rates are zero for each reach.

DIVERSION—keyword to indicate diversion record.

`idv`—diversion number.

`divrate`—real or character value that defines the volumetric diversion (`divflow`) rate for the stream-flow routing reach. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

`upstream_fraction`—real value that defines the fraction of upstream flow (`ustrf`) from each upstream reach that is applied as upstream inflow to the reach. The sum of all `ustrf` values for all reaches connected to the same upstream reach must be equal to one.

`AUXILIARY`—keyword for specifying auxiliary variable.

`auxname`—name for the auxiliary variable to be assigned `auxval`. `auxname` must match one of the auxiliary variable names defined in the `OPTIONS` block. If `auxname` does not match one of the auxiliary variable names defined in the `OPTIONS` block the data are ignored.

`auxval`—value for the auxiliary variable. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

Example Input File

```
BEGIN OPTIONS
  UNIT_CONVERSION 1.486
  BOUNDNAMES
  PRINT_STAGE
  PRINT_FLOWS
  STAGE FILEOUT sfr-1.stage.bin
  BUDGET FILEOUT sfr-1.cbc
END OPTIONS

#dimension block is required
BEGIN DIMENSIONS
  MAXBOUND 37
END DIMENSIONS

BEGIN PACKAGEDATA
#rno k  i j rlen rwid     rgrd      rtp   rbth      rhk   man ncon ustrf ndv boundname
   1 1  1 1 4500. 12    8.67E-04 1093.048  3.0 0.00003 0.03    1  1.0   0   reach1
   2 1  2 2 7000. 12    8.67E-04 1088.059  3.0 0.00003 0.03    2  1.0   0   reach2
   3 1  3 3 6000. 12    8.67E-04 1082.419  3.0 0.00003 0.03    2  1.0   0   reach3
   4 1  3 4 5550. 12    8.67E-04 1077.408  3.0 0.00003 0.03    3  1.0   1   reach4
   5 1  4 5 6500. 12    9.43E-04 1071.934  3.0 0.00003 0.03    2  1.0   0
   6 1  5 6 5000. 12    9.43E-04 1066.509  3.0 0.00003 0.03    2  1.0   0
   7 1  6 6 5000. 12    9.43E-04 1061.792  3.0 0.00003 0.03    2  1.0   0
   8 1  7 6 5000. 12    9.43E-04 1057.075  3.0 0.00003 0.03    2  1.0   0
   9 1  8 6 5000. 12    9.43E-04 1052.359  3.0 0.00003 0.03    2  1.0   0
  10 1  3 5 5000. 10    5.45E-04 1073.636  2.0 0.00003 0.03    2  0.0   0   canal
  11 1  3 6 5000. 10    5.45E-04 1070.909  2.0 0.00003 0.03    2  1.0   0   canal
  12 1  3 7 4500. 10    5.45E-04 1068.318  2.0 0.00003 0.03    2  1.0   0   canal
  13 1  4 8 6000. 10    5.45E-04 1065.455  2.0 0.00003 0.03    2  1.0   0   canal
  14 1  5 8 5000. 10    5.45E-04 1062.455  2.0 0.00003 0.03    2  1.0   0   canal
```

```
15 1  6 8 2000. 10    5.45E-04 1060.545  2.0 0.00003 0.03    2   1.0    0    canal
16 1  510 2500. 10    1.81E-03 1077.727  3.0 0.00003 0.03    1   1.0    0
17 1  5 9 5000. 10    1.81E-03 1070.909  3.0 0.00003 0.03    2   1.0    0
18 1  6 8 3500. 10    1.81E-03 1063.182  3.0 0.00003 0.03    2   1.0    0
19 1  6 8 4000. 15    1.00E-03 1058.000  3.0 0.00003 0.03    3   1.0    0
20 1  7 7 5000. 15    1.00E-03 1053.500  3.0 0.00003 0.03    2   1.0    0
21 1  8 7 3500. 15    1.00E-03 1049.250  3.0 0.00003 0.03    2   1.0    0
22 1  8 6 2500. 15    1.00E-03 1046.250  3.0 0.00003 0.03    2   1.0    0
23 1  9 6 5000. 12    9.09E-04 1042.727  3.0 0.00003 0.03    3   1.0    0
24 1 10 7 5000. 12    9.09E-04 1038.182  3.0 0.00003 0.03    2   1.0    0
25 1 11 7 5000. 12    9.09E-04 1033.636  3.0 0.00003 0.03    2   1.0    0
26 1 12 7 5000. 12    9.09E-04 1029.091  3.0 0.00003 0.03    2   1.0    0
27 1 13 7 2000. 12    9.09E-04 1025.909  3.0 0.00003 0.03    2   1.0    0
28 1 14 9 5000. 55    9.67E-04 1037.581  3.0 0.00006 0.025   1   1.0    0
29 1 13 8 5500. 55    9.67E-04 1032.500  3.0 0.00006 0.025   2   1.0    0
30 1 13 7 5000. 55    9.67E-04 1027.419  3.0 0.00006 0.025   2   1.0    0
31 1 13 6 5000. 40    1.25E-03 1021.875  3.0 0.00006 0.025   3   1.0    0
32 1 13 5 5000. 40    1.25E-03 1015.625  3.0 0.00006 0.025   2   1.0    0
33 1 13 4 5000. 40    1.25E-03 1009.375  3.0 0.00006 0.025   2   1.0    0
34 1 13 3 5000. 40    1.25E-03 1003.125  3.0 0.00006 0.025   2   1.0    0
35 1 13 2 5000. 40    1.25E-03 996.8750  3.0 0.00006 0.025   2   1.0    0
36 1 13 1 3000. 40    1.25E-03 991.8750  3.0 0.00006 0.025   2   1.0    0
37 none  5000. 40    1.25E-03 985.6250  3.0 0.00006 0.025   1   1.0    0
END PACKAGEDATA

BEGIN CONNECTIONDATA
#rno ic1 ic2 ic3
   1  -2
   2   1  -3
   3   2  -4
   4   3  -5 -10
   5   4  -6
   6   5  -7
   7   6  -8
   8   7  -9
   9   8 -23
  10   4 -11
  11  10 -12
  12  11 -13
  13  12 -14
  14  13 -15
  15  14 -19
  16 -17
  17  16 -18
  18  17 -19
  19  15  18 -20
  20  19 -21
  21  20 -22
  22  21 -23
  23   9  22 -24
  24  23 -25
  25  24 -26
  26  25 -27
  27  26 -31
  28 -29
  29  28 -30
  30  29 -31
  31  27  30 -32
  32  31 -33
```

```
 33  32 -34
 34  33 -35
 35  34 -36
 36  35 -37
 37  36
END CONNECTIONDATA

BEGIN DIVERSIONS
#  rno idv iconr cprior
    4   1    10 UPTO
END DIVERSIONS

BEGIN PERIOD 1
# rno sfrsetting
   1 inflow 25.
  16 inflow 10.
  28 inflow 150.
   4 diversion 1 10.
  10 status simple
  11 status simple
  12 status simple
  13 status simple
  14 status simple
  15 status simple
  10 stage 1075.5454
  11 stage 1072.6363
  12 stage 1069.8727
  13 stage 1066.8181
  14 stage 1063.6181
  15 stage 1061.5818
END PERIOD
```

Available observation types

Streamflow Routing Package observations include reach stage and all of the terms that contribute to the continuity equation for each stream reach. Additional SFR Package observations include the sum of inflows from upstream reaches and from mover terms (`upstream-flow`) and downstream outflow from a reach prior to diversions and the mover package (`downstream-flow`). The data required for each SFR Package observation type is defined in table 15. Negative and positive values for `sfr` observations represent a loss from and gain to the GWF model, respectively. For all other flow terms, negative and positive values represent a loss from and gain from the SFR package, respectively.

Table 15: Available SFR Package observation types

Stress Package	Observation type	ID	ID2	Description
SFR	stage	rno or boundname	–	Surface-water stage in a stream-reach boundary. If boundname is specified, boundname must be unique for each reach.
SFR	ext-inflow	rno or boundname	–	Inflow into a stream-reach from an external boundary for a stream-reach or a group of stream-reaches.

Table 15.   Available SFR Package observation types.—Continued

Stress Package	Observation type	ID	ID2	Description
SFR	inflow	rno or boundname	–	Inflow into a stream-reach from upstream reaches for a stream-reach or a group of stream-reaches.
SFR	from-mvr	rno or boundname	–	Inflow into a stream-reach from the MVR package for a stream-reach or a group of stream-reaches.
SFR	rainfall	rno or boundname	–	Rainfall rate applied to a stream-reach or a group of stream-reaches.
SFR	runoff	rno or boundname	–	Runoff rate applied to a stream-reach or a group of stream-reaches.
SFR	sfr	rno or boundname	–	Simulated flow rate for a stream-reach and its aquifer connection for a stream-reach or a group of stream-reaches.
SFR	evaporation	rno or boundname	–	Simulated evaporation rate from a stream-reach or a group of stream-reaches.
SFR	outflow	rno or boundname	–	Outflow from a stream-reach to downstream reaches for a stream-reach or a group of stream-reaches.
SFR	ext-outflow	rno or boundname	–	Outflow from a stream-reach to an external boundary for a stream-reach or a group of stream-reaches.
SFR	to-mvr	rno or boundname	–	Outflow from a stream-reach that is available for the MVR package for a stream-reach or a group of stream-reaches.
SFR	upstream-flow	rno or boundname	–	Upstream flow for a stream-reach or a group of stream-reaches from upstream reaches and the MVR package.
SFR	downstream-flow	rno or boundname	–	Downstream flow for a stream-reach or a group of stream-reaches prior to diversions and the MVR package.

Example Observation Input File

```
BEGIN OPTIONS
  DIGITS 8
  PRINT_INPUT
END OPTIONS

BEGIN CONTINUOUS FILEOUT my_model.sfr.csv
# obsname    obstype        id
 gage1stage STAGE          reach4
 gage2stage STAGE          7
 gage2inflow INFLOW        7
 gage2disch DOWNSTREAM-FLOW 7
```

```
  gage3stage STAGE          14
END CONTINUOUS


BEGIN CONTINUOUS FILEOUT my_model.sfr.leakage.csv
# obsname    obstype         id
 leak1      SFR             reach1
 leak10     SFR             10
 leak11     SFR             11
 leak12     SFR             12
 leak13     SFR             13
 leak14     SFR             14
 leak15     SFR             15
 leakcanal SFR             canal #Sum of flows between canal reaches and groundwater
END CONTINUOUS
```

Lake (LAK) Package

Input to the Lake (LAK) Package is read from the file that has type "LAK6" in the Name File. Any number of LAK Packages can be specified for a single groundwater flow model. All single valued variables are free format.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [AUXILIARY <auxiliary(naux)>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_STAGE]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [STAGE FILEOUT <stagefile>]
  [BUDGET FILEOUT <budgetfile>]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
  [MOVER]
  [SURFDEP <surfdep>]
  [TIME_CONVERSION <time_conversion>]
  [LENGTH_CONVERSION <length_conversion>]
END OPTIONS
```

```
BEGIN DIMENSIONS
  NLAKES <nlakes>
  NOUTLETS <noutlets>
  NTABLES <ntables>
END DIMENSIONS
```

```
BEGIN PACKAGEDATA
  <lakeno> <strt> <nlakeconn> [<aux(naux)>] [<boundname>]
  <lakeno> <strt> <nlakeconn> [<aux(naux)>] [<boundname>]
  ...
END PACKAGEDATA
```

IF `nlakeconn` IS GREATER THAN ZERO FOR ANY LAKE

```
BEGIN CONNECTIONDATA
  <lakeno> <iconn> <cellid(ncelldim)> <claktype> <bedleak> <belev> <telev> <connlen> <connwidth>
  <lakeno> <iconn> <cellid(ncelldim)> <claktype> <bedleak> <belev> <telev> <connlen> <connwidth>
  ...
END CONNECTIONDATA
```

IF `ntables` IS GREATER THAN ZERO

```
BEGIN TABLES
  <lakeno> TAB6 FILEIN <tab6_filename>
  <lakeno> TAB6 FILEIN <tab6_filename>
  ...
END TABLES
```

IF `noutlets` IS GREATER THAN ZERO FOR ANY LAKE

```
BEGIN OUTLETS
  <outletno> <lakein> <lakeout> <couttype> <invert> <width> <rough> <slope>
  <outletno> <lakein> <lakeout> <couttype> <invert> <width> <rough> <slope>
  ...
END OUTLETS
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <lakeno> <laksetting>
  <lakeno> <laksetting>
  ...
  <outletno> <outletsetting>
  <outletno> <outletsetting>
  ...
END PERIOD
```

Explanation of Variables

`auxiliary`—defines an array of one or more auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided on this line; however, lists of information provided in subsequent blocks must have a column of data for each auxiliary variable name defined here. The number of auxiliary variables detected on this line determines the value for naux. Comments cannot be provided anywhere on this line as they will be interpreted as auxiliary variable names. Auxiliary variables may not be used by the package, but they will be available for use by other parts of the program. The program will terminate with an error if auxiliary variables are specified on more than one line in the options block.

`BOUNDNAMES`—keyword to indicate that boundary names may be provided with the list of lake cells.

`PRINT_INPUT`—keyword to indicate that the list of lake information will be written to the listing file immediately after it is read.

`PRINT_STAGE`—keyword to indicate that the list of lake stages will be printed to the listing file for every stress period in which "HEAD PRINT" is specified in Output Control. If there is no Output Control option and `PRINT_STAGE` is specified, then stages are printed for the last time step of each stress period.

`PRINT_FLOWS`—keyword to indicate that the list of lake flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and `PRINT_FLOWS` is specified, then flow rates are printed for the last time step of each stress period.

`SAVE_FLOWS`—keyword to indicate that lake flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

`STAGE`—keyword to specify that record corresponds to stage.

`stagefile`—name of the binary output file to write stage information.

`BUDGET`—keyword to specify that record corresponds to the budget.

`FILEOUT`—keyword to specify that an output filename is expected next.

`budgetfile`—name of the binary output file to write budget information.

`TS6`—keyword to specify that record corresponds to a time-series file.

`FILEIN`—keyword to specify that an input filename is expected next.

`ts6_filename`—defines a time-series file defining time series that can be used to assign time-varying values. See the "Time-Variable Input" section for instructions on using the time-series capability.

`OBS6`—keyword to specify that record corresponds to an observations file.

`obs6_filename`—name of input file to define observations for the LAK package. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the LAK package.

`MOVER`—keyword to indicate that this instance of the LAK Package can be used with the Water Mover (MVR) Package. When the `MOVER` option is specified, additional memory is allocated within the package to store the available, provided, and received water.

`surfdep`—real value that defines the surface depression depth for `VERTICAL` lake-`GWF` connections. If specified, `surfdep` must be greater than or equal to zero. If `SURFDEP` is not specified, a default value of zero is used for all vertical lake-`GWF` connections.

`time_conversion`—value that is used in converting outlet flow terms that use Manning's equation or gravitational acceleration to consistent time units. `time_conversion` should be set to 1.0, 60.0, 3,600.0, 86,400.0, and 31,557,600.0 when using time units (`time_units`) of seconds, minutes, hours, days, or years in the simulation, respectively. `convtime` does not need to be specified if no lake outlets are specified or `time_units` are seconds.

`length_conversion`—real value that is used in converting outlet flow terms that use Manning's equation or gravitational acceleration to consistent length units. `length_conversion` should be set to 3.28081, 1.0, and 100.0 when using length units (`length_units`) of feet, meters, or centimeters in the simulation, respectively. `length_conversion` does not need to be specified if no lake outlets are specified or `length_units` are meters.

`nlakes`—value specifying the number of lakes that will be simulated for all stress periods.

`noutlets`—value specifying the number of outlets that will be simulated for all stress periods. If `NOUTLETS` is not specified, a default value of zero is used.

`ntables`—value specifying the number of lakes tables that will be used to define the lake stage, volume relation, and surface area. If `NTABLES` is not specified, a default value of zero is used.

`lakeno`—integer value that defines the lake number for this lake entry. `lakeno` must be greater than zero and less than or equal to `nlakes`.

`strt`—real value that defines the starting stage for the lake.

`nlakeconn`—integer value that defines the number of `GWF` nodes connected to this (`lakeno`) lake. There can only be one vertical lake connection to each `GWF` node. `nlakeconn` must be greater than zero.

aux—represents the values of the auxiliary variables for each lake. The values of auxiliary variables must be present for each lake. The values must be specified in the order of the auxiliary variables specified in the OPTIONS block. If the package supports time series and the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

boundname—name of the lake cell. boundname is an ASCII character variable that can contain as many as 40 characters. If boundname contains spaces in it, then the entire name must be enclosed within single quotes.

lakeno—integer value that defines the lake number for this lake entry. lakeno must be greater than zero and less than or equal to nlakes.

iconn—integer value that defines the GWF connection number for this lake connection entry. iconn must be greater than zero and less than or equal to nlakeconn for lake lakeno.

cellid—is the cell identifier, and depends on the type of grid that is used for the simulation. For a structured grid that uses the DIS input file, cellid is the layer, row, and column. For a grid that uses the DISV input file, cellid is the layer and cell2d number. If the model uses the unstructured discretization (DISU) input file, then cellid is the node number for the cell.

claktype—character string that defines the lake-GWF connection type for the lake connection. Possible lake-GWF connection type strings include: VERTICAL–character keyword to indicate the lake-GWF connection is vertical and connection conductance calculations use the hydraulic conductivity corresponding to the K_{33} tensor component defined for cellid in the NPF package. HORIZONTAL– character keyword to indicate the lake-GWF connection is horizontal and connection conductance calculations use the hydraulic conductivity corresponding to the K_{11} tensor component defined for cellid in the NPF package. EMBEDDEDH–character keyword to indicate the lake-GWF connection is embedded in a single cell and connection conductance calculations use the hydraulic conductivity corresponding to the K_{11} tensor component defined for cellid in the NPF package. EMBEDDEDV– character keyword to indicate the lake-GWF connection is embedded in a single cell and connection conductance calculations use the hydraulic conductivity corresponding to the K_{33} tensor component defined for cellid in the NPF package. Embedded lakes can only be connected to a single cell (nlakconn = 1) and there must be a lake table associated with each embedded lake.

bedleak—real value that defines the bed leakance for the lake-GWF connection. bedk must be greater than or equal to zero.

belev—real value that defines the bottom elevation for a HORIZONTAL lake-GWF connection. Any value can be specified if claktype is VERTICAL, EMBEDDEDH, or EMBEDDEDV. If claktype is HORIZONTAL and belev is not equal to telev, belev must be greater than or equal to the bottom of the GWF cell cellid. If belev is equal to telev, belev is reset to the bottom of the GWF cell cellid.

telev—real value that defines the top elevation for a HORIZONTAL lake-GWF connection. Any value can be specified if claktype is VERTICAL, EMBEDDEDH, or EMBEDDEDV. If claktype is HORIZONTAL and telev is not equal to belev, telev must be less than or equal to the top of the GWF cell cellid. If telev is equal to belev, telev is reset to the top of the GWF cell cellid.

connlen—real value that defines the distance between the connected GWF cellid node and the lake for a HORIZONTAL, EMBEDDEDH, or EMBEDDEDV lake-GWF connection. connlen must be greater than zero for a HORIZONTAL, EMBEDDEDH, or EMBEDDEDV lake-GWF connection. Any value can be specified if claktype is VERTICAL.

connwidth—real value that defines the connection face width for a `HORIZONTAL` lake-`GWF` connection. `connwidth` must be greater than zero for a `HORIZONTAL` lake-`GWF` connection. Any value can be specified if `claktype` is `VERTICAL`, `EMBEDDEDH`, or `EMBEDDEDV`.

lakeno—integer value that defines the lake number for this lake entry. `lakeno` must be greater than zero and less than or equal to `nlakes`.

TAB6—keyword to specify that record corresponds to a table file.

FILEIN—keyword to specify that an input filename is expected next.

tab6_filename—character string that defines the path and filename for the file containing lake table data for the lake connection. The `ctabname` file includes the number of entries in the file and the relation between stage, surface area, and volume for each entry in the file. Lake table files for `EMBEDDEDH` and `EMBEDDEDV` lake-`GWF` connections also include lake-`GWF` exchange area data for each entry in the file. Input instructions for the `ctabname` file is included at the LAK package lake table file input instructions section.

outletno—integer value that defines the outlet number for this outlet entry. `outletno` must be greater than zero and less than or equal to `noutlets`.

lakein—integer value that defines the lake number that outlet is connected to. `lakein` must be greater than zero and less than or equal to `nlakes`.

lakeout—integer value that defines the lake number that outlet discharge from lake outlet `outletno` is routed to. `lakeout` must be greater than or equal to zero and less than or equal to `nlakes`. If `lakeout` is zero, outlet discharge from lake outlet `outletno` is discharged to an external boundary.

couttype—character string that defines the outlet type for the outlet `outletno`. Possible `couttype` strings include: `SPECIFIED`–character keyword to indicate the outlet is defined as a specified flow. `MANNING`–character keyword to indicate the outlet is defined using Manning's equation. `WEIR`–character keyword to indicate the outlet is defined using a sharp weir equation.

invert—real value that defines the invert elevation for the lake outlet. Any value can be specified if `couttype` is `SPECIFIED`. If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

width—real value that defines the width of the lake outlet. Any value can be specified if `couttype` is `SPECIFIED`. If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

rough—real value that defines the roughness coefficient for the lake outlet. Any value can be specified if `couttype` is not `MANNING`. If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

slope—real value that defines the bed slope for the lake outlet. Any value can be specified if `couttype` is not `MANNING`. If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. `iper` must be less than `nper` in the TDIS Package and greater than zero. The `iper` value assigned to a stress period block must be greater than the `iper` value assigned for the previous block.

lakeno—integer value that defines the lake number associated with the specified data on the line. `lakeno` must be greater than zero and less than or equal to `nlakes`.

laksetting—line of information that is parsed into a keyword and values. Keyword values that can be used to start the `laksetting` string include: STATUS, STAGE, RAINFALL, EVAPORATION, RUNOFF, WITHDRAWAL, and AUXILIARY.

```
STATUS <status>
STAGE <stage>
RAINFALL <rainfall>
EVAPORATION <evaporation>
RUNOFF <runoff>
WITHDRAWAL <withdrawal>
AUXILIARY <auxname> <auxval>
```

status—keyword option to define lake status. `status` can be ACTIVE, INACTIVE, or CONSTANT. By default, `status` is ACTIVE.

rate—real or character value that defines the extraction rate for the lake outflow. A positive value indicates inflow and a negative value indicates outflow from the lake. `rate` only applies to active (IBOUND> 0) lakes. A specified `rate` is only applied if `couttype` for the `outletno` is SPECIFIED. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value. By default, the `rate` for each SPECIFIED lake outlet is zero.

stage—real or character value that defines the stage for the lake. The specified `stage` is only applied if the lake is a constant stage lake. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

rainfall—real or character value that defines the rainfall rate for the lake. `value` must be greater than or equal to zero. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

evaporation—real or character value that defines the maximum evaporation rate for the lake. `value` must be greater than or equal to zero. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

runoff—real or character value that defines the runoff rate for the lake. `value` must be greater than or equal to zero. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

withdrawal—real or character value that defines the maximum withdrawal rate for the lake. `value` must be greater than or equal to zero. If the Options block includes a TIMESERIESFILE entry (see

the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

AUXILIARY—keyword for specifying auxiliary variable.

auxname—name for the auxiliary variable to be assigned auxval. auxname must match one of the auxiliary variable names defined in the OPTIONS block. If auxname does not match one of the auxiliary variable names defined in the OPTIONS block the data are ignored.

auxval—value for the auxiliary variable. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

outletno—integer value that defines the outlet number for this outlet entry. outletno must be greater than zero and less than or equal to noutlets.

outletsetting—line of information that is parsed into a keyword and values. Keyword values that can be used to start the outletsetting string include: RATE, INVERT, WIDTH, SLOPE, and ROUGH.

```
RATE <rate>
INVERT <invert>
WIDTH <width>
SLOPE <slope>
ROUGH <rough>
```

invert—real or character value that defines the invert elevation for the lake outlet. A specified invert value is only used for active lakes if couttype for lake outlet outletno is not SPECIFIED. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

rough—real or character value that defines the width of the lake outlet. A specified width value is only used for active lakes if couttype for lake outlet outletno is not SPECIFIED. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

width—real or character value that defines the width of the lake outlet. A specified width value is only used for active lakes if couttype for lake outlet outletno is not SPECIFIED. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

slope—real or character value that defines the bed slope for the lake outlet. A specified slope value is only used for active lakes if couttype for lake outlet outletno is MANNING. If the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

Example Input File

```
begin options
  PRINT_INPUT
  BOUNDNAMES
  PRINT_STAGE
  PRINT_FLOWS
  STAGE FILEOUT lak-1.stage.bin
  BUDGET FILEOUT lak-1.cbc
end options

begin dimensions
  nlakes 1
  noutlets 1
end dimensions

begin packagedata
# lakeno    strt lakeconn boundname
      1 110.00      57 LAKE_1
end packagedata

begin connectiondata
# lakeno iconn layer row column claketype bedleak belev telev connlen connwidth
      1     1     2 10    11    VERTICAL   0.1    0     0       0         0
      1     2     2 10    11    VERTICAL   0.1    0     0       0         0
      1     3     2 10    11    VERTICAL   0.1    0     0       0         0
      1     4     2 10    11    VERTICAL   0.1    0     0       0         0
      1     5     2 10    11    VERTICAL   0.1    0     0       0         0
      1     6     2 10    11    VERTICAL   0.1    0     0       0         0
      1     7     3 10    11    VERTICAL   0.1    0     0       0         0
      1     8     3 10    11    VERTICAL   0.1    0     0       0         0
      1     9     3 10    11    VERTICAL   0.1    0     0       0         0
      1    10     2 10    11    VERTICAL   0.1    0     0       0         0
      1    11     2 10    11    VERTICAL   0.1    0     0       0         0
      1    12     3 10    11    VERTICAL   0.1    0     0       0         0
      1    13     3 10    11    VERTICAL   0.1    0     0       0         0
      1    14     3 10    11    VERTICAL   0.1    0     0       0         0
      1    15     2 10    11    VERTICAL   0.1    0     0       0         0
      1    16     2 10    11    VERTICAL   0.1    0     0       0         0
      1    17     3 10    11    VERTICAL   0.1    0     0       0         0
      1    18     3 10    11    VERTICAL   0.1    0     0       0         0
      1    19     3 10    11    VERTICAL   0.1    0     0       0         0
      1    20     2 10    11    VERTICAL   0.1    0     0       0         0
      1    21     2 10    11    VERTICAL   0.1    0     0       0         0
      1    22     2 10    11    VERTICAL   0.1    0     0       0         0
      1    23     2 10    11    VERTICAL   0.1    0     0       0         0
      1    24     2 10    11    VERTICAL   0.1    0     0       0         0
      1    25     2 10    11    VERTICAL   0.1    0     0       0         0
      1    26     1  6     7 HORIZONTAL   0.1    0     0     500       500
      1    27     1  7     6 HORIZONTAL   0.1    0     0     500       500
      1    28     1  6     8 HORIZONTAL   0.1    0     0     500       500
      1    29     1  6     9 HORIZONTAL   0.1    0     0     500       500
      1    30     1  6    10 HORIZONTAL   0.1    0     0     500       500
      1    31     1  6    11 HORIZONTAL   0.1    0     0     500       500
      1    32     1  7    12 HORIZONTAL   0.1    0     0     500       500
      1    33     1  8     6 HORIZONTAL   0.1    0     0     500       500
      1    34     1  8    12 HORIZONTAL   0.1    0     0     500       500
      1    35     1  9     6 HORIZONTAL   0.1    0     0     500       500
      1    36     1  9    12 HORIZONTAL   0.1    0     0     500       500
      1    37     1 10     6 HORIZONTAL   0.1    0     0     500       500
      1    38     1 10    12 HORIZONTAL   0.1    0     0     500       500
```

```
    1   39    1  12     7  HORIZONTAL    0.1    0    0    500    500
    1   40    1  11     6  HORIZONTAL    0.1    0    0    500    500
    1   41    1  12     8  HORIZONTAL    0.1    0    0    500    500
    1   42    1  12     9  HORIZONTAL    0.1    0    0    500    500
    1   43    1  12    10  HORIZONTAL    0.1    0    0    500    500
    1   44    1  12    11  HORIZONTAL    0.1    0    0    500    500
    1   45    1  11    12  HORIZONTAL    0.1    0    0    500    500
    1   46    2   7     8  HORIZONTAL    0.1    0    0    500    500
    1   47    2   8     7  HORIZONTAL    0.1    0    0    500    500
    1   48    2   7     9  HORIZONTAL    0.1    0    0    500    500
    1   49    2   7    10  HORIZONTAL    0.1    0    0    500    500
    1   50    2   8    11  HORIZONTAL    0.1    0    0    500    500
    1   51    2   9     7  HORIZONTAL    0.1    0    0    500    500
    1   52    2   9    11  HORIZONTAL    0.1    0    0    500    500
    1   53    2  11     8  HORIZONTAL    0.1    0    0    500    500
    1   54    2  10     7  HORIZONTAL    0.1    0    0    500    500
    1   55    2  11     9  HORIZONTAL    0.1    0    0    500    500
    1   56    2  11    10  HORIZONTAL    0.1    0    0    500    500
    1   57    2  10    11  HORIZONTAL    0.1    0    0    500    500
end connectiondata

begin outlets
# outletno lakein lakeout  couttype invert  width  rough  slope
      1       1       0  SPECIFIED     0      0      0      0
end outlets

begin period 1
 1 rainfall 0.0116
 1 evaporation 0.0103
 1 rate -100.
end period

begin period 100
  1 constant
  1 rate 0.
end period
```

Available observation types

Lake Package observations include lake stage and all of the terms that contribute to the continuity equation for each lake. Additional LAK Package observations include flow rates for individual outlets, lakes, or groups of lakes (`outlet`); the lake volume (`volume`); lake surface area (`surface-area`); wetted area for a lake-aquifer connection (`wetted-area`); and the conductance for a lake-aquifer connection conductance (`conductance`). The data required for each LAK Package observation type is defined in table 16. Negative and positive values for `lak` observations represent a loss from and gain to the GWF model, respectively. For all other flow terms, negative and positive values represent a loss from and gain from the LAK package, respectively.

Table 16: Available LAK Package observation types

Stress Package	Observation type	ID	ID2	Description
LAK	stage	lakeno or boundname	–	Surface-water stage in a lake. If bound-name is specified, boundname must be unique for each lake.

Table 16. Available LAK Package observation types.—Continued

Stress Package	Observation type	ID	ID2	Description
LAK	ext-inflow	lakeno or boundname	–	Specified inflow into a lake or group of lakes.
LAK	outlet-inflow	lakeno or boundname	–	Simulated inflow from upstream lake outlets into a lake or group of lakes.
LAK	inflow	lakeno or boundname	–	Sum of specified inflow and simulated inflow from upstream lake outlets into a lake or group of lakes.
LAK	from-mvr	lakeno or boundname	–	Inflow into a lake or group of lakes from the MVR package.
LAK	rainfall	lakeno or boundname	–	Rainfall rate applied to a lake or group of lakes.
LAK	runoff	lakeno or boundname	–	Runoff rate applied to a lake or group of lakes.
LAK	lak	lakeno or boundname	`iconn` or –	Simulated flow rate for a lake or group of lakes and its aquifer connection(s). If boundname is not specified for ID, then the simulated lake-aquifer flow rate at a specific lake connection is observed. In this case, ID2 must be specified and is the connection number `iconn`.
LAK	withdrawal	lakeno or boundname	–	Specified withdrawal rate from a lake or group of lakes.
LAK	evaporation	lakeno or boundname	–	Simulated evaporation rate from a lake or group of lakes.
LAK	ext-outflow	outletno or boundname	–	External outflow from a lake outlet, a lake, or a group of lakes to an external boundary. If boundname is not specified for ID, then the external outflow from a specific lake outlet is observed. In this case, ID is the outlet number outletno.
LAK	to-mvr	outletno or boundname	–	Outflow from a lake outlet, a lake, or a group of lakes that is available for the MVR package. If boundname is not specified for ID, then the outflow available for the MVR package from a specific lake outlet is observed. In this case, ID is the outlet number outletno.
LAK	storage	lakeno or boundname	–	Simulated storage flow rate for a lake or group of lakes.
LAK	constant	lakeno or boundname	–	Simulated constant-flow rate for a lake or group of lakes.

Table 16. Available LAK Package observation types.—Continued

Stress Package	Observation type	ID	ID2	Description
LAK	outlet	outletno or boundname	–	Simulate outlet flow rate from a lake outlet, a lake, or a group of lakes. If boundname is not specified for ID, then the flow from a specific lake outlet is observed. In this case, ID is the outlet number outletno.
LAK	volume	lakeno or boundname	–	Simulated lake volume or group of lakes.
LAK	surface-area	lakeno or boundname	–	Simulated surface area for a lake or group of lakes.
LAK	wetted-area	lakeno or boundname	iconn or –	Simulated wetted-area for a lake or group of lakes and its aquifer connection(s). If boundname is not specified for ID, then the wetted area of a specific lake connection is observed. In this case, ID2 must be specified and is the connection number iconn.
LAK	conductance	lakeno or boundname	iconn or –	Calculated conductance for a lake or group of lakes and its aquifer connection(s). If boundname is not specified for ID, then the calculated conductance of a specific lake connection is observed. In this case, ID2 must be specified and is the connection number iconn.

Example Observation Input File

```
BEGIN OPTIONS
  PRECISION SINGLE
  DIGITS 10
  PRINT_INPUT
END OPTIONS

BEGIN CONTINUOUS FILEOUT my_model.lak.csv
  l1stage stage 1
  l1vol    volume 1
  vflow    lak    1 1
  hflow1   lak    1 2
  hflow2   lak    1 3
  hflow3   lak    1 4
  hflow4   lak    1 5
  lakflow  lak    lake_1
END CONTINUOUS
```

Lake Table Input File

Lake tables of stage, volume, and surface area can be specified for individual lakes. Lake tables are specified by including file names in the LAKE_TABLES block of the LAK Package. These file names correspond to a lake table input file. The format of the lake table input file is described here. All single valued variables are free format.

Structure of Blocks

```
BEGIN DIMENSIONS
  NROW <nrow>
  NCOL <ncol>
END DIMENSIONS
```

```
BEGIN TABLE
  <stage> <volume> <sarea> [<barea>]
  <stage> <volume> <sarea> [<barea>]
  ...
END TABLE
```

Explanation of Variables

nrow—integer value specifying the number of rows in the lake table. There must be nrow rows of data in the TABLE block.

ncol—integer value specifying the number of colums in the lake table. There must be ncol columns of data in the TABLE block. For lakes with HORIZONTAL and/or VERTICAL ctype connections, NROW must be equal to 3. For lakes with EMBEDDEDH or EMBEDDEDV ctype connections, NROW must be equal to 4.

stage—real value that defines the stage corresponding to the remaining data on the line.

volume—real value that defines the lake volume corresponding to the stage specified on the line.

sarea—real value that defines the lake surface area corresponding to the stage specified on the line.

barea—real value that defines the lake-GWF exchange area corresponding to the stage specified on the line. barea is only specified if the claktype for the lake is EMBEDDEDH or EMBEDDEDV.

Example Input File

```
begin dimensions
  nrow 11
  ncol 3
end dimensions

begin table
# stage    volume     sarea
     0        0.        0.
     1        0.5       1.
```

```
    2      1.0      2.
    3      2.0      2.
    4      3.0      2.
    5      4.0      2.
    6      5.0      2.
    7      6.0      2.
    8      7.0      2.
    9      8.0      2.
   10      9.0      2.
end table
```

Unsaturated Zone Flow (UZF) Package

Input to the Unsaturated Zone Flow (UZF) Package is read from the file that has type "UZF6" in the Name File. All single valued variables are free format.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [AUXILIARY <auxiliary(naux)>]
  [AUXMULTNAME <auxmultname>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [BUDGET FILEOUT <budgetfile>]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
  [MOVER]
  [SIMULATE_ET]
  [LINEAR_GWET]
  [SQUARE_GWET]
  [SIMULATE_GWSEEP]
  [UNSAT_ETWC]
  [UNSAT_ETAE]
END OPTIONS
```

```
BEGIN DIMENSIONS
  NUZFCELLS <nuzfcells>
  NTRAILWAVES <ntrailwaves>
  NWAVESETS <nwavesets>
END DIMENSIONS
```

```
BEGIN PACKAGEDATA
  <cellid(ncelldim)> <landflag> <ivertcon> <surfdep> <vks> <thtr> <thts> <thti> <eps> [<boundname>]
  <cellid(ncelldim)> <landflag> <ivertcon> <surfdep> <vks> <thtr> <thts> <thti> <eps> [<boundname>]
  ...
END PACKAGEDATA
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <uzfno> <finf> <pet> <extdp> <extwc> <ha> <hroot> <rootact> [<aux(naux)>]
  <uzfno> <finf> <pet> <extdp> <extwc> <ha> <hroot> <rootact> [<aux(naux)>]
  ...
END PERIOD
```

Explanation of Variables

auxiliary—defines an array of one or more auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided on this line; however, lists of information provided in subsequent blocks must have a column of data for each auxiliary variable name defined here. The

number of auxiliary variables detected on this line determines the value for naux. Comments cannot be provided anywhere on this line as they will be interpreted as auxiliary variable names. Auxiliary variables may not be used by the package, but they will be available for use by other parts of the program. The program will terminate with an error if auxiliary variables are specified on more than one line in the options block.

auxmultname—name of auxiliary variable to be used as multiplier of GWF cell area used by UZF cell.

BOUNDNAMES—keyword to indicate that boundary names may be provided with the list of UZF cells.

PRINT_INPUT—keyword to indicate that the list of UZF information will be written to the listing file immediately after it is read.

PRINT_FLOWS—keyword to indicate that the list of UZF flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and PRINT_FLOWS is specified, then flow rates are printed for the last time step of each stress period.

SAVE_FLOWS—keyword to indicate that UZF flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

BUDGET—keyword to specify that record corresponds to the budget.

FILEOUT—keyword to specify that an output filename is expected next.

budgetfile—name of the binary output file to write budget information.

TS6—keyword to specify that record corresponds to a time-series file.

FILEIN—keyword to specify that an input filename is expected next.

ts6_filename—defines a time-series file defining time series that can be used to assign time-varying values. See the "Time-Variable Input" section for instructions on using the time-series capability.

OBS6—keyword to specify that record corresponds to an observations file.

obs6_filename—name of input file to define observations for the UZF package. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the UZF package.

MOVER—keyword to indicate that this instance of the UZF Package can be used with the Water Mover (MVR) Package. When the MOVER option is specified, additional memory is allocated within the package to store the available, provided, and received water.

SIMULATE_ET—keyword specifying that ET in the unsaturated (UZF) and saturated zones (GWF) will be simulated. ET can be simulated in the UZF cell and not the GWF cell by emitting keywords LINEAR_GWET and SQUARE_GWET.

LINEAR_GWET—keyword specifying that groundwater ET will be simulated using the original ET formulation of MODFLOW-2005.

SQUARE_GWET—keyword specifying that groundwater ET will be simulated by assuming a constant ET rate for groundwater levels between land surface (TOP) and land surface minus the ET extinction depth (TOP-EXTDP). Groundwater ET is smoothly reduced from the PET rate to zero over a nominal interval at TOP-EXTDP.

SIMULATE_GWSEEP—keyword specifying that groundwater discharge (GWSEEP) to land surface will be simulated. Groundwater discharge is nonzero when groundwater head is greater than land surface.

UNSAT_ETWC—keyword specifying that ET in the unsaturated zone will be simulated as a function of the specified PET rate while the water content (THETA) is greater than the ET extinction water content (extwc).

UNSAT_ETAE—keyword specifying that ET in the unsaturated zone will be simulated simulated using a capillary pressure based formulation. Capillary pressure is calculated using the Brooks-Corey retention function.

nuzfcells—is the number of UZF cells. More than 1 UZF cell can be assigned to a GWF cell; however, only 1 GWF cell can be assigned to a single UZF cell. If the MULTILAYER option is used then UZF cells can be assigned to GWF cells below (in deeper layers than) the upper most active GWF cells.

ntrailwaves—is the number of trailing waves. NTRAILWAVES has a default value of 7 and can be increased to lower mass balance error in the unsaturated zone.

nwavesets—is the number of UZF cells specified. NWAVSETS has a default value of 40 and can be increased if more waves are required to resolve variations in water content within the unsaturated zone.

cellid—is the cell identifier, and depends on the type of grid that is used for the simulation. For a structured grid that uses the DIS input file, cellid is the layer, row, and column. For a grid that uses the DISV input file, cellid is the layer and cell2d number. If the model uses the unstructured discretization (DISU) input file, then cellid is the node number for the cell.

landflag—integer value set to one for land surface cells indicating that boundary conditions can be applied and data can be specified in the PERIOD block. A value of 0 specifies a non-land surface cell.

ivertcon—integer value set to specify underlying UZF cell that receives water flowing to bottom of cell. If unsaturated zone flow reaches water table before the cell bottom then water is added to GWF cell instead of flowing to underlying UZF cell. A value of 0 indicates the UZF cell is not connected to an underlying UZF cell.

surfdep—is the surface depression depth of the UZF cell.

vks—is the vertical saturated hydraulic conductivity of the UZF cell.

thtr—is the residual (irreducible) water content of the UZF cell.

thts—is the saturated water content of the UZF cell.

thti—is the initial water content of the UZF cell.

eps—is the epsilon exponent of the UZF cell.

boundname—name of the UZF cell cell. boundname is an ASCII character variable that can contain as many as 40 characters. If boundname contains spaces in it, then the entire name must be enclosed within single quotes.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. `iper` must be less than `nper` in the TDIS Package and greater than zero. The `iper` value assigned to a stress period block must be greater than the `iper` value assigned for the previous block.

uzfno—is the UZF id number as determined by the order of information specified in the PACKAGE-DATA block.

finf—real or character value that defines the applied infiltration rate of the UZF cell (LT^{-1}). If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

pet—real or character value that defines the potential evapotranspiration rate of the UZF cell and specified GWF cell. Evapotranspiration is first removed from the unsaturated zone and any remaining potential evapotranspiration is applied to the saturated zone. If `ivertcon` is greater than zero then residual potential evapotranspiration not satisfied in the UZF cell is applied to the underlying UZF and GWF cells. `pet` is always specified, but is only used if `SIMULATE_ET` is specified in the OPTIONS block. If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

extdp—real or character value that defines the evapotranspiration extinction depth of the UZF cell. If `ivertcon` is greater than zero and `extdp` extends below the GWF cell bottom then remaining potential evapotranspiration is applied to the underlying UZF and GWF cells. `extdp` is always specified, but is only used if `SIMULATE_ET` is specified in the OPTIONS block. If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

extwc—real or character value that defines the evapotranspiration extinction water content of the UZF cell. `extwc` is always specified, but is only used if `SIMULATE_ET` and `UNSAT_ETWC` are specified in the OPTIONS block. If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

ha—real or character value that defines the air entry potential (head) of the UZF cell. `ha` is always specified, but is only used if `SIMULATE_ET` and `UNSAT_ETAE` are specified in the OPTIONS block. If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

hroot—real or character value that defines the root potential (head) of the UZF cell. `hroot` is always specified, but is only used if `SIMULATE_ET` and `UNSAT_ETAE` are specified in the OPTIONS block. If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

rootact—real or character value that defines the root activity function of the UZF cell. `rootact` is the length of roots in a given volume of soil divided by that volume. Values range from 0 to about 3 cm^{-2}, depending on the plant community and its stage of development. `rootact` is always specified, but is only used if `SIMULATE_ET` and `UNSAT_ETAE` are specified in the OPTIONS block. If the Options block includes a `TIMESERIESFILE` entry (see the "Time-Variable Input" section),

values can be obtained from a time series by entering the time-series name in place of a numeric value.

aux—represents the values of the auxiliary variables for each UZF. The values of auxiliary variables must be present for each UZF. The values must be specified in the order of the auxiliary variables specified in the OPTIONS block. If the package supports time series and the Options block includes a TIMESERIESFILE entry (see the "Time-Variable Input" section), values can be obtained from a time series by entering the time-series name in place of a numeric value.

Example Input File

```
BEGIN OPTIONS
  OBS6 UZF.obs
  SIMULATE_ET
  UNSAT_ETWC
  LINEAR_GWET
END OPTIONS

BEGIN DIMENSIONS
  NUZFCELLS   10
  NTRAILWAVES 7
  NWAVESETS   40
END DIMENSIONS

BEGIN PACKAGEDATA
  1 1 1 1 1.0 1.0 0.05 0.35 0.1 4.0
  1 1 2 1 1.0 1.0 0.05 0.35 0.1 4.0
  1 1 3 1 1.0 1.0 0.05 0.35 0.1 4.0
  1 1 4 1 1.0 1.0 0.05 0.35 0.1 4.0
  1 1 5 1 1.0 1.0 0.05 0.35 0.1 4.0
  1 1 6 1 1.0 1.0 0.05 0.35 0.1 4.0
  1 1 7 1 1.0 1.0 0.05 0.35 0.1 4.0
  1 1 8 1 1.0 1.0 0.05 0.35 0.1 4.0
  1 1 9 1 1.0 1.0 0.05 0.35 0.1 4.0
  1 1 10 1 1.0 1.0 0.05 0.35 0.1 4.0
END PACKAGEDATA

BEGIN PERIOD 1
  2 0.00005 0.00002 2.0 0.10
  3 0.00008 0.00002 2.0 0.10
  4 0.00009 0.00002 2.0 0.10
  5 0.0001 0.00002 2.0 0.10
  6 0.0001 0.00002 2.0 0.10
  7 0.00009 0.00002 2.0 0.10
  8 0.00008 0.00002 2.0 0.10
  9 0.00005 0.00002 2.0 0.10
END PERIOD

BEGIN PERIOD 2
  2 0.00009 0.00003 2.0 0.10
  3 0.0001 0.00003 2.0 0.10
  4 0.0001 0.00003 2.0 0.10
  5 0.00015 0.00003 2.0 0.10
  6 0.00015 0.00003 2.0 0.10
  7 0.0001 0.00003 2.0 0.10
```

```
  8 0.0001 0.00003 2.0 0.10
  9 0.00009 0.00003 2.0 0.10
END PERIOD
```

Available observation types

Unsaturated Zone Flow Package observations include all exchange terms with the GWF model and all of the terms that contribute to the continuity equation for each UZF cell. Additional UZF Package observations include the net infiltration into UZF cells in land-surface cells (`net-infiltration`) and the water content in UZF cells a specified depth below the top of a UZF cell (`water-content`). The data required for each UZF Package observation type is defined in table 17. Negative and positive values for `uzf-gwrch`, `uzf-gwd`, `uzf-gwd-to-mvr`, and `uzf-gwet` observations represent a loss from and gain to the GWF model, respectively. For all other flow terms, negative and positive values represent a loss from and gain from the UZF package, respectively.

Table 17: Available UZF Package observation types

Stress Package	Observation type	ID	ID2	Description
UZF	uzf-gwrch	uzfno or boundname	–	Simulated recharge to the aquifer calculated by the UZF package for a UZF cell or a group of UZF cells.
UZF	uzf-gwd	uzfno or boundname	–	Simulated groundwater discharge to the land surface calculated by the UZF package for a UZF cell or a group of UZF cells.
UZF	uzf-gwd-to-mvr	uzfno or boundname	–	Simulated groundwater discharge to the land surface calculated by the UZF package that is available to the MVR package for a UZF cell or a group of UZF cells.
UZF	uzf-gwet	uzfno or boundname	–	Simulated groundwater evapotranspiration calculated by the UZF package for a UZF cell or a group of UZF cells.
UZF	infiltration	uzfno or boundname	–	Specified infiltration rate applied to a UZF package for a UZF cell or a group of UZF cells with landflag values not equal to zero.
UZF	from-mvr	uzfno or boundname	–	Inflow into a UZF cell from the MVR package for a UZF cell or a group of UZF cells.
UZF	rej-inf	uzfno or boundname	–	Simulated rejected infiltration calculated by the UZF package for a UZF cell or a group of UZF cells.
UZF	rej-inf-to-mvr	uzfno or boundname	–	Simulated rejected infiltration calculated by the UZF package that is available to the MVR package for a UZF cell or a group of UZF cells.
UZF	uzet	uzfno or boundname	–	Simulated unsaturated evapotranspiration calculated by the UZF package for a UZF cell or a group of UZF cells.

Table 17.   Available UZF Package observation types.—Continued

Stress Package	Observation type	ID	ID2	Description
UZF	storage	uzfno or boundname	–	Simulated storage flow rate for a UZF package cell or a group of UZF cells.
UZF	net-infiltration	uzfno or boundname	–	Simulated net infiltration rate for a UZF package cell or a group of UZF cells.
UZF	water-content	uzfno or boundname	depth	Unsaturated-zone water content at a user-specified depth (ID2) relative to the top of GWF cellid for a UZF cell. The user-specified depth must be greater than or equal to zero and less than the thickness of GWF cellid (TOP - BOT). If boundname is specified, boundname must be unique for each UZF cell.

Example Observation Input File

```
BEGIN CONTINUOUS FILEOUT my_model.obs.uzf.csv
  id26_infil   infiltration 26
  id126_infil  infiltration 126
  id26_dpth=20 water-content 26 20.0
  id126_dpth=51 water-content 126 1.0 #depth is below celtop
  id126_rch    uzf-gwrch    126
END CONTINUOUS

BEGIN CONTINUOUS FILEOUT my_model.uzf.budget.uzf.csv
  sinf        infiltration    uzfcells
  frommvr     from-mvr        uzfcells
  rejinf      rej-inf         uzfcells
  rejinftomvr rej-inf-to-mvr uzfcells
  uzet        uzet            uzfcells
  storage     storage         uzfcells
  net-inf     net-infiltration uzfcells
END CONTINUOUS

BEGIN CONTINUOUS FILEOUT my_model.uzf.budget.gwf.csv
  gwrch     uzf-gwrch       uzfcells
  gwd       uzf-gwd         uzfcells
  gwdtomvr  uzf-gwd-to-mvr uzfcells
  gwet      uzf-gwet        uzfcells
END CONTINUOUS
```

Water Mover (MVR) Package

The MVR Package can be used to transfer water from a provider to a receiver. Providers are extraction wells, streamflow routing reaches, lakes and other model features that can be conceptualized as having water available. The list of packages that can provide water to the MVR Package are:

- Well Package

- Drain Package

- River Package

- General-Head Boundary Package

- Multi-Aquifer Well Package

- Streamflow Routing Package

- Unsaturated Zone Flow Package

- Lake Package

Receivers are package features within the model that solve a continuity equation of inflows, outflows, and change in storage. These features include multi-aquifer wells, streamflow routing reaches, lakes, and unsaturated zone flow cells. The list of packages that can receive water is shorter than the provider list, because the WEL, DRN, RIV, and GHB Packages do not represent a continuity equation (boundary stages or elevations are specified by the user). Therefore, the list of packages that can act as receivers are:

- Multi-Aquifer Well Package

- Streamflow Routing Package

- Unsaturated Zone Flow Package

- Lake Package

The program will terminate with an error if the MVR is used with an unsupported package type.

The MVR Package is based on the calculation of available water that can be moved from one package feature to another. The equations used to determine how much water can be transferred are as follows, where Q_P is the flow rate that can be supported by the provider (the available flow rate), and Q_R is the actual rate of water transferred to the receiver.

1. A FACTOR can be specified such that

   $Q_R = \alpha Q_P$

   where α is the factor to convert the provider flow rate to the receiver flow rate.

2. An EXCESS rate can be specified by the user as Q_S such that

$$Q_R = \begin{cases} Q_P - Q_S, & \text{if } Q_P > Q_S \\ 0, & \text{otherwise} \end{cases}$$

   In the EXCESS case, any water that exceeds the user specified rate is provided to the receiver. No water is provided to the receiver if the available water is less than the user specified value.

3. A THRESHOLD rate can be specified for Q_S such that

$$Q_R = \begin{cases} 0, & \text{if } Q_S > Q_P \\ Q_S, & \text{otherwise} \end{cases}$$

In the THRESHOLD case, no flow is provided to the receiver until the available water exceeds the user specified Q_S rate. Once the available water exceeds the user specified rate, then the Q_S rate is provided to the receiver.

4. An UPTO rate can be specified for Q_S such that

$$Q_R = \begin{cases} Q_S, & \text{if } Q_P > Q_S \\ Q_P, & \text{otherwise} \end{cases}$$

In the UPTO case, all of the available water will be taken from the provider up to the Q_S value specified by the user. Once Q_S is exceeded, the receiver will continue to get the Q_S value specified by the user.

In the MVR PERIOD block (as shown below), the user assigns the equation to used for each individual entry by specifying FACTOR, EXCESS, THRESHOLD, or UPTO to the input variable `mvrtype`.

Input to the Water Mover (MVR) Package is read from the file that has type "MVR6" in the Name File. Only one MVR Package can be used per GWF Model. All single valued variables are free format.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [MODELNAMES]
  [BUDGET FILEOUT <budgetfile>]
END OPTIONS
```

```
BEGIN DIMENSIONS
  MAXMVR <maxmvr>
  MAXPACKAGES <maxpackages>
END DIMENSIONS
```

```
BEGIN PACKAGES
  [<mname>] <pname>
  [<mname>] <pname>
  ...
END PACKAGES
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  [<mname1>] <pname1> <id1> [<mname2>] <pname2> <id2> <mvrtype> <value>
  [<mname1>] <pname1> <id1> [<mname2>] <pname2> <id2> <mvrtype> <value>
  ...
END PERIOD
```

Explanation of Variables

PRINT_INPUT—keyword to indicate that the list of MVR information will be written to the listing file immediately after it is read.

PRINT_FLOWS—keyword to indicate that the list of MVR flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and PRINT_FLOWS is specified, then flow rates are printed for the last time step of each stress period.

MODELNAMES—keyword to indicate that all package names will be preceded by the model name for the package. Model names are required when the Mover Package is used with a GWF-GWF Exchange. The MODELNAME keyword should not be used for a Mover Package that is for a single GWF Model.

BUDGET—keyword to specify that record corresponds to the budget.

FILEOUT—keyword to specify that an output filename is expected next.

budgetfile—name of the output file to write budget information.

maxmvr—integer value specifying the maximum number of water mover entries that will specified for any stress period.

maxpackages—integer value specifying the number of unique packages that are included in this water mover input file.

mname—name of model containing the package.

pname—is the name of a package that may be included in a subsequent stress period block.

iper—integer value specifying the starting stress period number for which the data specified in the PERIOD block apply. iper must be less than nper in the TDIS Package and greater than zero. The iper value assigned to a stress period block must be greater than the iper value assigned for the previous block.

mname1—name of model containing the package, pname1.

pname1—is the package name for the provider. The package pname1 must be designated to provide water through the MVR Package by specifying the keyword "MOVER" in its OPTIONS block.

id1—is the identifier for the provider. This is the well number, reach number, lake number, etc.

mname2—name of model containing the package, pname2.

pname2—is the package name for the receiver. The package pname2 must be designated to receive water from the MVR Package by specifying the keyword "MOVER" in its OPTIONS block.

id2—is the identifier for the receiver. This is the well number, reach number, lake number, etc.

mvrtype—is the character string signifying the method for determining how much water will be moved. Supported values are "FACTOR" "EXCESS" "THRESHOLD" and "UPTO". These four options determine how the receiver flow rate, Q_R, is calculated. These options are based the options available in the SFR2 Package for diverting stream flow.

value—is the value to be used in the equation for calculating the amount of water to move. For the
"FACTOR" option, value is the α factor. For the remaining options, value is the specified flow
rate, Q_S.

Example Input File

```
BEGIN OPTIONS
  PRINT_INPUT
  PRINT_FLOWS
END OPTIONS

BEGIN DIMENSIONS
  MAXMVR 16
  MAXPACKAGES 5
END DIMENSIONS

BEGIN PACKAGES
  MAW_1
  MAW_2
  SFR_1
  LAK_1
  UZF_1
END PACKAGES

BEGIN PERIOD 1
# ***PROVIDER*** ***RECEIVER*** ***FLOW INFO**
#  PAK1 PAK1RCH   PAK2 PAK2RCH TYPE     VALUE
  MAW_1       1  MAW_2      21 EXCESS   5.00
  MAW_1      11  SFR_1      77 FACTOR   0.25
  MAW_1      21  UZF_1      93 FACTOR   0.01
  MAW_1      21  LAK_1       3 FACTOR   1.00

  SFR_1    1021  MAW_1      21 THRESHOLD 10.0
  SFR_1     441  SFR_1      77 FACTOR   0.10
  SFR_1      56  UZF_1      93 FACTOR   0.10
  SFR_1    4587  LAK_1       3 FACTOR   1.00

  UZF_1       4  MAW_1      11 FACTOR   1.00
  UZF_1       5  SFR_1      22 FACTOR   1.00
  UZF_1       6  UZF_1      45 FACTOR   1.00
  UZF_1       7  LAK_1       3 FACTOR   1.00

  LAK_1       1  MAW_1      11 EXCESS   1000.
  LAK_1       2  SFR_1      22 UPTO     2000.
  LAK_1       3  UZF_1      45 UPTO     3000.
  LAK_1       4  LAK_1       3 UPTO     3000.
END PERIOD 1
```

Ghost-Node Correction (GNC) Package

Input to the Ghost-Node Correction (GNC) Package is read from the file that has type "GNC6" in the Name File. Only one GNC Package can be used per GWF Model. All single valued variables are free format.

The GNC Package has two options for adding the correction terms to the system of equations. The implicit option, which is the default, adds the terms on both the left-hand and right-hand sides of the equations. When this default option is used, the BICGSTAB linear acceleration option should be specified within the LINEAR block of the Sparse Matrix Solver. The BICGSTAB acceleration option is designed to handle the asymmetry in the conductance matrix. When the EXPLICIT option is specified for the GNC Package, then the correction terms are added to the right-hand side, and either the CG or BICGSTAB acceleration methods can be used.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [EXPLICIT]
END OPTIONS
```

```
BEGIN DIMENSIONS
  NUMGNC <numgnc>
  NUMALPHAJ <numalphaj>
END DIMENSIONS
```

```
BEGIN GNCDATA
  <cellidn> <cellidm> <cellidsj(numalphaj)> <alphasj(numalphaj)>
  <cellidn> <cellidm> <cellidsj(numalphaj)> <alphasj(numalphaj)>
  ...
END GNCDATA
```

Explanation of Variables

PRINT_INPUT—keyword to indicate that the list of GNC information will be written to the listing file immediately after it is read.

PRINT_FLOWS—keyword to indicate that the list of GNC flow rates will be printed to the listing file for every stress period time step in which "BUDGET PRINT" is specified in Output Control. If there is no Output Control option and PRINT_FLOWS is specified, then flow rates are printed for the last time step of each stress period.

EXPLICIT—keyword to indicate that the ghost node correction is applied in an explicit manner on the right-hand side of the matrix. The explicit approach will likely require additional outer iterations. If the keyword is not specified, then the correction will be applied in an implicit manner on the left-hand side. The implicit approach will likely converge better, but may require additional memory. If the EXPLICIT keyword is not specified, then the BICGSTAB linear acceleration option should be specified within the LINEAR block of the Sparse Matrix Solver.

numgnc—is the number of GNC entries.

numalphaj—is the number of contributing factors.

cellidn—is the cellid of the cell, n, in which the ghost node is located. For a structured grid that uses the DIS input file, cellidn is the layer, row, and column numbers of the cell. For a grid that uses the DISV input file, cellidn is the layer number and cell2d number for the two cells. If the model uses the unstructured discretization (DISU) input file, then cellidn is the node number for the cell.

cellidm—is the cellid of the connecting cell, m, to which flow occurs from the ghost node. For a structured grid that uses the DIS input file, cellidm is the layer, row, and column numbers of the cell. For a grid that uses the DISV input file, cellidm is the layer number and cell2d number for the two cells. If the model uses the unstructured discretization (DISU) input file, then cellidm is the node number for the cell.

cellidsj—is the array of cellids for the contributing j cells, which contribute to the interpolated head value at the ghost node. This item contains one cellid for each of the contributing cells of the ghost node. Note that if the number of actual contributing cells needed by the user is less than numalphaj for any ghost node, then a dummy cellid of zero(s) should be inserted with an associated contributing factor of zero. For a structured grid that uses the DIS input file, cellid is the layer, row, and column numbers of the cell. For a grid that uses the DISV input file, cellid is the layer number and cell2d number for the two cells. If the model uses the unstructured discretization (DISU) input file, then cellid is the node number for the cell.

alphasj—is the contributing factors for each contributing node in cellidsj. Note that if the number of actual contributing cells is less than numalphaj for any ghost node, then dummy cellids should be inserted with an associated contributing factor of zero.

Example Input File

```
BEGIN OPTIONS
  PRINT_INPUT
  PRINT_FLOWS
END OPTIONS

BEGIN DIMENSIONS
  NUMGNC 24
  NUMALPHAJ 1
END DIMENSIONS

BEGIN GNCDATA
  10 41 9 0.333333333333
  10 43 11 0.333333333333
  11 44 10 0.333333333333
  11 46 12 0.333333333333
  12 47 11 0.333333333333
  12 49 13 0.333333333333
  16 41 9 0.333333333333
  16 59 20 0.333333333333
  17 49 13 0.333333333333
  17 67 21 0.333333333333
```

```
 20  68  16 0.333333333333
 20  86  24 0.333333333333
 21  76  17 0.333333333333
 21  94  25 0.333333333333
 24  95  20 0.333333333333
 24 113  28 0.333333333333
 25 103  21 0.333333333333
 25 121  32 0.333333333333
 29 113  28 0.333333333333
 29 115  30 0.333333333333
 30 116  29 0.333333333333
 30 118  31 0.333333333333
 31 119  30 0.333333333333
 31 121  32 0.333333333333
END GNCDATA
```

Groundwater Flow (GWF) Exchange

Input to the Groundwater Flow (GWF-GWF) Exchange is read from the file that has type "GWF-GWF" in the Simulation Name File. All single valued variables are free format.

The XT3D capability, which can be used to improve the accuracy of the flow calculation for certain types of cell connections and to represent anisotropic groundwater flow, is not implemented for the GWF-GWF Exchange.

Structure of Blocks

```
BEGIN OPTIONS
  [AUXILIARY <auxiliary(naux)>]
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [CELL_AVERAGING <cell_averaging>]
  [VARIABLECV [DEWATERED]]
  [NEWTON]
  [GNC6 FILEIN <gnc6_filename>]
  [MVR6 FILEIN <mvr6_filename>]
  [OBS6 FILEIN <obs6_filename>]
END OPTIONS
```

```
BEGIN DIMENSIONS
  NEXG <nexg>
END DIMENSIONS
```

```
BEGIN EXCHANGEDATA
  <cellidm1> <cellidm2> <ihc> <cl1> <cl2> <hwva> [<aux(naux)>]
  <cellidm1> <cellidm2> <ihc> <cl1> <cl2> <hwva> [<aux(naux)>]
  ...
END EXCHANGEDATA
```

Explanation of Variables

auxiliary—an array of auxiliary variable names. There is no limit on the number of auxiliary variables that can be provided. Most auxiliary variables will not be used by the GWF-GWF Exchange, but they will be available for use by other parts of the program. If an auxiliary variable with the name "ANGLDEGX" is found, then this information will be used as the angle (provided in degrees) between the connection face normal and the x axis. Additional information on "ANGLDEGX" is provided in the description of the DISU Package.

PRINT_INPUT—keyword to indicate that the list of exchange entries will be echoed to the listing file immediately after it is read.

PRINT_FLOWS—keyword to indicate that the list of exchange flow rates will be printed to the listing file for every stress period in which "SAVE BUDGET" is specified in Output Control.

SAVE_FLOWS—keyword to indicate that cell-by-cell flow terms will be written to the budget file for each model provided that the Output Control for the models are set up with the "BUDGET SAVE FILE" option.

cell_averaging—is a keyword and text keyword to indicate the method that will be used for calculating the conductance for horizontal cell connections. The text value for cell_averaging can be "HARMONIC", "LOGARITHMIC", or "AMT-LMK", which means "arithmetic-mean thickness and logarithmic-mean hydraulic conductivity". If the user does not specify a value for cell_averaging, then the harmonic-mean method will be used.

VARIABLECV—keyword to indicate that the vertical conductance will be calculated using the saturated thickness and properties of the overlying cell and the thickness and properties of the underlying cell. If the DEWATERED keyword is also specified, then the vertical conductance is calculated using only the saturated thickness and properties of the overlying cell if the head in the underlying cell is below its top. If these keywords are not specified, then the default condition is to calculate the vertical conductance at the start of the simulation using the initial head and the cell properties. The vertical conductance remains constant for the entire simulation.

DEWATERED—If the DEWATERED keyword is specified, then the vertical conductance is calculated using only the saturated thickness and properties of the overlying cell if the head in the underlying cell is below its top.

NEWTON—keyword that activates the Newton-Raphson formulation for groundwater flow between connected, convertible groundwater cells. Cells will not dry when this option is used.

FILEIN—keyword to specify that an input filename is expected next.

GNC6—keyword to specify that record corresponds to a ghost-node correction file.

gnc6_filename—is the file name for ghost node correction input file. Information for the ghost nodes are provided in the file provided with these keywords. The format for specifying the ghost nodes is the same as described for the GNC Package of the GWF Model. This includes specifying OPTIONS, DIMENSIONS, and GNCDATA blocks. The order of the ghost nodes must follow the same order as the order of the cells in the EXCHANGEDATA block. For the GNCDATA, noden and all of the nodej values are assumed to be located in model 1, and nodem is assumed to be in model 2.

MVR6—keyword to specify that record corresponds to a mover file.

mvr6_filename—is the file name of the water mover input file to apply to this exchange. Information for the water mover are provided in the file provided with these keywords. The format for specifying the water mover information is the same as described for the Water Mover (MVR) Package of the GWF Model, with two exceptions. First, in the PACKAGES block, the model name must be included as a separate string before each package. Second, the appropriate model name must be included before pname1 and pname2 in the BEGIN PERIOD block. This allows providers and receivers to be located in both models listed as part of this exchange.

OBS6—keyword to specify that record corresponds to an observations file.

obs6_filename—is the file name of the observations input file for this exchange. See the "Observation utility" section for instructions for preparing observation input files. Table 20 lists observation type(s) supported by the GWF-GWF package.

nexg—keyword and integer value specifying the number of GWF-GWF exchanges.

cellidm1—is the cellid of the cell in model 1 as specified in the simulation name file. For a structured grid that uses the DIS input file, **cellidm1** is the layer, row, and column numbers of the cell. For a grid that uses the DISV input file, **cellidm1** is the layer number and cell2d number for the two cells. If the model uses the unstructured discretization (DISU) input file, then **cellidm1** is the node number for the cell.

cellidm2—is the cellid of the cell in model 2 as specified in the simulation name file. For a structured grid that uses the DIS input file, **cellidm2** is the layer, row, and column numbers of the cell. For a grid that uses the DISV input file, **cellidm2** is the layer number and cell2d number for the two cells. If the model uses the unstructured discretization (DISU) input file, then **cellidm2** is the node number for the cell.

ihc—is an integer flag indicating the direction between node n and all of its m connections. If $ihc = 0$ then the connection is vertical. If $ihc = 1$ then the connection is horizontal. If $ihc = 2$ then the connection is horizontal for a vertically staggered grid.

cl1—is the distance between the center of cell nodem1 and the its shared face with nodem2.

cl2—is the distance between the center of cell nodem2 and the its shared face with nodem1.

hwva—is the horizontal width of the flow connection between **nodem1** and **nodem2** if **ihc** > 0, or it is the area of the vertical connection between **nodem1** and **nodem2** if **ihc** $= 0$.

aux—represents the values of the auxiliary variables for each GWFGWF Exchange. The values of auxiliary variables must be present for each exchange. The values must be specified in the order of the auxiliary variables specified in the OPTIONS block.

Example Input File

```
BEGIN OPTIONS
  PRINT_INPUT
  PRINT_FLOWS
  SAVE_FLOWS
  AUXILIARY testaux
  GNC6 FILEIN simulation.gnc
  MVR6 FILEIN simulation.mvr
END OPTIONS

BEGIN DIMENSIONS
  NEXG 36
END DIMENSIONS

  # nodem1 nodem2 ihc cl1   cl2    fahl  testaux
BEGIN EXCHANGEDATA
#
#      left side
        16      1   1  50. 16.67 33.33 100.99
        16     10   1  50. 16.67 33.33 100.99
        16     19   1  50. 16.67 33.33 100.99
        23     28   1  50. 16.67 33.33 100.99
        23     37   1  50. 16.67 33.33 100.99
        23     46   1  50. 16.67 33.33 100.99
```

```
      30      55    1  50. 16.67 33.33 100.99
      30      64    1  50. 16.67 33.33 100.99
      30      73    1  50. 16.67 33.33 100.99
#
#     right side
      20       9    1  50. 16.67 33.33 100.99
      20      18    1  50. 16.67 33.33 100.99
      20      27    1  50. 16.67 33.33 100.99
      27      36    1  50. 16.67 33.33 100.99
      27      45    1  50. 16.67 33.33 100.99
      27      54    1  50. 16.67 33.33 100.99
      34      63    1  50. 16.67 33.33 100.99
      34      72    1  50. 16.67 33.33 100.99
      34      81    1  50. 16.67 33.33 100.99
#
#     back
      10       1    1  50. 17.67 33.33 100.99
      10       2    1  50. 17.67 33.33 100.99
      10       3    1  50. 17.67 33.33 100.99
      11       4    1  50. 17.67 33.33 100.99
      11       5    1  50. 17.67 33.33 100.99
      11       6    1  50. 17.67 33.33 100.99
      12       7    1  50. 17.67 33.33 100.99
      12       8    1  50. 17.67 33.33 100.99
      12       9    1  50. 17.67 33.33 100.99
#
#     front
      38      73    1  50. 17.67 33.33 100.99
      38      74    1  50. 17.67 33.33 100.99
      38      75    1  50. 17.67 33.33 100.99
      39      76    1  50. 17.67 33.33 100.99
      39      77    1  50. 17.67 33.33 100.99
      39      78    1  50. 17.67 33.33 100.99
      40      79    1  50. 17.67 33.33 100.99
      40      80    1  50. 17.67 33.33 100.99
      40      81    1  50. 17.67 33.33 100.99
END EXCHANGEDATA
```

Available observation types

GWF-GWF Exchange observations include the simulated flow between two connected nodes (`flow-ja-face`). The data required for each GWF-GWF Exchange observation type is defined in table 18. For `flow-ja-face` observation types, negative and positive values represent a loss from and gain to the `cellid` specified for ID, respectively.

Table 18: Available GWF-GWF Exchange observation types

Exchange	Observation type	ID	ID2	Description
GWF-GWF	flow-ja-face	cellid	cellid	Flow rate for specified exchange.

Example Observation Input File

```
BEGIN OPTIONS
  PRECISION SINGLE
  DIGITS 10
  PRINT_INPUT
END OPTIONS


# Block defining continuous observations
BEGIN CONTINUOUS FILEOUT simulation.obs.csv
# obsname        obstype
  exg1           flow-ja-face  1
END CONTINUOUS
```

Iterative Model Solution

An iterative model solution (IMS) is specified within the SOLUTIONGROUP block in the simulation name file. The model solution will solve all of the models that are added to it, as specified in the simulation name file, and will include Numerical Exchanges, if they are present. The iterative model solution requires specification of both nonlinear and linear settings.

Structure of Blocks

```
BEGIN OPTIONS
  [PRINT_OPTION <print_option>]
  [COMPLEXITY <complexity>]
  [CSV_OUTPUT FILEOUT <csvfile>]
END OPTIONS
```

```
BEGIN NONLINEAR
  OUTER_HCLOSE <outer_hclose>
  OUTER_MAXIMUM <outer_maximum>
  [UNDER_RELAXATION <under_relaxation>]
  [UNDER_RELAXATION_THETA <under_relaxation_theta>]
  [UNDER_RELAXATION_KAPPA <under_relaxation_kappa>]
  [UNDER_RELAXATION_GAMMA <under_relaxation_gamma>]
  [UNDER_RELAXATION_MOMENTUM <under_relaxation_momentum>]
  [BACKTRACKING_NUMBER <backtracking_number>]
  [BACKTRACKING_TOLERANCE <backtracking_tolerance>]
  [BACKTRACKING_REDUCTION_FACTOR <backtracking_reduction_factor>]
  [BACKTRACKING_RESIDUAL_LIMIT <backtracking_residual_limit>]
END NONLINEAR
```

```
BEGIN LINEAR
  INNER_MAXIMUM <inner_maximum>
  INNER_HCLOSE <inner_hclose>
  INNER_RCLOSE <inner_rclose> [<rclose_option>]
  LINEAR_ACCELERATION <linear_acceleration>
  [RELAXATION_FACTOR <relaxation_factor>]
  [PRECONDITIONER_LEVELS <preconditioner_levels>]
  [PRECONDITIONER_DROP_TOLERANCE <preconditioner_drop_tolerance>]
  [NUMBER_ORTHOGONALIZATIONS <number_orthogonalizations>]
  [SCALING_METHOD <scaling_method>]
  [REORDERING_METHOD <reordering_method>]
END LINEAR
```

Explanation of Variables

print_option—is a flag that controls printing of convergence information from the solver. NONE means print nothing. SUMMARY means print only the total number of iterations and nonlinear residual reduction summaries. ALL means print linear matrix solver convergence information to the solution listing file and model specific linear matrix solver convergence information to each model listing file in addition to SUMMARY information. NONE is default if print_option is not specified.

complexity—is an optional keyword that defines default non-linear and linear solver parameters. SIMPLE - indicates that default solver input values will be defined that work well for nearly linear

models. This would be used for models that do not include nonlinear stress packages and models that are either confined or consist of a single unconfined layer that is thick enough to contain the water table within a single layer. `MODERATE` - indicates that default solver input values will be defined that work well for moderately nonlinear models. This would be used for models that include nonlinear stress packages and models that consist of one or more unconfined layers. The `MODERATE` option should be used when the `SIMPLE` option does not result in successful convergence. `COMPLEX` - indicates that default solver input values will be defined that work well for highly nonlinear models. This would be used for models that include nonlinear stress packages and models that consist of one or more unconfined layers representing complex geology and surface-water/groundwater interaction. The `COMPLEX` option should be used when the `MODERATE` option does not result in successful convergence. Non-linear and linear solver parameters assigned using a specified complexity can be modified in the `NONLINEAR` and `LINEAR` blocks. If the `complexity` option is not specified, `NONLINEAR` and `LINEAR` variables will be assigned the simple complexity values.

`CSV_OUTPUT`—keyword to specify that the record corresponds to the comma separated values solver convergence output.

`FILEOUT`—keyword to specify that an output filename is expected next.

`csvfile`—name of the ascii comma separated values output file to write solver convergence information. If `PRINT_OPTION` is `NONE` or `SUMMARY`, comma separated values output includes maximum head change convergence information at the end of each outer iteration for each time step. If `PRINT_OPTION` is `ALL`, comma separated values output includes maximum head change and maximum residual convergence information for the solution and each model (if the solution includes more than one model) and linear acceleration information for each inner iteration.

`outer_hclose`—real value defining the head change criterion for convergence of the outer (nonlinear) iterations, in units of length. When the maximum absolute value of the head change at all nodes during an iteration is less than or equal to `outer_hclose`, iteration stops. Commonly, `outer_hclose` equals 0.01.

`outer_maximum`—integer value defining the maximum number of outer (nonlinear) iterations – that is, calls to the solution routine. For a linear problem `outer_maximum` should be 1.

`under_relaxation`—is an optional keyword that defines the nonlinear under-relaxation schemes used. By default under-relaxation is not used. `NONE` - under-relaxation is not used. `SIMPLE` - Simple under-relaxation scheme with a fixed relaxation factor is used. `COOLEY` - Cooley under-relaxation scheme is used. `DBD` - delta-bar-delta under-relaxation is used. Note that the under-relaxation schemes are used in conjunction with problems that use the Newton-Raphson formulation, however, experience has indicated that the Cooley under-relaxation and damping work well also for the Picard scheme with the wet/dry options of MODFLOW 6.

`under_relaxation_theta`—real value defining the reduction factor for the learning rate (under-relaxation term) of the delta-bar-delta algorithm. The value of `under_relaxation_theta` is between zero and one. If the change in the variable (head) is of opposite sign to that of the previous iteration, the under-relaxation term is reduced by a factor of `under_relaxation_theta`. The value usually ranges from 0.3 to 0.9; a value of 0.7 works well for most problems. `under_relaxation_theta` only needs to be specified if `under_relaxation` is DBD.

under_relaxation_kappa—real value defining the increment for the learning rate (under-relaxation term) of the delta-bar-delta algorithm. The value of under_relaxation_kappa is between zero and one. If the change in the variable (head) is of the same sign to that of the previous iteration, the under-relaxation term is increased by an increment of under_relaxation_kappa. The value usually ranges from 0.03 to 0.3; a value of 0.1 works well for most problems. under_relaxation_kappa only needs to be specified if under_relaxation is DBD.

under_relaxation_gamma—real value defining the history or memory term factor of the delta-bar-delta algorithm. under_relaxation_gamma is between zero and 1 but cannot be equal to one. When under_relaxation_gamma is zero, only the most recent history (previous iteration value) is maintained. As under_relaxation_gamma is increased, past history of iteration changes has greater influence on the memory term. The memory term is maintained as an exponential average of past changes. Retaining some past history can overcome granular behavior in the calculated function surface and therefore helps to overcome cyclic patterns of non-convergence. The value usually ranges from 0.1 to 0.3; a value of 0.2 works well for most problems. under_relaxation_gamma only needs to be specified if under_relaxation is not NONE.

under_relaxation_momentum—real value defining the fraction of past history changes that is added as a momentum term to the step change for a nonlinear iteration. The value of under_relaxation_momentum is between zero and one. A large momentum term should only be used when small learning rates are expected. Small amounts of the momentum term help convergence. The value usually ranges from 0.0001 to 0.1; a value of 0.001 works well for most problems. under_relaxation_momentum only needs to be specified if under_relaxation is DBD.

backtracking_number—integer value defining the maximum number of backtracking iterations allowed for residual reduction computations. If backtracking_number = 0 then the backtracking iterations are omitted. The value usually ranges from 2 to 20; a value of 10 works well for most problems.

backtracking_tolerance—real value defining the tolerance for residual change that is allowed for residual reduction computations. backtracking_tolerance should not be less than one to avoid getting stuck in local minima. A large value serves to check for extreme residual increases, while a low value serves to control step size more severely. The value usually ranges from 1.0 to 10^6; a value of 10^4 works well for most problems but lower values like 1.1 may be required for harder problems. backtracking_tolerance only needs to be specified if backtracking_number is greater than zero.

backtracking_reduction_factor—real value defining the reduction in step size used for residual reduction computations. The value of backtracking_reduction_factor is between zero and one. The value usually ranges from 0.1 to 0.3; a value of 0.2 works well for most problems. backtracking_reduction_factor only needs to be specified if backtracking_number is greater than zero.

backtracking_residual_limit—real value defining the limit to which the residual is reduced with backtracking. If the residual is smaller than backtracking_residual_limit, then further backtracking is not performed. A value of 100 is suitable for large problems and residual reduction to smaller values may only slow down computations. backtracking_residual_limit only needs to be specified if backtracking_number is greater than zero.

inner_maximum—integer value defining the maximum number of inner (linear) iterations. The number typically depends on the characteristics of the matrix solution scheme being used. For nonlinear

problems, `inner_maximum` usually ranges from 60 to 600; a value of 100 will be sufficient for most linear problems.

`inner_hclose`—real value defining the head change criterion for convergence of the inner (linear) iterations, in units of length. When the maximum absolute value of the head change at all nodes during an iteration is less than or equal to `inner_hclose`, the matrix solver assumes convergence. Commonly, `inner_hclose` is set an order of magnitude less than the `outer_hclose` value specified for the `NONLINEAR` block.

`inner_rclose`—real value that defines the flow residual tolerance for convergence of the IMS linear solver and specific flow residual criteria used. This value represents the maximum allowable residual at any single node. Value is in units of length cubed per time, and must be consistent with MODFLOW 6 length and time units. Usually a value of 1.0×10^{-1} is sufficient for the flow-residual criteria when meters and seconds are the defined MODFLOW 6 length and time.

`rclose_option`—an optional keyword that defines the specific flow residual criterion used. `L2NORM_RCLOSE`–an optional keyword that is used to specify that `inner_rclose` represents a L-2 Norm closure criteria instead of a infinity-Norm (absolute convergence criteria). When `L2NORM_RCLOSE` is specified, a reasonable initial `inner_rclose` value is $(1.0 \times 10^{-1} \times \text{active nodes})$ when meters and seconds are the defined MODFLOW 6 length and time. `RELATIVE_RCLOSE`–an optional keyword that is used to specify that `inner_rclose` represents a relative L-2 Norm reduction closure criteria instead of a infinity-Norm (absolute convergence criteria). When `RELATIVE_RCLOSE` is specified, a reasonable initial `inner_rclose` value is 1.0×10^{-4} and convergence is achieved for a given inner (linear) iteration when $\Delta h \leq$ `inner_hclose` and the current L-2 Norm is $\leq$ the product of the `RELATIVE_RCLOSE` and the initial L-2 Norm for the current inner (linear) iteration. If `rclose_option` is not specified, an absolute residual (infinity-norm) criterion is used.

`linear_acceleration`—a keyword that defines the linear acceleration method used by the default IMS linear solvers. `CG` - preconditioned conjugate gradient method. `BICGSTAB` - preconditioned bi-conjugate gradient stabilized method.

`relaxation_factor`—optional real value that defines the relaxation factor used by the incomplete LU factorization preconditioners (MILU(0) and MILUT). `relaxation_factor` is unitless and should be greater than or equal to 0.0 and less than or equal to 1.0. `relaxation_factor` values of about 1.0 are commonly used, and experience suggests that convergence can be optimized in some cases with relax values of 0.97. A `relaxation_factor` value of 0.0 will result in either ILU(0) or ILUT preconditioning (depending on the value specified for `preconditioner_levels` and/or `preconditioner_drop_tolerance`). By default, `relaxation_factor` is zero.

`preconditioner_levels`—optional integer value defining the level of fill for ILU decomposition used in the ILUT and MILUT preconditioners. Higher levels of fill provide more robustness but also require more memory. For optimal performance, it is suggested that a large level of fill be applied (7 or 8) with use of a drop tolerance. Specification of a `preconditioner_levels` value greater than zero results in use of the ILUT preconditioner. By default, `preconditioner_levels` is zero and the zero-fill incomplete LU factorization preconditioners (ILU(0) and MILU(0)) are used.

`preconditioner_drop_tolerance`—optional real value that defines the drop tolerance used to drop preconditioner terms based on the magnitude of matrix entries in the ILUT and MILUT preconditioners. A value of 10^{-4} works well for most problems. By default, `preconditioner_drop_tolerance` is zero and the zero-fill incomplete LU factorization preconditioners (ILU(0) and MILU(0)) are used.

number_orthogonalizations—optional integer value defining the interval used to explicitly recalculate the residual of the flow equation using the solver coefficient matrix, the latest head estimates, and the right hand side. For problems that benefit from explicit recalculation of the residual, a number between 4 and 10 is appropriate. By default, **number_orthogonalizations** is zero.

scaling_method—an optional keyword that defines the matrix scaling approach used. By default, matrix scaling is not applied. **NONE** - no matrix scaling applied. **DIAGONAL** - symmetric matrix scaling using the POLCG preconditioner scaling method in Hill (1992). **L2NORM** - symmetric matrix scaling using the L2 norm.

reordering_method—an optional keyword that defines the matrix reordering approach used. By default, matrix reordering is not applied. **NONE** - original ordering. **RCM** - reverse Cuthill McKee ordering. **MD** - minimum degree ordering.

IMS Default and Specified Complexity Values

The values that are assigned to the **nonlinear** and **linear** variables for the the **simple**, **moderate**, and **complex** complexity options are summarized in table 19. The values defined for the **simple** complexity option are assigned if the **COMPLEXITY** keyword is not specified in the **OPTIONS** block.

Table 19: IMS variable values for the available complexity options.

Nonlinear Variable	default/simple	moderate	complex
OUTER_HCLOSE	0.001	0.01	0.1
OUTER_MAXIMUM	25	50	100
UNDER_RELAXATION	NONE	DBD	DBD
UNDER_RELAXATION_THETA	0.0	0.9	0.8
UNDER_RELAXATION_KAPPA	0.0	0.0001	0.0001
UNDER_RELAXATION_GAMMA	0.0	0.0	0.0
UNDER_RELAXATION_MOMENTUM	0.0	0.0	0.0
BACKTRACKING_NUMBER	0	0	20
BACKTRACKING_TOLERANCE	0.0	0.0	1.05
BACKTRACKING_REDUCTION_FACTOR	0.0	0.0	0.1
BACKTRACKING_RESIDUAL_LIMIT	0.0	0.0	0.002

Linear Variable	default/simple	moderate	complex
INNER_MAXIMUM	50	100	500
INNER_HCLOSE	0.001	0.01	0.1
INNER_RCLOSE	0.1	0.1	0.1
RCLOSE_OPTION	infinity-norm	infinity-norm	infinity-norm
LINEAR_ACCELERATION	CG	BICGSTAB	BICGSTAB
RELAXATION_FACTOR	0.0	0.97	0.0
PRECONDITIONER_LEVELS	0	0	5
PRECONDITIONER_DROP_TOLERANCE	0.0	0.0	0.0001
NUMBER_ORTHOGONALIZATIONS	0	0	2
SCALING_METHOD	NONE	NONE	NONE
REORDERING_METHOD	NONE	NONE	NONE

Example Input File

```
BEGIN OPTIONS
  PRINT_OPTION 1
  COMPLEXITY MODERATE
END OPTIONS

BEGIN NONLINEAR
  OUTER_HCLOSE 1.E-4
  OUTER_MAXIMUM 2000
  UNDER_RELAXATION DBD
  UNDER_RELAXATION_THETA 0.70
  UNDER_RELAXATION_KAPPA 0.100000E-03
  UNDER_RELAXATION_GAMMA 0.
  UNDER_RELAXATION_MOMENTUM 0.
  BACKTRACKING_NUMBER 20
  BACKTRACKING_TOLERANCE 2.
  BACKTRACKING_REDUCTION_FACTOR 0.6
  BACKTRACKING_RESIDUAL_LIMIT 5.000000E-04
END NONLINEAR

BEGIN LINEAR
  INNER_MAXIMUM 100
  INNER_HCLOSE 1.0E-4
  INNER_RCLOSE 0.001
  LINEAR_ACCELERATION BIGSTAB
  RELAXATION_FACTOR 0.97
  SCALING_METHOD NONE
  REORDERING_METHOD NONE
END LINEAR
```

Observation (OBS) Utility

For consistency with earlier versions of MODFLOW (specifically, MODFLOW-2000 and MODFLOW-2005), MODFLOW 6 supports an "Observation" utility. Unlike the earlier versions of MODFLOW, the Observation utility of MODFLOW 6 does not require input of "observed" values, which typically were field- or lab-measured values. The Observation utility described here provides options for extracting numeric values of interest generated in the course of a model run. The Observation utility does not calculate residual values (differences between observed and model-calculated values). Output generated by the Observation utility is designed to facilitate further processing. For convenience and for consistency with earlier terminology, individual entries of the Observation utility are referred to as "observations."

Input for the Observation utility is read from one or more input files, where each file is associated with a specific model or package. For extracting values simulated by a GWF model, input is read from a file that is specified as type "OBS6" in the Name File. For extracting model values associated with a package, input is read from a file designated by the keyword "OBS6" in the Options block of the package of interest. The structures of observation input files for models and packages do not differ. Where a file name (or path name) containing spaces is to be read, enclose the name in single quotation marks.

Each OBS6 file can contain an OPTIONS block and one or more CONTINUOUS blocks. Each OBS6 file must contain at least one block. If present, the OPTIONS block must appear first. The CONTINUOUS blocks can be listed in any order. Comments, indicated by the presence of the "#" character in column 1, can appear anywhere in the file and are ignored.

Observations are output at the end of each time step and represent the value used by MODFLOW 6 during the time step. When input to the OBS utility references a stress-package boundary (for packages other than the advanced stress packages) that is not defined for a stress period of interest, a special NODATA value, indicating that a simulated value is not available, is written to output. The NODATA value is 3.0×10^{30}.

Output files to be generated by the Observation utility can be either text or binary. When a text file is used for output, the user can specify the number of digits of precision are to be used in writing values. For compatibility with common spreadsheet programs, text files are written in Comma-Separated Values (CSV) format. For this reason, text output files are commonly named with "csv" as the extension. By convention, binary output files are named with "bsv" (for "binary simulated values") as the extension.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [PRECISION <precision>]
  [DIGITS <digits>]
  [PRINT_INPUT]
END OPTIONS
```

```
BEGIN CONTINUOUS FILEOUT <obs_output_file_name> [BINARY]
  <obsname> <obstype> <id> [<id2>]
  <obsname> <obstype> <id> [<id2>]
  ...
END CONTINUOUS
```

Explanation of Variables

`precision`—Keyword and precision specifier for output of binary data, which can be either SINGLE or DOUBLE. The default is DOUBLE. When simulated values are written to a file specified as file type DATA(BINARY) in the Name File, the precision specifier controls whether the data (including simulated values and, for continuous observations, time values) are written as single- or double-precision.

`digits`—Keyword and an integer digits specifier used for conversion of simulated values to text on output. The default is 5 digits. When simulated values are written to a file specified as file type DATA in the Name File, the digits specifier controls the number of significant digits with which simulated values are written to the output file. The digits specifier has no effect on the number of significant digits with which the simulation time is written for continuous observations.

`PRINT_INPUT`—keyword to indicate that the list of observation information will be written to the listing file immediately after it is read.

`FILEOUT`—keyword to specify that an output filename is expected next.

`obs_output_file_name`—Name of a file to which simulated values corresponding to observations in the block are to be written. The file name can be an absolute or relative path name. A unique output file must be specified for each SINGLE or CONTINUOUS block. If the "BINARY" option is used, output is written in binary form. By convention, text output files have the extension "csv" (for "Comma-Separated Values") and binary output files have the extension "bsv" (for "Binary Simulated Values").

`BINARY`—an optional keyword used to indicate that the output file should be written in binary (unformatted) form.

`obsname`—string of 1 to 40 nonblank characters used to identify the observation. The identifier need not be unique; however, identification and post-processing of observations in the output files are facilitated if each observation is given a unique name.

`obstype`—a string of characters used to identify the observation type.

`id`—Text identifying cell where observation is located. For packages other than NPF, if boundary names are defined in the corresponding package input file, ID can be a boundary name. Otherwise ID is a cellid. If the model discretization is type DIS, cellid is three integers (layer, row, column). If the discretization is DISV, cellid is two integers (layer, cell number). If the discretization is DISU, cellid is one integer (node number).

`id2`—Text identifying cell adjacent to cell identified by ID. The form of ID2 is as described for ID. ID2 is used for intercell-flow observations of a GWF model, for three observation types of the LAK Package, for two observation types of the MAW Package, and one observation type of the UZF Package.

Available Observation Types

Observations are available for GWF models, GWF-GWF exchanges, and all stress packages. Available observation types have been listed for each package that supports observations (tables 6 to 18). All available observation types are repeated in Table 20 for convenience.

The sign convention adopted for flow observations are identical to the conventions used in budgets contained in listing files and used in the cell-by-cell budget output. For flow-ja-face observation types, negative and positive values represent a loss from and gain to the cellid specified for ID, respectively. For standard stress packages (Package = CHD, DRN, EVT, GHB, RCH, RIV, and WEL), negative and positive values represent a loss from and gain to the GWF model, respectively. For advanced packages (Package = LAK, MAW, SFR, and UZF), negative and positive values for exchanges with the GWF model (Observation type = lak, maw, sfr, uzf-gwrch, uzf-gwd, uzf-gwd-to-mvr, and uzf-gwet) represent a loss from and gain to the GWF model, respectively. For other advanced stress package flow terms, negative and positive values represent a loss from and gain from the advanced package, respectively.

Table 20: Available observation types

Model	Observation types	ID	ID2	Description
GWF	head	cellid	–	Head at a specified cell.
GWF	drawdown	cellid	–	Drawdown at a specified cell calculated as difference between starting head and simulated head for the time step.
GWF	flow-ja-face	cellid	cellid	Flow between two adjacent cells.

Exchange	Observation type	ID	ID2	Description
GWF-GWF	flow-ja-face	cellid	cellid	Flow rate for specified exchange.

Stress Package	Observation type	ID	ID2	Description
CHD	chd	cellid or boundname	–	Flow between the groundwater system and a constant-head boundary or a group of cells with constant-head boundaries.
DRN	drn	cellid or boundname	–	Flow between the groundwater system and a drain boundary or group of drain boundaries.
DRN	to-mvr	cellid or boundname	–	Drain boundary discharge that is available for the MVR package for a drain boundary or a group of drain boundaries.
EVT	evt	cellid or boundname	–	Flow from the groundwater system through an evapotranspiration boundary or group of evapotranspiration boundaries.
GHB	ghb	cellid or boundname	–	Flow between the groundwater system and a general-head boundary or group of general-head boundaries.

Table 20. Available observation types.—Continued

Model	Observation types	ID	ID2	Description
GHB	to-mvr	cellid or boundname	–	General-head boundary discharge that is available for the MVR package from a general-head boundary or group of general-head boundaries.
RCH	rch	cellid or boundname	–	Flow to the groundwater system through a recharge boundary or a group of recharge boundaries.
RIV	riv	cellid or boundname	–	Flow between the groundwater system and a river boundary.
RIV	to-mvr	cellid or boundname	–	River boundary discharge that is available for the MVR package.
WEL	wel	cellid or boundname	–	Flow between the groundwater system and a well boundary or a group of well boundaries.
WEL	to-mvr	cellid or boundname	–	Well boundary discharge that is available for the MVR package for a well boundary or a group of well boundaries.
LAK	stage	lakeno or boundname	–	Surface-water stage in a lake. If boundname is specified, boundname must be unique for each lake.
LAK	ext-inflow	lakeno or boundname	–	Specified inflow into a lake or group of lakes.
LAK	outlet-inflow	lakeno or boundname	–	Simulated inflow from upstream lake outlets into a lake or group of lakes.
LAK	inflow	lakeno or boundname	–	Sum of specified inflow and simulated inflow from upstream lake outlets into a lake or group of lakes.
LAK	from-mvr	lakeno or boundname	–	Inflow into a lake or group of lakes from the MVR package.
LAK	rainfall	lakeno or boundname	–	Rainfall rate applied to a lake or group of lakes.
LAK	runoff	lakeno or boundname	–	Runoff rate applied to a lake or group of lakes.
LAK	lak	lakeno or boundname	`iconn` or –	Simulated flow rate for a lake or group of lakes and its aquifer connection(s). If boundname is not specified for ID, then the simulated lake-aquifer flow rate at a specific lake connection is observed. In this case, ID2 must be specified and is the connection number `iconn`.
LAK	withdrawal	lakeno or boundname	–	Specified withdrawal rate from a lake or group of lakes.
LAK	evaporation	lakeno or boundname	–	Simulated evaporation rate from a lake or group of lakes.

Table 20. Available observation types.—Continued

Model	Observation types	ID	ID2	Description
LAK	ext-outflow	outletno or boundname	–	External outflow from a lake outlet, a lake, or a group of lakes to an external boundary. If boundname is not specified for ID, then the external outflow from a specific lake outlet is observed. In this case, ID is the outlet number outletno.
LAK	to-mvr	outletno or boundname	–	Outflow from a lake outlet, a lake, or a group of lakes that is available for the MVR package. If boundname is not specified for ID, then the outflow available for the MVR package from a specific lake outlet is observed. In this case, ID is the outlet number outletno.
LAK	storage	lakeno or boundname	–	Simulated storage flow rate for a lake or group of lakes.
LAK	constant	lakeno or boundname	–	Simulated constant-flow rate for a lake or group of lakes.
LAK	outlet	outletno or boundname	–	Simulate outlet flow rate from a lake outlet, a lake, or a group of lakes. If boundname is not specified for ID, then the flow from a specific lake outlet is observed. In this case, ID is the outlet number outletno.
LAK	volume	lakeno or boundname	–	Simulated lake volume or group of lakes.
LAK	surface-area	lakeno or boundname	–	Simulated surface area for a lake or group of lakes.
LAK	wetted-area	lakeno or boundname	iconn or –	Simulated wetted-area for a lake or group of lakes and its aquifer connection(s). If boundname is not specified for ID, then the wetted area of a specific lake connection is observed. In this case, ID2 must be specified and is the connection number iconn.
LAK	conductance	lakeno or boundname	iconn or –	Calculated conductance for a lake or group of lakes and its aquifer connection(s). If boundname is not specified for ID, then the calculated conductance of a specific lake connection is observed. In this case, ID2 must be specified and is the connection number iconn.
MAW	head	wellno or boundname	–	Head in a multi-aquifer well. If boundname is specified, boundname must be unique for each multi-aquifer well.

Table 20.   Available observation types.—Continued

Model	Observation types	ID	ID2	Description
MAW	from-mvr	wellno or boundname	–	Simulated inflow to a well from the MVR package for a multi-aquifer well or a group of multi-aquifer wells.
MAW	maw	wellno or boundname	icon or –	Simulated flow rate for a multi-aquifer well or a group of multi-aquifer wells and its aquifer connection(s). If boundname is not specified for ID, then the simulated multi-aquifer well-aquifer flow rate at a specific multi-aquifer well connection is observed. In this case, ID2 must be specified and is the connection number icon.
MAW	rate	wellno or boundname	–	Simulated pumping rate for a multi-aquifer well or a group of multi-aquifer wells.
MAW	rate-to-mvr	wellno or boundname	–	Simulated well discharge that is available for the MVR package for a multi-aquifer well or a group of multi-aquifer wells.
MAW	fw-rate	wellno or boundname	–	Simulated flowing well flow rate for a multi-aquifer well or a group of multi-aquifer wells.
MAW	fw-to-mvr	wellno or boundname	–	Simulated flowing well discharge rate that is available for the MVR package for a multi-aquifer well or a group of multi-aquifer wells.
MAW	storage	wellno or boundname	–	Simulated storage flow rate for a multi-aquifer well or a group of multi-aquifer wells.
MAW	constant	wellno or boundname	–	Simulated constant-flow rate for a multi-aquifer well or a group of multi-aquifer wells.
MAW	conductance	wellno or boundname	icon or –	Simulated well conductance for a multi-aquifer well or a group of multi-aquifer wells and its aquifer connection(s). If boundname is not specified for ID, then the simulated multi-aquifer well conductance at a specific multi-aquifer well connection is observed. In this case, ID2 must be specified and is the connection number icon.
MAW	fw-conductance	wellno or boundname	–	Simulated flowing well conductance for a multi-aquifer well or a group of multi-aquifer wells.

Table 20. Available observation types.—Continued

Model	Observation types	ID	ID2	Description
SFR	stage	rno or boundname	–	Surface-water stage in a stream-reach boundary. If boundname is specified, boundname must be unique for each reach.
SFR	ext-inflow	rno or boundname	–	Inflow into a stream-reach from an external boundary for a stream-reach or a group of stream-reaches.
SFR	inflow	rno or boundname	–	Inflow into a stream-reach from upstream reaches for a stream-reach or a group of stream-reaches.
SFR	from-mvr	rno or boundname	–	Inflow into a stream-reach from the MVR package for a stream-reach or a group of stream-reaches.
SFR	rainfall	rno or boundname	–	Rainfall rate applied to a stream-reach or a group of stream-reaches.
SFR	runoff	rno or boundname	–	Runoff rate applied to a stream-reach or a group of stream-reaches.
SFR	sfr	rno or boundname	–	Simulated flow rate for a stream-reach and its aquifer connection for a stream-reach or a group of stream-reaches.
SFR	evaporation	rno or boundname	–	Simulated evaporation rate from a stream-reach or a group of stream-reaches.
SFR	outflow	rno or boundname	–	Outflow from a stream-reach to downstream reaches for a stream-reach or a group of stream-reaches.
SFR	ext-outflow	rno or boundname	–	Outflow from a stream-reach to an external boundary for a stream-reach or a group of stream-reaches.
SFR	to-mvr	rno or boundname	–	Outflow from a stream-reach that is available for the MVR package for a stream-reach or a group of stream-reaches.
SFR	upstream-flow	rno or boundname	–	Upstream flow for a stream-reach or a group of stream-reaches from upstream reaches and the MVR package.
SFR	downstream-flow	rno or boundname	–	Downstream flow for a stream-reach or a group of stream-reaches prior to diversions and the MVR package.
UZF	uzf-gwrch	uzfno or boundname	–	Simulated recharge to the aquifer calculated by the UZF package for a UZF cell or a group of UZF cells.
UZF	uzf-gwd	uzfno or boundname	–	Simulated groundwater discharge to the land surface calculated by the UZF package for a UZF cell or a group of UZF cells.

Table 20.   Available observation types.—Continued

Model	Observation types	ID	ID2	Description
UZF	uzf-gwd-to-mvr	uzfno or boundname	–	Simulated groundwater discharge to the land surface calculated by the UZF package that is available to the MVR package for a UZF cell or a group of UZF cells.
UZF	uzf-gwet	uzfno or boundname	–	Simulated groundwater evapotranspiration calculated by the UZF package for a UZF cell or a group of UZF cells.
UZF	infiltration	uzfno or boundname	–	Specified infiltration rate applied to a UZF package for a UZF cell or a group of UZF cells with landflag values not equal to zero.
UZF	from-mvr	uzfno or boundname	–	Inflow into a UZF cell from the MVR package for a UZF cell or a group of UZF cells.
UZF	rej-inf	uzfno or boundname	–	Simulated rejected infiltration calculated by the UZF package for a UZF cell or a group of UZF cells.
UZF	rej-inf-to-mvr	uzfno or boundname	–	Simulated rejected infiltration calculated by the UZF package that is available to the MVR package for a UZF cell or a group of UZF cells.
UZF	uzet	uzfno or boundname	–	Simulated unsaturated evapotranspiration calculated by the UZF package for a UZF cell or a group of UZF cells.
UZF	storage	uzfno or boundname	–	Simulated storage flow rate for a UZF package cell or a group of UZF cells.
UZF	net-infiltration	uzfno or boundname	–	Simulated net infiltration rate for a UZF package cell or a group of UZF cells.
UZF	water-content	uzfno or boundname	depth	Unsaturated-zone water content at a user-specified depth (ID2) relative to the top of GWF cellid for a UZF cell. The user-specified depth must be greater than or equal to zero and less than the thickness of GWF cellid (TOP - BOT). If boundname is specified, boundname must be unique for each UZF cell.

Time-Variable Input

In earlier versions of MODFLOW, most stress-boundary packages read input on a stress period-by-stress period basis, and those values were held constant during the stress period. In MODFLOW 6, many stress values can be specified with a higher degree of time resolution (from time step to time step or from subtime step to subtime step) by using one of two time-variable approaches. Boundaries for which data are read as lists of cells can reference "time series" to implement the time variation. Boundaries for which data are read as 2-D arrays can reference "time-array series" to do so.

When MODFLOW 6 needs data from a time series or time-array series for a time interval representing a time step or subtime step, the series is queried to provide a time-averaged value or array of values for the requested time interval. For each series, the user specifies an interpolation method that determines how the value is assumed to behave between listed times. The interpolation method thus determines how the time averaging is performed. When a time-array series is used, interpolation is performed on an element-by-element basis to generate a 2-D array of interpolated values as needed.

The supported interpolation methods are STEPWISE, LINEAR, and LINEAREND. When the STEPWISE interpolation method is used, the value is assumed to remain constant at the value specified in one time-series record until the time listed in the subsequent record, when the value changes abruptly to the new value. In the LINEAR interpolation method, the value is assumed to change linearly between times listed in sequential records. LINEAREND is like LINEAR, except that instead of using the average value over a time step, the value at the end of a time step is used. Following sections document the structure of time-series and time-array-series files and their use.

Time Series

Any package that reads data as a list of cells and associated time-dependent input values can obtain those values from time series. For example, flow rates for a well or stage for a river boundary can be extracted from time series. During a simulation, values used for time-varying stresses (or auxiliary values) are based on the values provided in the time series and are updated each time step (or each subtime step, as appropriate). Input to define and use time series is described in this section.

A time series consists of a chronologically ordered list of time-series records, where each record includes a discrete time and a corresponding value. The value can be used to provide any time-varying numeric input, including stresses and auxiliary variables. A time series can be referenced in input for one or multiple variables in a given package.

Time-Series Files

Each time-series file is associated with exactly one package, and the name of a time-series file associated with a package is listed in the OPTIONS block for the package, preceded by the keywords "TS6 FILEIN." Any number of time-series files can be associated with a given package; a TS6 entry is required for each time-series file. A time-series file can contain one or more time series. Time-series files are not listed in either the simulation Name File or the model Name File. A given time-series file cannot be associated with more than one package. By convention, the extension ".ts" is used in names of time-series files.

Each time-series file contains an ATTRIBUTES block followed by a TIMESERIES block containing a series of lines, where each line contains a time followed by values for one or more time series at the specified time. The ATTRIBUTES block is required to define the name for each time series and the interpolation method to be used when an operation requires interpolation between times listed in the time series.

The time-series name(s) and interpolation method(s) are specified in the ATTRIBUTES block. Scale factor(s) for multiplying values optionally can be provided in the ATTRIBUTES block. NAME, METHOD, METHODS, SFAC, and SFACS are keywords. For appearance when a time-series file includes multiple time series, NAMES can be used as a synonym for the NAME keyword.

The syntax of the ATTRIBUTES block for a time-series file containing a single time series is as follows:

```
BEGIN ATTRIBUTES
  NAME    time-series-name
  METHOD interpolation-method
  [ SFAC sfac ]
END ATTRIBUTES
```

When a time-series file contains multiple time series, the time-series names are listed in a NAME (or NAMES) entry, similar to the example above. If the time series are to have different interpolation methods, the METHODS keyword is used in place of the METHOD keyword, and an interpolation method corresponding to each name is listed. If the time series are to have different scale factors, the SFACS keyword is used in place of the SFAC keyword.

The syntax of the ATTRIBUTES block for a time-series file containing multiple time series is as follows:

```
BEGIN ATTRIBUTES
  NAMES   time-series-name-1 [ time-series-name-2 ... time-series-name-n ]
  METHODS interpolation-method-1 [ interpolation-method-2 ... ]
  [ SFACS sfac-1 [ sfac-2 ... sfac-n ] ]
END ATTRIBUTES
```

In a case where a time-series file contains multiple time series and a single interpolation method applies to all time series in the file, the METHOD keyword can be used, and a single interpolation method is read. Similary, if a single scale factor applies to all time series in the file, the SFAC keyword can be used, and a single scale factor is read.

The ATTRIBUTES block is followed by a TIMESERIES block of the form:

```
BEGIN TIMESERIES
  time-series record
  time-series record
  ...
  time-series record
END TIMESERIES
```

where each time-series record is of the form:

```
  tsr-time tsr-value-1 [ tsr-value-2 tsr-value-3 ... ]
```

In situations where an individual time series in a file containing multiple time series does not include values for all specified times, a "no-data" value (3.0E30) can be used as a placeholder. When the "no-data" value is read for a time series, that time series will not include a time-series record for the corresponding time.

Explanation of Variables

time-series-name—Name by which a package references a particular time series. The name must be unique among all time series used in a package.

interpolation-method—Interpolation method, which is either STEPWISE, LINEAR, or LINEAREND.

sfac—Scale factor, which will multiply all `tsr-value` values in the time series. SFAC and SFACS are optional attributes; if omitted, `sfac` = 1.0.

`tsr-time`—A numeric time relative to the start of the simulation, in the time unit used in the simulation. Times must be strictly increasing.

`tsr-value`—A numeric data value corresponding to tsr-time. The value 3.0E30 is treated as a "no-data" value and can be used wherever a time series in a file containing multiple time series does not have a value corresponding to the time specified by `tsr-time`.

Using Time Series in a Package

When one or more time series are to define numeric input for a package, the name(s) of time-series files need to be defined in an OPTIONS block at the top of the package input file. The keyword TS6 followed by the keyword FILEIN are used to identify the name of each time-series file. Each time-series file can contain one or more time series, and each OPTIONS block can contain zero or more TS6 entries. The syntax for a TS6 entry in an OPTIONS block is:

```
BEGIN OPTIONS
  TS6 FILEIN time-series-file-name
END OPTIONS
```

Explanation of Variables Read from a Package Input File:

TS6—Keyword to specify that record corresponds to a time-series file.

FILEIN—Keyword to specify that an input filename is expected next.

`time-series-file-name`—Name of a time-series file in which time series used in the package are defined.

Each time series has a name. To specify that time-dependent values for one or more stress periods is to be extracted from a time series, the time-series name is listed in the position where a numeric value normally would be provided.

Example use of time series to define package input

The following example illustrates the use of three time series in input for the Well Package in a model with a structured grid. For an unstructured grid, the layer, row, and column indices for each observation would be replaced by a node number.

Contents of file "well_pump_rates.ts":

```
BEGIN ATTRIBUTES
  NAMES well-A-series well-B-series well-C-series
  METHODS stepwise linear stepwise
END ATTRIBUTES

BEGIN TIMESERIES
  # time well-A-series    well-B-series    well-C-series
  0.0          0.0              0.0              0.0
  1.0       -500.0              0.0           -400.0
  2.0       -500.0          -1000.0           -500.0
  5.0       -500.0          -1200.0           -200.0
  8.0       -500.0          -1100.0              0.0
END TIMESERIES
```

Contents of the Well Package input file:

```
BEGIN OPTIONS
  TS6 FILEIN well_pump_rates.ts
END OPTIONS

BEGIN DIMENSIONS
  MAXBOUND 4
END DIMENSIONS

BEGIN PERIOD 2
  #layer  row  col  Q (or time series)
       9  192   44  well-A-series
      10   43   17  well-B-series
      11   12   17  well-C-series
END PERIOD

BEGIN PERIOD 4
  #layer  row  col  Q (or time series)
       9  192   44  well-A-series
      10   43   17  well-B-series
      11   12   17  well-C-series
       2   27   36  -900.0
END PERIOD

BEGIN PERIOD 8
       2   27   36  -900.0
END PERIOD
```

In the example above, the Well package would have zero wells active in stress period 1. Three wells whose discharge rates are controlled by time series well-A-series, well-B-series, and well-C-series would be active in stress periods 2 and 3. Stress periods 4 through 7 would include the three time-series-controlled wells plus a well with a constant discharge of 900 (L^3/T). In stress period 8, only the constant-discharge well would be active.

Time-Array Series

Any package that reads data for a structured model in the form of 2-D arrays can obtain those array data from a time-array series. For example, recharge rates or maximum evapotranspiration rates can be extracted from time-array series. During a simulation, values used for time-varying stresses (or auxiliary values) are based on the values provided in the time-array series and are updated each time step (or each subtime step, as appropriate). Input to define and use time-array series is described in this section.

A time-array series consists of a chronologically ordered list of arrays, where each array is associated with a discrete time. The array data can be used to provide any time-varying, array-based numeric input.

Time-Array-Series Files

Each time-array-series file is associated with exactly one package, and the name of a time-array-series file associated with a package is listed in the OPTIONS block for the package, preceded by the keywords "TAS6 FILEIN." Any number of time-array-series files can be associated with a given package; a TAS6 entry is required for each time-array-series file. Time-array-series files are not listed in either the simulation Name File or the model Name File. A given time-array-series file cannot be associated with more than one package.

One time-array-series file defines a single time-array series. A time-array-series file contains an ATTRIBUTES block followed by a series of TIME blocks, where each TIME block contains data to define an array corresponding to a discrete time. The U2DREL array reading utility is used to read the array (see the "Input Instructions for Array Reading Utility Subroutines" section). The ATTRIBUTES block is required to define the name for the time-array series and the interpolation method to be used when an operation requires interpolation between times listed in the time-array series. By convention, the extension ".tas" is used in names of time-array-series files.

The syntax of the ATTRIBUTES block for a time-array-series file is as follows:

```
BEGIN ATTRIBUTES
  NAME   time-array-series-name
  METHOD interpolation-method
  [ SFAC sfac ]
END ATTRIBUTES
```

The ATTRIBUTES block is followed by any number of TIME blocks of the form:

```
BEGIN TIME tas-time
  tas-array
END TIME
```

Explanation of Variables

time-array-series-name—Name by which a package references a particular time-array series. The name must be unique among all time-array series used in a package.

interpolation-method—Interpolation method, which is either STEPWISE or LINEAR.

sfac—Scale factor, which will multiply all array values in time series. SFAC is an optional attribute; if omitted, SFAC = 1.0.

tas-time—A numeric time relative to the start of the simulation, in the time unit used in the simulation. Times must be strictly increasing.

tas-array—A 2-D array of numeric, floating-point values, or a constant value, readable by the U2DREL array-reading utility.

Using Time-Array Series in a Package

When one or more time-array series are to define numeric input for a package, the name(s) of time-array-series file(s) need to be defined in an OPTIONS block at the top of the package input file. The keywords "TAS6 FILEIN" are used to identify the name of each time-array-series file. Each time-array-series file contains exactly one time-array series, and each OPTIONS block can contain zero or more TAS6 entries. The syntax for a TAS6 entry in an OPTIONS block is:

```
BEGIN OPTIONS
  TAS6 FILEIN time-array-series-file-name
END OPTIONS
```

A time-array series is linked to an array in one or more stress-period blocks used to define package input. To indicate that an array is to be controlled by a time-array series, the array property word is followed by the keyword TIMEARRAYSERIES and the time-array series name. When the TIMEAR-RAYSERIES keyword is found (and the array to be populated supports time-array series), the array

reader is not invoked. Consequently, the array-control record and any associated input are omitted. The syntax to define the link is:

```
BEGIN PERIOD kper
  property-name TIMEARRAYSERIES time-array-series-name
END PERIOD
```

Explanation of Variables Read from a Package Input File:

TAS6—Keyword to specify that record corresponds to a time-array-series file.

FILEIN—Keyword to specify that an input filename is expected next.

time-array-series-file-name—Name of a time-array-series file in which a time-array series used in the package is defined.

property-name—Name of property represented by array to be controlled by a time-array series.

time-array-series-name—Name of time-array series. The time-array series must be defined in one of the files listed in the OPTIONS block with the TAS6 FILEIN keywords.

Example use of time-array series to define package input

The following example illustrates the use of a time-array series to control the Recharge property of the Recharge package in a model with a structured grid. In this example time-array series values are obtained from the time-array series "RchArraySeries_1" defined in file "rch_time_array_series.tas." The RchMult array is an auxiliary-variable array that is identified by the AUXMULTNAME keyword to be a multiplier for the recharge array. Accordingly, the recharge array is defined each time step as the element-by-element product of values interpolated from the "RchArraySeries_1" time-array series and values from the auxiliary-variable RchMult array.

Contents of Recharge package input file:

```
BEGIN OPTIONS
  READASARRAYS
  AUX RchMult
  TAS6 FILEIN rch_time_array_series.tas
  AUXMULTNAME RchMult
  PRINT_INPUT
END OPTIONS

BEGIN PERIOD 1
  IRCH
    CONSTANT 1
  RECHARGE TIMEARRAYSERIES RchArraySeries_1
  RchMult
    INTERNAL FACTOR 1.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  1.0  1.0  0.5  0.5  0.0  0.0  0.0  0.0  0.0
  0.0  1.0  1.0  1.0  1.0  0.5  0.0  0.0  0.0  0.0
  0.0  1.0  1.0  1.0  1.0  1.0  0.5  0.0  0.0  0.0
  0.0  0.2  1.0  1.0  1.0  1.0  1.0  0.5  0.2  0.0
  0.0  0.0  0.5  1.0  1.0  1.0  1.0  0.5  0.0  0.0
  0.0  0.0  0.0  0.2  0.2  0.2  0.2  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
END PERIOD
```

Contents of file "rch_time_array_series.tas":

```
BEGIN ATTRIBUTES
  NAME RchArraySeries_1
  METHOD LINEAR
END ATTRIBUTES

BEGIN TIME 0.0
  CONSTANT 0.0033
END TIME

BEGIN TIME 91.0
  CONSTANT 0.0035
END TIME

BEGIN TIME 183.0
  CONSTANT 0.0037
END TIME

BEGIN TIME 274.0
  CONSTANT 0.0039
END TIME

BEGIN TIME 365.0
  CONSTANT 0.0035
END TIME
```

Description of Groundwater Flow (GWF) Model Binary Output Files

Users can optionally write MODFLOW 6 output to binary files. There are several different types of binary output files. The first type is new to MODFLOW and is called a binary grid file. The binary grid file contains all of the information necessary for a post-processing program to quickly reconstruct the the model grid and understand how cells are connected within the grid. The option to specify an IDOMAIN array for DIS and DISV grids may result in cells being connected across model layers. For this reason, cell connectivity information is written to the binary grid file. The second type of binary file is one that contains simulated results, such as head. Simulated flows are written to a third type of binary file, called a budget file. The budget file contains simulated flows between connected cells and flows from stress packages. Lastly, observations can also be written to binary output files.

All floating point variables are written to the binary output files as DOUBLE PRECISION Fortran variables. Integer variables are written to the output files as Fortran integer variables. Some variables are character strings and are indicated as so in the following descriptions.

The file formats for the binary files are described in the following sections. The frequency of output and the types of output files that are created is described in the Output Control Option and in the individual package input files.

Binary Grid File

MODFLOW 6   writes a binary grid file that can be used for post processing model results. The file name is assigned automatically by the program by adding ".grb" to the end of the discretization input file name. The structure of the binary grid file depends on the type of discretization package that is used. The following subsections summarize the binary grid file for the different grid types. The red text in is not written to the binary grid file.

DIS Grids

Header 1: 'GRID DIS' CHARACTER(LEN=50)
Header 2: 'VERSION 1' CHARACTER(LEN=50)
Header 3: 'NTXT 16' CHARACTER(LEN=50)
Header 4: 'LENTXT 100' CHARACTER(LEN=50)

Read NTXT strings of size LENTXT. Set the number of data records (NDAT) equal to number of lines that do not begin with #.
Definition 0: '#Comment ...' CHARACTER(LEN=LENTXT), comments not presently written
Definition 1: 'NCELLS INTEGER NDIM 0 # ncells' CHARACTER(LEN=LENTXT)
Definition 2: 'NLAY INTEGER NDIM 0 # nlay' CHARACTER(LEN=LENTXT)
Definition 3: 'NROW INTEGER NDIM 0 # nrow' CHARACTER(LEN=LENTXT)
Definition 4: 'NCOL INTEGER NDIM 0 # ncol' CHARACTER(LEN=LENTXT)
Definition 5: 'NJA INTEGER NDIM 0 # nja' CHARACTER(LEN=LENTXT)
Definition 6: 'XORIGIN DOUBLE NDIM 0 # xorigin' CHARACTER(LEN=LENTXT)
Definition 7: 'YORIGIN DOUBLE NDIM 0 # yorigin' CHARACTER(LEN=LENTXT)
Definition 8: 'ANGROT DOUBLE NDIM 0 # angrot' CHARACTER(LEN=LENTXT)
Definition 9: 'DELR DOUBLE NDIM 1 ncol' CHARACTER(LEN=LENTXT)
Definition 10: 'DELC DOUBLE NDIM 1 nrow' CHARACTER(LEN=LENTXT)
Definition 11: 'TOP DOUBLE NDIM 1 nrow*ncol' CHARACTER(LEN=LENTXT)
Definition 12: 'BOTM DOUBLE NDIM 1 ncells' CHARACTER(LEN=LENTXT)
Definition 13: 'IA INTEGER NDIM 1 ncells+1' CHARACTER(LEN=LENTXT)
Definition 14: 'JA INTEGER NDIM 1 nja' CHARACTER(LEN=LENTXT)
Definition 15: 'IDOMAIN INTEGER NDIM 1 ncells' CHARACTER(LEN=LENTXT)
Definition 16: 'ICELLTYPE INTEGER NDIM 1 ncells' CHARACTER(LEN=LENTXT)

Read NDAT data variables using the definitions defined above.
Record 1: NCELLS INTEGER
Record 2: NLAY INTEGER
Record 3: NROW INTEGER
Record 4: NCOL INTEGER
Record 5: NJA INTEGER
Record 6: XORIGIN DOUBLE
Record 7: YORIGIN DOUBLE
Record 8: ANGROT DOUBLE
Record 9: DELR DOUBLE PRECISION ARRAY SIZE(NCOL)
Record 10: DELC DOUBLE PRECISION ARRAY SIZE (NROW)
Record 11: (TOP(J),J=1,NROW*NCOL) DOUBLE PRECISION ARRAY SIZE(NROW*NCOL)
Record 12: (BOTM(J),J=1,NCELLS) DOUBLE PRECISION ARRAY SIZE(NCELLS)
Record 13: (IA(J),J=1,NCELLS+1) INTEGER ARRAY SIZE(NCELLS+1)
Record 14: (JA(J),J=1,NJA) INTEGER ARRAY SIZE(NJA)
Record 15: (IDOMAIN(J),J=1,NCELLS) INTEGER ARRAY SIZE(NCELLS)
Record 16: (ICELLTYPE(J),J=1,NCELLS) INTEGER ARRAY SIZE(NCELLS)

DISV Grids

The binary grid file for DISV grids contains information on the vertices and which vertices comprise a cell. The x, y coordinates for each vertex are stored in the VERTICES array. The list of vertices that comprise all of the cells is stored in the JAVERT array. The list of vertices for any cell can be found using the IAVERT array. The following pseudocode shows how to loop through every cell in the DISV grid and obtain the cell vertices. The list of vertices is "closed" for each cell in that the first listed vertex is equal to the last listed vertex.

```
DO K = 1, NLAY
  DO N = 1, NCPL
    PRINT *, 'THIS IS CELL (LAYER, ICELL2D): ', K, N
    NVCELL = IAVERT(N+1) - IAVERT(N)
    PRINT*, 'NUMBER OF VERTICES FOR CELL IS', NVCELL
    DO IPOS = IAVERT(N), IAVERT(N + 1) - 1
      IVERT = JAVERT(IPOS)
      X = VERTICES(1,IVERT)
      Y = VERTICES(2,IVERT)
      PRINT *,'  VERTEX PAIR: ', X, Y
    ENDDO
  ENDDO
ENDDO
```

The IA and JA arrays are also contained in the DISV binary grid file. These arrays describe the cell connectivity. Connections in the JA array correspond directly with the FLOW-JA-FACE record that is written to the budget file.

The content of the DISV binary grid file is as follows.

Header 1: 'GRID DISV' CHARACTER(LEN=50)
Header 2: 'VERSION 1' CHARACTER(LEN=50)
Header 3: 'NTXT 20' CHARACTER(LEN=50)
Header 4: 'LENTXT 100' CHARACTER(LEN=50)

Read NTXT strings of size LENTXT. Set the number of data records (NDAT) equal to number of lines that do not begin with #.
Definition 0: '#Comment ...' CHARACTER(LEN=LENTXT), comments not presently written
Definition 1: 'NCELLS INTEGER NDIM 0 # ncells' CHARACTER(LEN=LENTXT)
Definition 2: 'NLAY INTEGER NDIM 0 # nlay' CHARACTER(LEN=LENTXT)
Definition 3: 'NCPL INTEGER NDIM 0 # ncpl' CHARACTER(LEN=LENTXT)
Definition 4: 'NVERT INTEGER NDIM 0 # nvert' CHARACTER(LEN=LENTXT)
Definition 5: 'NJAVERT INTEGER NDIM 0 # njavert' CHARACTER(LEN=LENTXT)
Definition 6: 'NJA INTEGER NDIM 0 # nja' CHARACTER(LEN=LENTXT)
Definition 7: 'XORIGIN DOUBLE NDIM 0 # xorigin' CHARACTER(LEN=LENTXT)
Definition 8: 'YORIGIN DOUBLE NDIM 0 # yorigin' CHARACTER(LEN=LENTXT)
Definition 9: 'ANGROT DOUBLE NDIM 0 # angrot' CHARACTER(LEN=LENTXT)
Definition 10: 'TOP DOUBLE NDIM 1 ncpl' CHARACTER(LEN=LENTXT)
Definition 11: 'BOTM DOUBLE NDIM 1 ncells' CHARACTER(LEN=LENTXT)
Definition 12: 'VERTICES DOUBLE NDIM 2 2 nvert' CHARACTER(LEN=LENTXT)
Definition 13: 'CELLX DOUBLE NDIM 1 ncpl' CHARACTER(LEN=LENTXT)
Definition 14: 'CELLY DOUBLE NDIM 1 ncpl' CHARACTER(LEN=LENTXT)

Definition 15: 'IAVERT INTEGER NDIM 1 ncpl+1' CHARACTER(LEN=LENTXT)
Definition 16: 'JAVERT INTEGER NDIM 1 njavert' CHARACTER(LEN=LENTXT)
Definition 17: 'IA INTEGER NDIM 1 ncells+1' CHARACTER(LEN=LENTXT)
Definition 18: 'JA INTEGER NDIM 1 nja' CHARACTER(LEN=LENTXT)
Definition 19: 'IDOMAIN INTEGER NDIM 1 ncells' CHARACTER(LEN=LENTXT)
Definition 20: 'ICELLTYPE INTEGER NDIM 1 ncells' CHARACTER(LEN=LENTXT)

Read NDAT data variables using the definitions defined above.
Record 1: NCELLS INTEGER
Record 2: NLAY INTEGER
Record 3: NCPL INTEGER
Record 4: NVERT INTEGER
Record 5: NJAVERT INTEGER
Record 6: NJA INTEGER
Record 7: XORIGIN DOUBLE
Record 8: YORIGIN DOUBLE
Record 9: ANGROT DOUBLE
Record 10: (TOP(J),J=1,NCPL) DOUBLE PRECISION ARRAY SIZE(NCPL)
Record 11: ((BOTM(J),J=1,NCELLS) DOUBLE PRECISION ARRAY SIZE(NCELLS)
Record 12: ((VERTICES(J,K),J=1,2),K=1,NVERT) DOUBLE PRECISION ARRAY SIZE(2,NVERT)
Record 13: (CELLX(J),J=1,NCPL) DOUBLE PRECISION ARRAY SIZE(NCPL)
Record 14: (CELLY(J),J=1,NCPL) DOUBLE PRECISION ARRAY SIZE(NCPL)
Record 15: (IAVERT(J),J=1,NCPL+1) INTEGER ARRAY SIZE(NCPL+1)
Record 16: (JAVERT(J),J=1,NJAVERT) INTEGER ARRAY SIZE(NJAVERT)
Record 17: (IA(J),J=1,NCELLS+1) INTEGER ARRAY SIZE(NCELLS+1)
Record 18: (JA(J),J=1,NJA) INTEGER ARRAY SIZE(NJA)
Record 19: (IDOMAIN(J),J=1,NCELLS) INTEGER ARRAY SIZE(NCELLS)
Record 20: (ICELLTYPE(J),J=1,NCELLS) INTEGER ARRAY SIZE(NCELLS)

DISU Grids

The binary grid file for DISU grids may contain information on the vertices and which vertices comprise a cell, but this depends on whether or not the user provided the information in the DISU Package. This information is not required unless the XT3D option is used. If provided, the x, y coordinates for each vertex are stored in the VERTICES array. The list of vertices that comprise all of the cells is stored in the JAVERT array. The list of vertices for any cell can be found using the IAVERT array. Pseudocode for looping through cells in the grid is listed above in the section on the binary grid file for the DISV Package. As for the DISV binary grid file, the list of vertices is "closed" for each cell in that the first listed vertex is equal to the last listed vertex.

Header 1: 'GRID DISU' CHARACTER(LEN=50)
Header 2: 'VERSION 1' CHARACTER(LEN=50)
Header 3: 'NTXT 10' or 'NTXT 15' CHARACTER(LEN=50)
Header 4: 'LENTXT 100' CHARACTER(LEN=50)

Read `NTXT` strings of size `LENTXT`. Set the number of data records (`NDAT`) equal to number of lines that do not begin with #.
Definition 0: '#Comment ...' CHARACTER(LEN=LENTXT), comments not presently written
Definition 1: 'NODES INTEGER NDIM 0 # nodes' CHARACTER(LEN=LENTXT)
Definition 2: 'NJA INTEGER NDIM 0 # nja' CHARACTER(LEN=LENTXT)
Definition 3: 'XORIGIN DOUBLE NDIM 0 # xorigin' CHARACTER(LEN=LENTXT)
Definition 4: 'YORIGIN DOUBLE NDIM 0 # yorigin' CHARACTER(LEN=LENTXT)
Definition 5: 'ANGROT DOUBLE NDIM 0 # angrot' CHARACTER(LEN=LENTXT)
Definition 6: 'TOP DOUBLE NDIM 1 nodes' CHARACTER(LEN=LENTXT)
Definition 7: 'BOT DOUBLE NDIM 1 nodes' CHARACTER(LEN=LENTXT)
Definition 8: 'IA INTEGER NDIM 1 ncells+1' CHARACTER(LEN=LENTXT)
Definition 9: 'JA INTEGER NDIM 1 nja' CHARACTER(LEN=LENTXT)
Definition 10: 'ICELLTYPE INTEGER NDIM 1 ncells' CHARACTER(LEN=LENTXT)

If vertices are provided in the DISU Package, then 5 additional definitions are included:
Definition 11: 'VERTICES DOUBLE NDIM 2 2 nvert' CHARACTER(LEN=LENTXT)
Definition 12: 'CELLX DOUBLE NDIM 1 nodes' CHARACTER(LEN=LENTXT)
Definition 13: 'CELLY DOUBLE NDIM 1 nodes' CHARACTER(LEN=LENTXT)
Definition 14: 'IAVERT INTEGER NDIM 1 nodes+1' CHARACTER(LEN=LENTXT)
Definition 15: 'JAVERT INTEGER NDIM 1 njavert' CHARACTER(LEN=LENTXT)

Read `NDAT` data variables using the definitions defined above.
Record 1: NODES INTEGER
Record 2: NJA INTEGER
Record 3: XORIGIN DOUBLE
Record 4: YORIGIN DOUBLE
Record 5: ANGROT DOUBLE
Record 6: (TOP(J),J=1,NODES) DOUBLE PRECISION ARRAY SIZE(NODES)
Record 7: ((BOT(J),J=1,NODES) DOUBLE PRECISION ARRAY SIZE(NODES)

Record 8: `(IA(J),J=1,NODES+1)` INTEGER ARRAY SIZE(NODES+1)
Record 9: `(JA(J),J=1,NJA)` INTEGER ARRAY SIZE(NJA)
Record 10: `(ICELLTYPE(J),J=1,NCELLS)` INTEGER ARRAY SIZE(NCELLS)

If vertices are provided in the DISU Package, then 5 additional records are included:
Record 11: `((VERT(J,K),J=1,2),K=1,NVERT)` DOUBLE PRECISION ARRAY SIZE(2,NVERT)
Record 12: `(CELLX(J),J=1,NODES)` DOUBLE PRECISION ARRAY SIZE(NODES)
Record 13: `(CELLY(J),J=1,NODES)` DOUBLE PRECISION ARRAY SIZE(NODES)
Record 14: `(IAVERT(J),J=1,NODES+1)` INTEGER ARRAY SIZE(NODES+1)
Record 15: `(JAVERT(J),J=1,NJAVERT)` INTEGER ARRAY SIZE(NJAVERT)

Dependent Variable File

In the present MODFLOW 6 version, the `TEXT` value is specified as "HEAD". Cells that have been assigned an IDOMAIN value of zero or less are assigned a head value of 1.0×10^{30}. Cells that have converted to dry are assigned a dry value of -1.0×10^{30}. The large negative value allows the results from a previous simulation to be used as starting heads for a subsequent simulation. Cells assigned a large negative value as an initial condition will start the simulation as dry. Note that the dry value is not used if the Newton-Raphson Formulation is active. In this case, a dry cell will have a calculated head value that is below or at the bottom of the cell.

DIS Grids

For each stress period, time step, and layer for which data are saved to the binary output file, the following two records are written:

Record 1: `KSTP,KPER,PERTIM,TOTIM,TEXT,NCOL,NROW,ILAY`
Record 2: `((DATA(J,I,ILAY),J=1,NCOL),I=1,NROW)`

where

 `KSTP` is the time step number;
 `KPER` is the stress period number;
 `PERTIM` is the time value for the current stress period;
 `TOTIM` is the total simulation time;
 `TEXT` is a character string (character*16);
 `NCOL` is the number of columns;
 `NROW` is the number of rows;
 `ILAY` is the layer number; and
 `DATA` is the head data of size (NCOL,NROW,NLAY).

DISV Grids

For each stress period, time step, and layer for which data are saved to the binary output file, the following two records are written:

Record 1: `KSTP,KPER,PERTIM,TOTIM,TEXT,NCPL,1,ILAY`
Record 2: `(DATA(J,ILAY),J=1,NCPL)`

where

 `KSTP` is the time step number;
 `KPER` is the stress period number;
 `PERTIM` is the time value for the current stress period;
 `TOTIM` is the total simulation time;
 `TEXT` is a character string (character*16);
 `NCPL` is the number of cells per layer;
 `ILAY` is the layer number; and

`DATA` is the head data of size (NCPL,NLAY).

DISU Grids

For each stress period, time step, and layer for which data are saved to the binary output file, the following two records are written:

Record 1: KSTP,KPER,PERTIM,TOTIM,TEXT,NODES,1,1
Record 2: (DATA(N),N=1,NODES)

where

 KSTP is the time step number;
 KPER is the stress period number;
 PERTIM is the time value for the current stress period;
 TOTIM is the total simulation time;
 TEXT is a character string (character*16);
 NODES is the number cells in the model grid;
 DATA is unstructured head data of size (NODES).

LAK, MAW, and SFR Packages

For each stress period, time step, and layer for which data are saved to the binary output file, the following two records are written:

Record 1: `KSTP,KPER,PERTIM,TOTIM,TEXT,MAXBOUND,1,1`
Record 2: `(DATA(N),N=1,MAXBOUND)`

where

`KSTP` is the time step number;
`KPER` is the stress period number;
`PERTIM` is the time value for the current stress period;
`TOTIM` is the total simulation time;
`TEXT` is a character string (`character*16`);
`MAXBOUND` is the number advanced boundary items in the package;
`DATA` is unstructured dependent variable data of size (`MAXBOUND`).

Groundwater Flow Model Budget File

The budget file for the GWF Model contains intercell flows, flows due to changes in storage, flows from the stress packages and advanced stress packages, and exchange flows with another model. The intent of budget file is to contain all flow to and from any cell in the model. Users must activate saving of flow terms in the Output Control Package and in the individual packages.

The format for the budget file is different from the formats for previous MODFLOW versions. Specifically, intercell flows are written in a different manner using a compressed sparse row storage scheme. The record structure for the stress packages is also different and uses a method code 6, to distinguish it from the five method codes available in previous MODFLOW versions. The new code 6 indicates that additional text identifiers are present, that auxiliary variables may be present, and that two identifying integer numbers are contained in the list (one for the node number of the GWF Model cell, and the other for an identifier to where the flow is from).

Format of Budget File

The generalized form of the budget file is described so that utilities may be created to read the budget file. Additional information about the content and the form of the content for different grid types is described in subsequent sections.

Record 1: `KSTP,KPER,TEXT,NDIM1,NDIM2,-NDIM3`
Record 2: `IMETH,DELT,PERTIM,TOTIM`

 `IMETH=1:` *Read 1D array of size NDIM1*NDIM2*NDIM3.*
    Record 3: `(DATA(J),J=1,NDIM1*NDIM2*NDIM3)`

 `IMETH=6:` *Read text identifiers, auxiliary text labels, and list of information.*
    Record 3: `TXT1ID1`
    Record 4: `TXT2ID1`
    Record 5: `TXT1ID2`
    Record 6: `TXT2ID2`
    Record 7: `NDAT`
    Record 8: `(AUXTXT(N),N=1,NDAT-1)`
    Record 9: `NLIST`
    Record 10: `((ID1(N),ID2(N),(DATA2D(I,N),I=1,NDAT)),N=1,NLIST)`

where

 `KSTP` is the integer time step number;
 `KPER` is the integer stress period number;
 `TEXT` is a character string (character*16) indicating the flow type;
 `PERTIM` is the double precision time value for the current stress period;
 `TOTIM` is the double precision total simulation time;
 `NDIM1` is the integer size of first dimension;
 `NDIM2` is the integer size of second dimension;
 `NDIM3` is the integer size of third dimension;
 `IMETH` is an integer code that specifies the form of the remaining data;

`DELT` is the double precision length of the timestep;

`PERTIM` is the double precision time value for the current stress period;

`TOTIM` is the double precision total simulation time;

`DATA` is a double precision array of budget values;

`TXT1ID1` is a character string (character*16) containing the first text identifier for information in ID1;

`TXT2ID1` is a character string (character*16) containing the second text identifier for information in ID1;

`TXT1ID2` is a character string (character*16) containing the model name for information in ID2;

`TXT2ID2` is a character string (character*16) containing the package or model name for information in ID2;

`NDAT` is the number of columns in DATA2D, which is the number of auxiliary values plus 1;

`AUXTXT` is an array of size NDAT - 1 containing character*16 text names for each auxiliary variable;

`NLIST` is the size of the list;

`ID1` is the first identifying number;

`ID2` is the second identifying number, and

`DATA2D` is a double precision 2D array of size (NDAT,NLIST). The first column in DATA2D is the budget term; any remaining columns are auxiliary variable values.

Variations for Discretization Types

The format for the GWF Model budget file is the same no matter what discretization package is used; however, the variables may have different meanings depending on the grid type and the TEXT identifier. If the TEXT identifier in Record 1 is FLOW-JA-FACE and IMETH is 1, then the DATA array contains intercell flows and is of size NJA. If the TEXT identifier in Record 1 is something other than FLOW-JA-FACE (STO-SS or STO-SY, for example), then the dimension variables in Record 1 (NDIM1, NDIM2, and NDIM3) provide information about the size of the grid (table 21).

Table 21: Budget file variations that depend on discretization package type

Grid or Flow Type	NDIM1	NDIM2	NDIM3
DIS	NCOL	NROW	NLAY
DISV	NCPL	1	NLAY
DISU	NODES	1	1
FLOW-JA-FACE, IMETH=1	NJA	1	1

Budget File Contents

The type of information that is written to the budget file for a GWF Model depends on the packages used for the model and whether or not the save flags are set. Table 22 contains a list of the types of information that may be contained in a GWF Model budget file. In all cases, the flows in table 22 are flows to or a from a GWF Model cell. As described in the next section, intercell flows are written as FLOW-JA-FACE using IMETH=1. If the model has an active Storage Package, then STORAGE-SS and STORAGE-SY are written to the budget file using IMETH=1.

The remaining flow terms in table 22 are all written using IMETH=6. When IMETH=6 is used, the records contain additional text descriptors and two identifying numbers. For all records in the GWF Model budget file, TXT1ID1 is the name of the GWF Model and TXT2ID1 is also the name of the GWF Model. These text identifiers describe what is contained in ID1. For the GWF Model budget file, ID1 is the cell or node number in the GWF Model grid. The second set of text identifiers refer to the information in ID2. Unless noted otherwise in the description in table 22, TXT1ID2 is the name of the GWF Model, TXT2ID2 is the name of the package, and ID2 is the bound number in the package; for example, this is the first constant head cell, second constant head cell, and so forth.

Table 22: Types of information that may be contained in the GWF Model budget file

Flow Type (TEXT)	Method Code (IMETH)	Description
FLOW-JA-FACE	1	intercell flow; array of size(NJA)
STO-SS	1	confined storage; array of size (NCELLS)
STO-SY	1	unconfined storage; array of size (NCELLS)
CHD	6	constant head flow
WEL	6	well flow
WEL-TO-MVR	6	well flow that is routed to Mover Package
DRN	6	drain flow
DRN-TO-MVR	6	drain flow that is routed to Mover Package
RIV	6	river leakage
RIV-TO-MVR	6	river leakage that is routed to Mover Package
GHB	6	general-head boundary flow
GHB-TO-MVR	6	general-head boundary flow that is routed to Mover Package
RCH	6	recharge flow
EVT	6	evapotranspiration flow
MAW	6	multi-aquifer well flow; ID2 contains the well number
LAK	6	lake leakage; ID2 contains the lake number
SFR	6	stream leakage; ID2 contains the stream reach number
UZF-GWRCH	6	water table recharge from UZF Package
UZF-GWET	6	water table evapotranspiration from UZF Package
UZF-GWD	6	groundwater discharge to land surface from UZF Package
UZF-GWD-TO-MVR	6	groundwater discharge to land surface from UZF Package that is routed to Mover Package
FLOW-JA-FACE	6	flow to or from a cell in another GWF Model; TXT1ID1 is the name of the GWF Model described by this budget file, TXT2ID1 is the name of the GWF-GWF Exchange, TXT1ID2 is the name of the connected GWF Model, TXT2ID2 is the name of the GWF-GWF Exchange, and ID2 is the cell or node number of the cell in the connected model

Intercell Flows

MODFLOW 6 writes a special budget record for flow between connected cells. This record has a TEXT identifier equal to FLOW-JA-FACE. For this record, the total number of values is equal to NJA, which is the total number of connections. For each cell, the number of connections is equal to the number of connections to adjacent cells plus one, to represent the cell itself. Therefore, this budget record corresponds to the JA array. A value of zero is written to the node positions in the FLOW-JA-FACE record. The JA array that is written in the binary grid corresponds directly to the FLOW-JA-FACE record.

For regular MODFLOW grids, there are no longer records for FLOW RIGHT FACE, FLOW FRONT FACE, and FLOW LOWER FACE. Instead, intercell flows are written to the FLOW-JA-FACE record. Writing FLOW-JA-FACE allows face flows to be specified in straightforward manner, particularly when the IDOMAIN capability is used to remove cells and specify vertical pass-through cells.

The following pseudocode shows how to loop through and process intercell flows using the IA and JA arrays (which can be read from the binary grid file) and the FLOWJA array, which is written to the budget file. For a cell (N) that has been eliminated with IDOMAIN, the value for IA(N) and IA(N+1) will be equal, indicating that there are no connections or flows for that cell.

```
DO N = 1, NCELLS
  PRINT *, 'THIS IS CELL: ', N
  NCON = IA(N+1) - IA(N) - 1
  IF(NCON<0) NCON=0
  PRINT*, 'NUMBER OF CONNECTED CELLS IS ', NCON
  DO IPOS = IA(N) + 1, IA(N + 1) - 1
    M = JA(IPOS)
    Q = FLOWJA(IPOS)
    PRINT *,'  N M Q: ', N,M,Q
  ENDDO
ENDDO
```

LAK, MAW, SFR, and UZF Packages

For each stress period, time step, and data type that is saved to the LAK, MAW, SFR, and UZF Packages binary output files as `IMETH=6` budget file type. For all advanced packages, `NDIM1` is equal to the number of nodes, `NDIM2` is equal to 1, and `NDIM3` is equal to -1. The data that are written to the LAK, MAW, SFR, and UZF Package binary files are summarized in Tables 23 to 26, respectively.

Table 23: Data written to the LAK Package binary output file. Flow terms are listed in the order they are written to the LAK Package binary output file

Flow term	IMETH	NDAT / NLIST	Description
FLOW-JA-FACE	6	1 / 2*nlen	Connection flow from lake (ID1) to lake through a lake outlet to another lake (ID2). nlen is calculated as the sum of lake outlets that are connected to another lake (lakeout for a lake outlet is not equal to 0).
GWF	6	2 / maxbound	Calculated flow from lake (ID1) to GWF cell (ID2). The lake connection-aquifer flow area (FLOW-AREA) is saved as an auxiliary data item for this flow term.
EXT-INFLOW	6	1 / nlakes	Specified inflow to reach. The lake number is written to (ID1) and (ID2).
RUNOFF	6	1 / nlakes	Specified runoff to reach. The lake number is written to (ID1) and (ID2).
RAINFALL	6	1 / nlakes	Specified rainfall on reach. The lake number is written to (ID1) and (ID2).
EVAPORATION	6	1 / nlakes	Calculated evaporation from lake. The lake number is written to (ID1) and (ID2).
WITHDRAWAL	6	1 / nlakes	Specified withdrawal from lake. The lake number is written to (ID1) and (ID2).
STORAGE	6	2 / nlakes	Calculated flow from storage for lake. The lake number is written to (ID1) and (ID2). The lake volume (VOLUME) is saved as an auxiliary data item for this flow term.
CONSTANT	6	1 / nlakes	Calculated flow to maintain constant stage for lake. The lake number is written to (ID1) and (ID2).
EXT-OUTFLOW	6	1 / nlakes	Calculated outflow to external boundaries (is nonzero for lakes with outlets not connected to another lake). The lake number is written to (ID1) and (ID2).

Flow term	IMETH	NDAT / NLIST	Description
FROM-MVR	6	1 / nlakes	Calculated flow to lake from the MVR Package. Only saved if MVR Package is used in the LAK Package. The lake number is written to (ID1) and (ID2).
TO-MVR	6	1 / noutlets	Calculated flow from a lake outlet to the MVR Package. Only saved if MVR Package is used in the LAK Package. The lake outlet number is written to (ID1) and (ID2).
AUXILIARY	6	naux+1 / nlakes	Auxiliary variables, if specified in the LAK Package, are saved to this flow term. The first entry of the DATA2D column has a value of zero. The lake number is written to (ID1) and (ID2).

Table 24: Data written to the MAW Package binary output file. Flow terms are listed in the order they are written to the MAW Package binary output file

Flow term	IMETH	NDAT / NLIST	Description
GWF	6	2 / maxbound	Calculated flow from multi-aquifer well (ID1) to GWF cell (ID2). The multi-aquifer well-aquifer flow area (FLOW-AREA) is saved as an auxiliary data item for this flow term.
RATE	6	1 / nmawwells	Calculated pumping rate from the multi-aquifer well. The multi-aquifer well number is written to (ID1) and (ID2).
FW-RATE	6	1 / nmawwells	calculated flowing well discharge rate from the multi-aquifer well. Only saved if FLOWING_WELLS is specified in the OPTIONS block. The multi-aquifer well number is written to (ID1) and (ID2).
STORAGE	6	2 / nmawwells	Calculated flow from storage for multi-aquifer well. Only saved if the NO_WELL_STORAGE is not specified in the OPTIONS block. The multi-aquifer well number is written to (ID1) and (ID2). The multi-aquifer well volume (VOLUME) is saved as an auxiliary data item for this flow term.
CONSTANT	6	1 / nmawwells	Calculated flow to maintain constant head in multi-aquifer well. The multi-aquifer well number is written to (ID1) and (ID2).
FROM-MVR	6	1 / nmawwells	Calculated flow to lake from the MVR Package. Only saved if MVR Package is used in the MAW Package. The lake number is written to (ID1) and (ID2).
RATE-TO-MVR	6	1 / nmawwells	Calculated pumping rate from the multi-aquifer well to the MVR Package. Only saved if MVR Package is used in the MAW Package. The multi-aquifer well number is written to (ID1) and (ID2).

Flow term	IMETH	NDAT / NLIST	Description
FW-RATE-TO-MVR	6	1 / nmawwells	Calculated flowing well flow from a multi-aquifer well to the MVR Package. Only saved if MVR Package is used in the MAW Package and the FLOWING_WELLS is specified in the OPTIONS block. The multi-aquifer well number is written to (ID1) and (ID2).
AUXILIARY	6	naux+1 / nmawwells	Auxiliary variables, if specified in the MAW Package, are saved to this flow term. The first entry of the DATA2D column has a value of zero. The multi-aquifer well number is written to (ID1) and (ID2).

Table 25: Data written to the SFR Package binary output file. Flow terms are listed in the order they are written to the SFR Package binary output file

Flow term	IMETH	NDAT / NLIST	Description
FLOW-JA-FACE	6	2 / $\sum_{n=1}^{\mathrm{maxbound}} \mathrm{nconn}_n$	Connection flow from reach (ID1) to unmanaged and managed (tributaries) connections (ID2). The cross-sectional flow area (FLOW-AREA) is saved as an auxiliary data item for this flow term.
GWF	6	2 / maxbound	Calculated flow from reach (ID1) to GWF cell (ID2). The reach-aquifer flow area (FLOW-AREA) is saved as an auxiliary data item for this flow term.
EXT-INFLOW	6	1 / maxbound	Specified inflow to reach. The reach number is written to (ID1) and (ID2).
RUNOFF	6	1 / maxbound	Specified runoff to reach. The reach number is written to (ID1) and (ID2).
RAIN	6	1 / maxbound	Specified rainfall on reach. The reach number is written to (ID1) and (ID2).
EVAPORATION	6	1 / maxbound	Calculated evaporation from reach. The reach number is written to (ID1) and (ID2).
EXT-OUTFLOW	6	1 / maxbound	Calculated outflow to external boundaries (is nonzero for reaches with no downstream connections). The reach number is written to (ID1) and (ID2).
FROM-MVR	6	1 / maxbound	Calculated flow to reach from the MVR Package. Only saved if MVR Package is used in the SFR Package. The reach number is written to (ID1) and (ID2).
TO-MVR	6	1 / maxbound	Calculated flow from reach to the MVR Package. Only saved if MVR Package is used in the SFR Package. The reach number is written to (ID1) and (ID2).
AUXILIARY	6	naux+1 / maxbound	Auxiliary variables, if specified in the SFR Package, are saved to this flow term. The first entry of the DATA2D column has a value of zero. The reach number is written to (ID1) and (ID2).

Table 26: Data written to the UZF Package binary output file. Flow terms are listed in the order they are written to the UZF Package binary output file

Flow term	IMETH	NDAT / NLIST	Description
FLOW-JA-FACE	6	1 / 2*nlen	Connection flow from UZF cell (ID1) to a connected UZF cell (ID2). `nlen` is calculated as the number of uzf cells with `vertcon` values greater than 0.
GWF	6	2 / maxbound	Calculated flow from UZF cell (ID1) to GWF cell (ID2). The UZF cell-aquifer flow area (FLOW-AREA) is saved as an auxiliary data item for this flow term.
INFILTRATION	6	1 / maxbound	Specified infiltration to UZF cell. The UZF cell number is written to (ID1) and (ID2).
REJ-INF	6	1 / maxbound	Calculated rejected infiltration from the UZF cell. The UZF cell number is written to (ID1) and (ID2).
UZET	6	1 / maxbound	Calculated evaporation from the UZF cell. The UZF cell number is written to (ID1) and (ID2).
STORAGE	6	2 / maxbound	Calculated flow from storage for the UZF cell. The UZF cell number is written to (ID1) and (ID2).
FROM-MVR	6	1 / maxbound	Calculated flow to the UZF cell from the MVR Package. Only saved if MVR Package is used in the UZF Package. The UZF cell number is written to (ID1) and (ID2).
REJ-INF-TO-MVR	6	1 / maxbound	Calculated rejected infiltration flow from the UZF cell to the MVR Package. Only saved if MVR Package is used in the UZF Package. The UZF cell number is written to (ID1) and (ID2).
AUXILIARY	6	naux+1 / maxbound	Auxiliary variables, if specified in the UZF Package, are saved to this flow term. The first entry of the DATA2D column has a value of zero. The UZF cell number is written to (ID1) and (ID2).

Observation Output File

When the BINARY option is used to open an observation output file (see section "Observation (OBS) Utility"), the output file has the following form. Record 1 has a length of 100 bytes.

Record 1: `TYPE, PRECISION, LENOBSNAME` *(Record 1 includes 85 blanks following LENOBSNAME.)*
Record 2: `NOBS`
Record 3: `OBSNAME(1), OBSNAME(2), ..., OBSNAME(NOBS)`

Repeat for each time step.

Record 4: `TIME, SIMVALUE(1), SIMVALUE(2), ..., SIMVALUE(NOBS)`

where

  `TYPE` (bytes 1–4 of Record 1) is "cont " — "cont" indicates the file contains continuous observations ;
  `PRECISION` (bytes 6–11 of Record 1) is either "single" or "double" — "single" indicates that floating-point values are written in single precision (4 bytes), and "double" indicates double precision (8 bytes);
  `LENOBSNAME` (bytes 12–15 of Record 1) is an integer indicating the number of characters used to store each observation name in following records (in the initial release of MODFLOW 6, LENOBSNAME equals 40);
  `NOBS` (4-byte integer) is the number of observations recorded in the file;
  `OBSNAME` (LENOBSNAME bytes) is an observation name;
  `TIME` (floating-point) is the simulation time; and
  `SIMVALUE` (floating-point) is the simulated value.

References Cited

Anderman, E.R., and Hill, M.C., 2000, MODFLOW-2000, the U.S. Geological Survey modular ground-water model-documentation of the Hydrogeologic-Unit Flow (HUF) Package: U.S. Geological Survey Open-File Report 2000–342, 89 p.

Anderman, E.R., and Hill, M.C., 2003, MODFLOW-2000, the U.S. Geological Survey modular ground-water model—Three additions to the Hydrogeologic-Unit Flow (HUF) Package: Alternative storage for the uppermost active cells, flows in hydrogeologic units, and the hydraulic-conductivity depth-dependence (KDEP) capability: U.S. Geological Survey Open-File Report 2003–347, 36 p.

Bakker, Mark, Schaars, Frans, Hughes, J.D., Langevin, C.D., and Dausman, A.M., 2013, Documentation of the seawater intrusion (SWI2) package for MODFLOW: U.S. Geological Survey Techniques and Methods, book 6, chap. A46, 47 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/tm6A46.

Banta, E.R., 2000, MODFLOW-2000, the U.S. Geological Survey Modular Ground-Water Model; documentation of packages for simulating evapotranspiration with a segmented function (ETS1) and drains with return flow (DRT1): U.S. Geological Survey Open File Report 2000–466, 127 p.

Banta, E.R., 2011, MODFLOW-CDSS, a version of MODFLOW-2005 with modifications for Colorado Decision Support Systems: U.S. Geological Survey Open-File Report 2011–1213, 19 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/ofr20111213.

Fenske, J.P., Leake, S.A., and Prudic, D.E., 1996, Documentation of a computer program (RES1) to simulate leakage from reservoirs using the modular finite-difference ground-water flow model (MODFLOW): U.S. Geological Survey Open-File Report 96–364, 51 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/ofr96364.

Halford, K.J., and Hanson, R.T., 2002, User guide for the drawdown-limited, multi-node well (MNW) package for the U.S. Geological Survey's modular three-dimensional finite-difference ground-water flow model, versions MODFLOW-96 and MODFLOW-2000: U.S. Geological Survey Open-File Report 02–293, 33 p.

Hanson, R.T., and Leake, S.A., 1999, Documentation for HYDMOD—A program for extracting and processing time-series data from the U.S. Geological Survey's modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 98–564, 57 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/ofr98564.

Harbaugh, A.W., 2005, MODFLOW-2005, the U.S. Geological Survey modular ground-water model—the Ground-Water Flow Process: U.S. Geological Survey Techniques and Methods, book 6, chap. A16, variously paged, accessed June 27, 2017, at https://pubs.usgs.gov/tm/2005/tm6A16/.

Hill, M.C., Banta, E.R., Harbaugh, A.W., and Anderman, E.R., 2000, MODFLOW-2000, the U.S. Geological Survey modular ground-water model—User guide to the observation, sensitivity, and parameter-estimation processes and three post-processing programs: U.S. Geological Survey Open-File Report 00–184, 210 p.

Hoffmann, Jörn, Leake, S.A., Galloway, D.L., and Wilson, A.M., 2003, MODFLOW-2000 Ground-Water Model—User Guide to the Subsidence and Aquifer-System Compaction (SUB) Package: U.S. Geological Survey Open-File Report 03–233, 44 p., accessed June 27, 2017, at https://pubs.usgs.gov/of/2003/ofr03-233/.

Hsieh, P.A., and Freckleton, J.R., 1993, Documentation of a computer program to simulate horizontal-flow barriers using the U.S. Geological Survey's modular three-dimensional finite-difference ground-water flow model: U.S. Geological Survey Open-File Report 92–477, 32 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/ofr92477.

Hughes, J.D., Langevin, C.D., Chartier, K.L., and White, J.T., 2012, Documentation of the Surface-Water Routing (SWR1) Process for modeling surface-water flow with the U.S. Geological Survey modular groundwater model (MODFLOW-2005): U.S. Geological Survey Techniques and Methods, book 6, chap. A40 (Version 1.0), 113 p., accessed June 27, 2017, at https://pubs.usgs.gov/tm/6a40/.

Hughes, J.D., Langevin, C.D., and Banta, E.R., 2017, Documentation for the MODFLOW 6 framework: U.S. Geological Survey Techniques and Methods, book 6, chap. A57, 36 p., accessed August 4, 2017, at https://doi.org/10.3133/tm6A57.

Konikow, L.F., Hornberger, G.Z., Halford, K.J., and Hanson, R.T., 2009, Revised multi-node well (MNW2) package for MODFLOW ground-water flow model: U.S. Geological Survey Techniques and Methods, book 6, chap. A30, 67 p., accessed June 27, 2017, at https://pubs.usgs.gov/tm/tm6a30/.

Langevin, C.D., Hughes, J.D., Provost, A.M., Banta, E.R., Niswonger, R.G., and Panday, Sorab, 2017, Documentation for the MODFLOW 6 Groundwater Flow (GWF) Model: U.S. Geological Survey Techniques and Methods, book 6, chap. A55, 197 p., accessed August 4, 2017, at https://doi.org/10.3133/tm6A55.

Leake, S.A., and Galloway, D.L., 2007, MODFLOW Ground-water model—User guide to the Subsidence and Aquifer-System Compaction Package (SUB-WT) for Water-Table Aquifers: U.S. Geological Survey Techniques and Methods, book 6, chap. A23, 42 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/tm6A23.

Leake, S.A., and Lilly, M.R., 1997, Documentation of computer program (FHB1) for assignment of transient specified-flow and specified-head boundaries in applications of the modular finite-diference ground-water flow model (MODFLOW): U.S. Geological Survey Open-File Report 97–571, 50 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/ofr97571.

Leake, S.A., and Prudic, D.E., 1991, Documentation of a computer program to simulate aquifer-system compaction using the modular finite-difference ground-water flow model: U.S. Geological Survey Techniques of Water-Resources Investigations, book 6, chap. A2, 68 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/twri06A2.

Maddock, Thomas, III, Baird, K.J., Hanson, R.T., Schmid, Wolfgang, and Ajami, Hoori, 2012, RIP-ET—A Riparian Evapotranspiration Package for MODFLOW-2005: U.S. Geological Survey Techniques and Methods, book 6, chap. A39, 76 p., accessed June 27, 2017, at https://pubs.usgs.gov/tm/tm6a39/.

Merritt, M.L., and Konikow, L.F., 2000, Documentation of a computer program to simulate lake-aquifer interaction using the MODFLOW ground-water flow model and the MOC3D solute-transport model: U.S. Geological Survey Water-Resources Investigations Report 00–4167, 146 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/wri004167.

Niswonger, R.G., and Prudic, D.E., 2005, Documentation of the Streamflow-Routing (SFR2) Package to include unsaturated flow beneath streams—A modification to SFR1: U.S. Geological Survey Techniques and Methods, book 6, chap. A13, 50 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/tm6A13.

Niswonger, R.G., Prudic, D.E., and Regan, R.S., 2006, Documentation of the Unsaturated-Zone Flow (UZF1) Package for modeling unsaturated flow between the land surface and the water table with MODFLOW-2005: U.S. Geological Survey Techniques and Methods, book 6, chap. A19, 62 p., accessed June 27, 2017, at https://pubs.usgs.gov/tm/2006/tm6a19/.

Panday, Sorab, Langevin, C.D., Niswonger, R.G., Ibaraki, Motomu, and Hughes, J.D., 2013, MODFLOW-USG version 1—An unstructured grid version of MODFLOW for simulating ground-water flow and tightly coupled processes using a control volume finite-difference formulation: U.S. Geological Survey Techniques and Methods, book 6, chap. A45, 66 p., accessed June 27, 2017, at https://pubs.usgs.gov/tm/06/a45/.

Provost, A.M., Langevin, C.D., and Hughes, J.D., 2017, Documentation for the "XT3D" Option in the Node Property Flow (NPF) Package of MODFLOW 6: U.S. Geological Survey Techniques and Methods, book 6, chap. A56, 46 p., accessed August 4, 2017, at https://doi.org/10.3133/tm6A56.

Prudic, D.E., 1989, Documentation of a computer program to simulate stream-aquifer relations using a modular, finite-difference, ground-water flow model: U.S. Geological Survey Open-File Report 88–729, 113 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/ofr88729.

Prudic, D.E., Konikow, L.F., and Banta, E.R., 2004, A New Streamflow-Routing (SFR1) Package to simulate stream-aquifer interaction with MODFLOW-2000: U.S. Geological Survey Open File Report 2004–1042, 104 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/ofr20041042.

Zheng, Chunmiao, Hill, M.C., and Hsieh, P.A., 2001, MODFLOW-2000, the U.S. Geological Survey Modular Ground-Water Model—User guide to the LMT6 package, the linkage with MT3DMS for multi-species mass transport modeling: U.S. Geological Survey Open-File Report 01–82, 43 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/ofr0182.

Appendix A. List of Blocks

Table A–1: List of block names organized by component and input file type. OPEN/CLOSE indicates whether or not the block information can be contained in separate file

Component	FTYPE	Blockname	OPEN/CLOSE
SIM	NAM	OPTIONS	yes
SIM	NAM	TIMING	yes
SIM	NAM	MODELS	yes
SIM	NAM	EXCHANGES	yes
SIM	NAM	SOLUTIONGROUP	yes
SIM	TDIS	OPTIONS	yes
SIM	TDIS	DIMENSIONS	yes
SIM	TDIS	PERIODDATA	yes
EXG	GWFGWF	OPTIONS	yes
EXG	GWFGWF	DIMENSIONS	yes
EXG	GWFGWF	EXCHANGEDATA	yes
SLN	IMS	OPTIONS	yes
SLN	IMS	NONLINEAR	yes
SLN	IMS	LINEAR	yes
GWF	NAM	OPTIONS	yes
GWF	NAM	PACKAGES	yes
GWF	DIS	OPTIONS	yes
GWF	DIS	DIMENSIONS	yes
GWF	DIS	GRIDDATA	no
GWF	DISV	OPTIONS	yes
GWF	DISV	DIMENSIONS	yes
GWF	DISV	GRIDDATA	no
GWF	DISV	VERTICES	yes
GWF	DISV	CELL2D	yes
GWF	DISU	OPTIONS	yes
GWF	DISU	DIMENSIONS	yes
GWF	DISU	GRIDDATA	no
GWF	DISU	CONNECTIONDATA	yes
GWF	DISU	VERTICES	yes
GWF	DISU	CELL2D	yes
GWF	IC	GRIDDATA	no
GWF	NPF	OPTIONS	yes
GWF	NPF	GRIDDATA	no
GWF	STO	OPTIONS	yes
GWF	STO	GRIDDATA	no
GWF	STO	PERIOD	yes
GWF	HFB	OPTIONS	yes
GWF	HFB	DIMENSIONS	yes
GWF	HFB	PERIOD	yes
GWF	CHD	OPTIONS	yes
GWF	CHD	DIMENSIONS	yes
GWF	CHD	PERIOD	yes
GWF	WEL	OPTIONS	yes
GWF	WEL	DIMENSIONS	yes
GWF	WEL	PERIOD	yes
GWF	DRN	OPTIONS	yes

Table A–1.   List of block names organized by component and input file type. OPEN/CLOSE indicates whether or not the block information can be contained in separate file.—Continued

Component	FTYPE	Blockname	OPEN/CLOSE
GWF	DRN	DIMENSIONS	yes
GWF	DRN	PERIOD	yes
GWF	RIV	OPTIONS	yes
GWF	RIV	DIMENSIONS	yes
GWF	RIV	PERIOD	yes
GWF	GHB	OPTIONS	yes
GWF	GHB	DIMENSIONS	yes
GWF	GHB	PERIOD	yes
GWF	RCH	OPTIONS	yes
GWF	RCH	DIMENSIONS	yes
GWF	RCH	PERIOD	yes
GWF	RCHA	OPTIONS	yes
GWF	RCHA	PERIOD	yes
GWF	EVT	OPTIONS	yes
GWF	EVT	DIMENSIONS	yes
GWF	EVT	PERIOD	yes
GWF	EVTA	OPTIONS	yes
GWF	EVTA	PERIOD	yes
GWF	MAW	OPTIONS	yes
GWF	MAW	DIMENSIONS	yes
GWF	MAW	PACKAGEDATA	yes
GWF	MAW	CONNECTIONDATA	yes
GWF	MAW	PERIOD	yes
GWF	SFR	OPTIONS	yes
GWF	SFR	DIMENSIONS	yes
GWF	SFR	PACKAGEDATA	yes
GWF	SFR	CONNECTIONDATA	yes
GWF	SFR	DIVERSIONS	yes
GWF	SFR	PERIOD	yes
GWF	LAK	OPTIONS	yes
GWF	LAK	DIMENSIONS	yes
GWF	LAK	PACKAGEDATA	yes
GWF	LAK	CONNECTIONDATA	yes
GWF	LAK	TABLES	yes
GWF	LAK	OUTLETS	yes
GWF	LAK	PERIOD	yes
GWF	LAKTAB	DIMENSIONS	yes
GWF	LAKTAB	TABLE	yes
GWF	UZF	OPTIONS	yes
GWF	UZF	DIMENSIONS	yes
GWF	UZF	PACKAGEDATA	yes
GWF	UZF	PERIOD	yes
GWF	MVR	OPTIONS	yes
GWF	MVR	DIMENSIONS	yes
GWF	MVR	PACKAGES	yes
GWF	MVR	PERIOD	yes
GWF	GNC	OPTIONS	yes
GWF	GNC	DIMENSIONS	yes
GWF	GNC	GNCDATA	yes

Table A–1. List of block names organized by component and input file type. OPEN/CLOSE indicates whether or not the block information can be contained in separate file.—Continued

Component	FTYPE	Blockname	OPEN/CLOSE
GWF	OC	OPTIONS	yes
GWF	OC	PERIOD	yes
UTL	OBS	OPTIONS	yes
UTL	OBS	CONTINUOUS	yes