USGS
science for a changing world

Prepared in cooperation with the U.S. Geological Survey Water Availability and Use Science Program

MODFLOW 6—Description of Input and Output

Version mf6.2.1—February 18, 2021

U.S. Department of the Interior
U.S. Geological Survey

Cover. Binary computer code illustration.

Contents

Introduction 1
Running a Simulation 1
Form of Input Instructions 2
- Block and Keyword Input 2
- Specification of Block Information in OPEN/CLOSE File 4
- File Name Input 4
- Lengths of Character Variables 4
- Integer and Floating Point Variables 5

Simulation Name File 6
- Structure of Blocks 6
- Explanation of Variables 6
- Example Input File 8

Temporal Discretization (TDIS) Package 9
- Structure of Blocks 9
- Explanation of Variables 9
- Example Input File 10

Groundwater Flow (GWF) Model Input 11
- Information for Existing MODFLOW Users 11
- Array Input (READARRAY) 16
  - READARRAY Control Line 16
  - READARRAY Variable Descriptions 17
  - READARRAY Examples 18
- List Input 19
- Units of Length and Time 20
- Steady-State Simulations 21
- Volumetric Budget 21
- Cell-By-Cell Flows 22
- GWF Model Name File 24
  - Structure of Blocks 24
  - Explanation of Variables 24
  - Example Input File 26
- Structured Discretization (DIS) Input File 27
  - Structure of Blocks 27
  - Explanation of Variables 27
  - Example Input File 28

Discretization by Vertices (DISV) Input File . . . 29
Structure of Blocks . . . 29
Explanation of Variables . . . 30
Example Input File . . . 31
Unstructured Discretization (DISU) Input File . . . 33
Structure of Blocks . . . 33
Explanation of Variables . . . 34
Example Input File . . . 36
Initial Conditions (IC) Package . . . 38
Structure of Blocks . . . 38
Explanation of Variables . . . 38
Example Input File . . . 38
Output Control (OC) Option . . . 39
Structure of Blocks . . . 39
Explanation of Variables . . . 39
Example Input File . . . 40
Observation (OBS) Utility for a GWF Model . . . 42
Structure of Blocks . . . 42
Explanation of Variables . . . 42
Example Observation Input File . . . 43
Node Property Flow (NPF) Package . . . 44
Structure of Blocks . . . 44
Explanation of Variables . . . 44
Example Input File . . . 47
Horizontal Flow Barrier (HFB) Package . . . 48
Structure of Blocks . . . 48
Explanation of Variables . . . 48
Example Input File . . . 49
Storage (STO) Package . . . 50
Structure of Blocks . . . 50
Explanation of Variables . . . 50
Example Input File . . . 51
Skeletal Storage, Compaction, and Subsidence (CSUB) Package . . . 52
Structure of Blocks . . . 52
Explanation of Variables . . . 53
Example Input File . . . 57
Available observation types . . . 58
Example Observation Input File . . . 60
Buoyancy (BUY) Package . . . 62
Stress Packages . . . 62
Structure of Blocks . . . 63
Explanation of Variables . . . 64
Example Input File . . . 65

Constant-Head (CHD) Package 66
Structure of Blocks 66
Explanation of Variables 66
Example Input File 68
Available observation types 68
Example Observation Input File 69
Well (WEL) Package 70
Structure of Blocks 70
Explanation of Variables 70
Example Input File 72
Available observation types 73
Example Observation Input File 73
Drain (DRN) Package 74
Structure of Blocks 74
Explanation of Variables 74
Example Input File 76
Available observation types 76
Example Observation Input File 77
River (RIV) Package 78
Structure of Blocks 78
Explanation of Variables 78
Example Input File 80
Available observation types 80
Example Observation Input File 81
General-Head Boundary (GHB) Package 82
Structure of Blocks 82
Explanation of Variables 82
Example Input File 84
Available observation types 84
Example Observation Input File 85
Recharge (RCH) Package – List-Based Input 86
Structure of Blocks 86
Explanation of Variables 87
Example Input File 88
Available observation types 88
Example Observation Input File 89
Recharge (RCH) Package – Array-Based Input 90
Structure of Blocks 90
Explanation of Variables 90
Example Input File 92

Evapotranspiration (EVT) Package – List-Based Input . . . 93
Structure of Blocks . . . 93
Explanation of Variables . . . 94
Example Input File . . . 96
Available observation types . . . 96
Example Observation Input File . . . 96
Evapotranspiration (EVT) Package – Array-Based Input . . . 98
Structure of Blocks . . . 98
Explanation of Variables . . . 99
Example Input File . . . 100
Multi-Aquifer Well (MAW) Package . . . 101
Structure of Blocks . . . 101
Explanation of Variables . . . 102
Example Input File – Conductance Calculated using Thiem Equation . . . 107
Example Input File – Conductance Calculated using Screen Geometry . . . 108
Example Input File – Flowing Well with Conductance Specified . . . 108
Available observation types . . . 109
Example Observation Input File . . . 110
Streamflow Routing (SFR) Package . . . 111
Structure of Blocks . . . 111
Explanation of Variables . . . 112
Example Input File . . . 118
Available observation types . . . 119
Example Observation Input File . . . 121
Lake (LAK) Package . . . 122
Structure of Blocks . . . 122
Explanation of Variables . . . 123
Example Input File . . . 129
Available observation types . . . 130
Example Observation Input File . . . 132
Lake Table Input File . . . 133
Structure of Blocks . . . 133
Explanation of Variables . . . 133
Example Input File . . . 133
Unsaturated Zone Flow (UZF) Package . . . 135
Structure of Blocks . . . 135
Explanation of Variables . . . 136
Example Input File . . . 139
Available observation types . . . 140
Example Observation Input File . . . 141

Water Mover (MVR) Package . . . 142
Structure of Blocks . . . 143
Explanation of Variables . . . 144
Example Input File . . . 145
Ghost-Node Correction (GNC) Package . . . 147
Structure of Blocks . . . 147
Explanation of Variables . . . 147
Example Input File . . . 148
Groundwater Flow (GWF) Exchange . . . 150
Structure of Blocks . . . 150
Explanation of Variables . . . 150
Example Input File . . . 152
Available observation types . . . 153
Example Observation Input File . . . 153
Groundwater Transport (GWT) Model Input . . . 155
Information for Existing Solute Transport Modelers . . . 156
Units of Length and Time . . . 157
Solute Mass Budget . . . 157
Time Stepping . . . 157
GWT Model Name File . . . 158
Structure of Blocks . . . 158
Explanation of Variables . . . 158
Example Input File . . . 159
Initial Conditions (IC) Package . . . 161
Structure of Blocks . . . 161
Explanation of Variables . . . 161
Example Input File . . . 161
Output Control (OC) Option . . . 162
Structure of Blocks . . . 162
Explanation of Variables . . . 162
Example Input File . . . 163
Observation (OBS) Utility for a GWT Model . . . 164
Structure of Blocks . . . 164
Explanation of Variables . . . 164
Example Observation Input File . . . 165
Advection (ADV) Package . . . 166
Structure of Blocks . . . 166
Explanation of Variables . . . 166
Example Input File . . . 166
Dispersion (DSP) Package . . . 167
Structure of Blocks . . . 167
Explanation of Variables . . . 167
Example Input File . . . 168

Source and Sink Mixing (SSM) Package . . . 169
Structure of Blocks . . . 169
Explanation of Variables . . . 169
Example Input File . . . 170
Mobile Storage and Transfer (MST) Package . . . 171
Structure of Blocks . . . 171
Explanation of Variables . . . 171
Example Input File . . . 172
Immobile Storage and Transfer (IST) Package . . . 173
Structure of Blocks . . . 173
Explanation of Variables . . . 173
Example Input File . . . 174
Constant Concentration (CNC) Package . . . 176
Structure of Blocks . . . 176
Explanation of Variables . . . 176
Example Input File . . . 178
Available observation types . . . 178
Example Observation Input File . . . 178
Mass Source Loading (SRC) Package . . . 180
Structure of Blocks . . . 180
Explanation of Variables . . . 180
Example Input File . . . 182
Available observation types . . . 182
Example Observation Input File . . . 182
Streamflow Transport (SFT) Package . . . 183
Structure of Blocks . . . 183
Explanation of Variables . . . 183
Example Input File . . . 186
Available observation types . . . 186
Example Observation Input File . . . 188
Lake Transport (LKT) Package . . . 189
Structure of Blocks . . . 189
Explanation of Variables . . . 189
Example Input File . . . 192
Available observation types . . . 192
Example Observation Input File . . . 194
Multi-Aquifer Well Transport (MWT) Package . . . 195
Structure of Blocks . . . 195
Explanation of Variables . . . 195
Example Input File . . . 198
Available observation types . . . 198
Example Observation Input File . . . 199

Unsaturated Zone Transport (UZT) Package . . . 201
Structure of Blocks . . . 201
Explanation of Variables . . . 201
Example Input File . . . 204
Available observation types . . . 204
Example Observation Input File . . . 205
Flow Model Interface (FMI) Package . . . 207
Structure of Blocks . . . 208
Explanation of Variables . . . 208
Example Input File . . . 209
Mover Transport (MVT) Package . . . 210
Structure of Blocks . . . 210
Explanation of Variables . . . 210
Example Input File . . . 210
Iterative Model Solution . . . 211
Structure of Blocks . . . 211
Explanation of Variables . . . 211
Example Input File . . . 216
Observation (OBS) Utility . . . 218
Structure of Blocks . . . 218
Explanation of Variables . . . 218
Available Observation Types . . . 219
GWF Observations . . . 219
GWT Observations . . . 228
Time-Variable Input . . . 234
Time Series . . . 234
Time-Series Files . . . 234
Explanation of Variables . . . 235
Using Time Series in a Package . . . 236
Time-Array Series . . . 237
Time-Array-Series Files . . . 237
Explanation of Variables . . . 238
Using Time-Array Series in a Package . . . 238
Description of Binary Output Files for the Groundwater Flow (GWF) and Groundwater Transport (GWT) Models . . . 240
Binary Grid File . . . 241
DIS Grids . . . 242
DISV Grids . . . 243
DISU Grids . . . 245
Dependent Variable File . . . 247
DIS Grids . . . 247
DISV Grids . . . 247
DISU Grids . . . 248

Advanced Flow and Transport Packages 249
Model Budget Files 250
Format of Budget File 250
Intercell Flows 251
Variations for Discretization Types 251
Budget File Contents for the GWF Model 253
GWF Model CSUB Package 255
GWF Model LAK, MAW, SFR, and UZF Packages 255
Budget File Contents for the GWT Model 262
GWT Model LKT, MWT, SFT, and UZT Packages 263
Observation Output File 264
References Cited R–1
Appendix A. List of Blocks A–1

Figures

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) 29
2. Illustration of a simple stream network having seven reaches with a junction having two reaches, a confluence of two reaches, and the resulting reach connectivity 111

Tables

1. Character variable maximum sizes 4
2. Model types available in Version mf6.2.1 8
3. Exchange types available in Version mf6.2.1 8
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 17
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 18
5. Ftype values described in this report. The `Pname` column indicates whether or not a package name can be provided in the name file 25
6. Available GWF model observation types 43
7. Available CSUB Package observation types 58
8. Description of density terms for stress packages 63
9. Available CHD Package observation types 68
10. Available WEL Package observation types 73
11. Available DRN Package observation types 77
12. Available RIV Package observation types 81

13. Available GHB Package observation types 85
14. Available RCH Package observation types 89
15. Available EVT Package observation types 96
16. Available MAW Package observation types 109
17. Available SFR Package observation types 120
18. Available LAK Package observation types 131
19. Available UZF Package observation types 140
20. Available GWF-GWF Exchange observation types 153
21. Ftype values described in this report. The `Pname` column indicates whether or not a package name can be provided in the name file. The capability to provide a package name also indicates that the GWT Model can have more than one package of that Ftype 159
22. Available GWT model observation types 165
23. Available CNC Package observation types 178
24. Available SRC Package observation types 182
25. Available SFT Package observation types 187
26. Available LKT Package observation types 193
27. Available MWT Package observation types 198
28. Available UZT Package observation types 205
29. IMS variable values for the available complexity options. 216
30. Available observation types for the GWF Model 220
31. Available observation types for the GWT Model 228
32. Dependent variable written for advanced flow and transport packages 249
33. Budget file variations that depend on discretization package type 252
34. Types of information that may be contained in the GWF Model budget file 253
34. Types of information that may be contained in the GWF Model budget file 254
35. Data written to the CSUB Package compaction binary output files 255
36. 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 255
37. 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 257
38. 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 259
39. 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 261
40. Types of information that may be contained in the GWT Model budget file. All terms represent solute flows in dimensions of mass per time 262
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 A–1

Blank page

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; Langevin and others, 2020). 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 6
               U.S. GEOLOGICAL SURVEY MODULAR HYDROLOGIC MODEL
                          VERSION 6.2.1 02/18/2021

   MODFLOW 6 compiled Feb 18 2021 08:24:05 with IFORT compiler (ver. 19.10.2)

This software has been approved for release by the U.S. Geological
Survey (USGS). Although the software has been subjected to rigorous
review, the USGS reserves the right to update the software as needed
pursuant to further analysis and review. No warranty, expressed or
implied, is made by the USGS or the U.S. Government as to the
functionality of the software and related material nor shall the
fact of release constitute any such warranty. Furthermore, the
software is released on condition that neither the USGS nor the U.S.
Government shall be held liable for any damages resulting from its
authorized or unauthorized use. Also refer to the USGS Water
Resources Software User Rights Notice for complete use, copyright,
and distribution information.


 Run start date and time (yyyy/mm/dd hh:mm:ss): 2021/02/18 8:32:29

 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): 2021/02/18 8:32:29
 Elapsed run time: 0.076 Seconds

 Normal termination of simulation.
```

MODFLOW 6 includes a number of switches that can be passed to the program in order to get additional information. The available switches can be found by running MODFLOW 6 with the -h switch, for help. In this case MODFLOW 6 will produce the following.

```
mf6.exe - MODFLOW 6.2.1 02/18/2021 (compiled Feb 18 2021 08:24:05)
usage: mf6.exe            run MODFLOW 6 using "mfsim.nam"
   or: mf6.exe [options]  retrieve program information

Options  GNU long option Meaning
 -h, -?   --help         Show this message
 -v       --version      Display program version information.
 -dev     --develop      Display program develop option mode.
 -c       --compiler     Display compiler information.
```

```
 -s       --silent       All STDOUT to mfsim.stdout.
 -l <str> --level <str>  STDOUT output to screen based on <str>.
                         <str>=summary Limited output to STDOUT.
                         <str>=debug  Enhanced output to STDOUT.
 -m <str> --mode <str>   MODFLOW 6 simulation mode based on <str>.
                         <str>=validate Check model input for
                                        errors but do assemble or
                                        solve matrix equations or
                                        write solution output.

Bug reporting and contributions are welcome from the community.
Questions can be asked on the issues page[1]. Before creating a new
issue, please take a moment to search and make sure a similar issue
does not already exist. If one does exist, you can comment (most
simply even with just :+1:) to show your support for that issue.

[1] https://github.com/MODFLOW-USGS/modflow6/issues
```

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.

```
ERROR REPORT:

  1. mf6.exe: mfsim.nam is not present in working directory.

2
```

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. 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

Integer and Floating Point Variables

MODFLOW 6 uses integer and floating point variables throughout the program. The sizes of these variables are defined in a single module within the program. Information about the precision, range, and size of integers and floating point real variables is written to the top of the simulation list file:

```
MODFLOW was compiled using uniform precision.

Real Variables
  KIND: 8
  TINY (smallest non-zero value): 2.225074-308
  HUGE (largest value):  1.797693+308
  PRECISION: 15
  BIT SIZE: 64

Integer Variables
  KIND: 4
  HUGE (largest value): 2147483647
  BIT SIZE: 32

Long Integer Variables
  KIND: 8
  HUGE (largest value): 9223372036854775807
  BIT SIZE: 64

Logical Variables
  KIND: 4
  BIT SIZE: 32
```

This information indicates that real variables have about 15 digits of precision. The smallest positive non-zero value that can be stored is 2.2e-308. The largest value that can be stored is 1.8e+308. If the user enters a value in an input file that cannot be stored, such as 1.9335e-310 for example, then the program can produce unexpected results. This does not affect an exact value of zero, which can be stored accurately. Integer variables also have a maximum and minimum value, which is about 2 billion. Values larger and smaller than this cannot be stored. These numbers are rarely exceeded for most practical problems, but as the size of models (number of nodes) increase into the billions, then the program may need to be recompiled using a larger size for integer variables. Long integers are used to calculate the amount of memory allocated in the memory manager:

```
 MEMORY MANAGER TOTAL STORAGE BY DATA TYPE, IN MEGABYTES
 -------------------------------
                  ALLOCATED
 DATA TYPE         MEMORY
 -------------------------------
 Character      1.53300000E-03
 Logical        4.40000000E-05
 Integer        100.03799
 Real           223.43994
 -------------------------------
 Total          323.47951
 -------------------------------
```

Currently, standard precision 4 byte logical variables are used throughout the program.

Simulation Name File

The simulation name file contains information about simulation options, simulation timing, models that are present in the simulation, how models exchange information, and how models are solved.

The present version of MODFLOW 6 uses the concept of a solution group. For most simulations, a solution group will contain one solution and one model within that solution. The solution group is designed, however, so that multiple solutions can be solved together in a picard iteration loop. This might be used in the future to iteratively couple models that cannot be tightly coupled at the matrix level within a single numerical solution. The solution group is flexible so that multiple solution groups can be included in a simulation. More information on solution groups will be added to this document as new model types and exchanges are added that can take advantage of the concept.

The simulation name file is read from a file in the current working directory with the name “mfsim.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>]
  [MAXERRORS <maxerrors>]
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(:)>
  <slntype> <slnfname> <slnmnames(:)>
  ...
END SOLUTIONGROUP
```

Explanation of Variables

Block: OPTIONS

`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.

`maxerrors`—maximum number of errors that will be stored and printed.

Block: TIMING

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

Block: MODELS

`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 16 characters and must not have blanks within the name. The model name is case insensitive; any lowercase letters are converted and stored as upper case letters.

Block: EXCHANGES

`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.

Block: SOLUTIONGROUP

`group_num`—is the group number of the solution group. Solution groups must be numbered sequentially, starting with group number one.

`mxiter`—is the maximum number of outer iterations for this solution group. The default value is 1. If there is only one solution in the solution group, then MXITER must be 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. The number of model names is determined by the number of model names the user provides on this line.

Table 2. Model types available in Version mf6.2.1.

Mtype	Type of Model
GWF6	Groundwater Flow Model
GWT6	Groundwater Transport Model

Table 3. Exchange types available in Version mf6.2.1.

Exgtype	Type of Exchange
GWF6-GWF6	Exchange between two Groundwater Flow Models. Input for this file is described in a dedicated section in this guide.
GWF6-GWT6	Exchange between a Groundwater Flow Model and a Groundwater Transport Model. In the present version, a filename is required for this exchange and the file must exist, however, nothing is read from this file.

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

Block: OPTIONS

`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.

Block: DIMENSIONS

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

Block: PERIODDATA

`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 has been updated from Langevin and others (2017), and 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 three grid types supported in the GWF Model (DIS, DISV, and DISU), 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," but this option is only available for DIS and DISV grids. 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 through specification of the IDOMAIN array.

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 the Langevin and others (2017) 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 status of a lake can change during the simulation between `ACTIVE`, `INACTIVE`, and `CONSTANT`. The new package contains most of the capabilities available in previous LAK Packages, including the ability to apply recharge and evapotranspiration to underlying cells if the lake is dry. 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. A new Skeletal Storage, Compaction, and Subsidence (CSUB) Package was added to MODFLOW 6 in version 6.1.0. The one-dimensional effective-stress based compaction theory implemented in the CSUB Package is documented in Leake and Galloway (2007). The numerical approach used for delay interbeds in the CSUB package is documented in Hoffmann and others (2003) and uses the same one-dimensional effective-stress based compaction theory as coarse-grained and fine-grained no-delay interbed sediments.

19. MODFLOW 6 contains a flexible new Observation (OBS) capability, which allows the user to define many different types of continuous-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.

20. 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.

    - 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). However, MT3D-USGS can read the head and budget files created by MODFLOW 6, but only if the GWF Model uses the DIS Package. MT3D-USGS will not work with GWF output if the DISV or DISU Packages are used.

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. “(BINARY)” can be specified 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

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
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 (BINARY) 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

Block: OPTIONS

`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.

Block: PACKAGES

`ftype`—is the file type, which must be one of the following character values shown in table 21. 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`
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	
CSUB6	Compaction and Subsidence Package	
BUY6	Buoyancy 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)> -- READARRAY
  DELC
    <delc(nrow)> -- READARRAY
  TOP
    <top(ncol, nrow)> -- READARRAY
  BOTM [LAYERED]
    <botm(ncol, nrow, nlay)> -- READARRAY
  [IDOMAIN [LAYERED]
    <idomain(ncol, nrow, nlay)> -- READARRAY]
END GRIDDATA
```

Explanation of Variables

Block: OPTIONS

`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.

Block: DIMENSIONS

`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.

Block: GRIDDATA

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

`delc`—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 or greater, 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 .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 by 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.

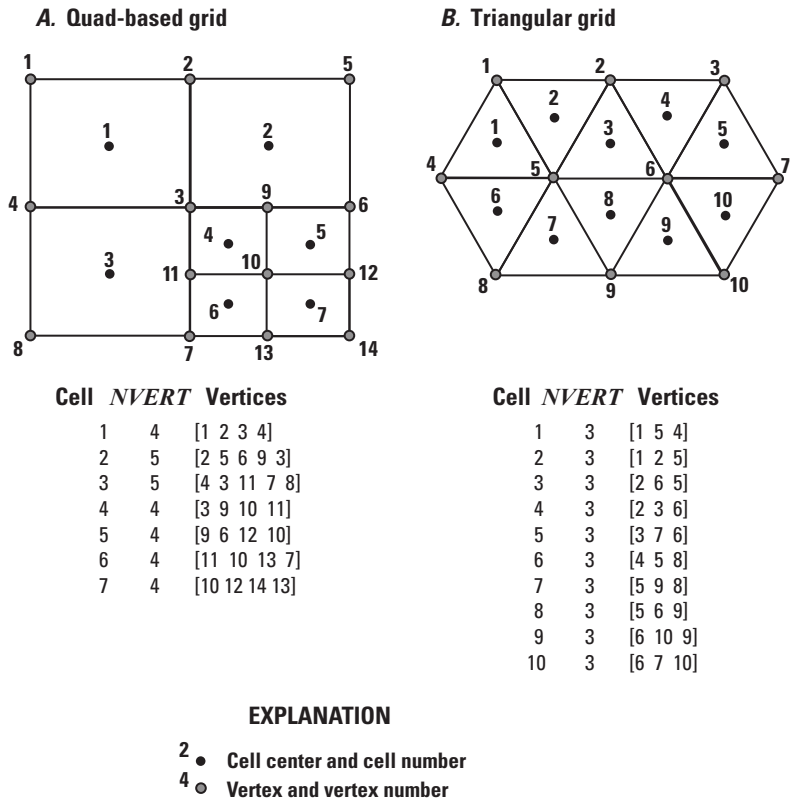

Figure 1. Schematic diagram showing the vertices and cells defined using the Discretization by Vertices Package. The list of vertices used to define each cell must be in clockwise order. From Langevin and others (2017).

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(ncpl)> -- READARRAY
  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

Block: OPTIONS

`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.

Block: DIMENSIONS

`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.

Block: GRIDDATA

`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 or greater, 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.

Block: VERTICES

`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.

Block: CELL2D

`icell2d`—is the CELL2D number. Records in the CELL2D block must be listed in consecutive order from the first to the last.

`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.

The DISU Package does not support the concept of layers, which is different from the DISU implementation in MODFLOW-USG. In MODFLOW 6 all grid input and output for models that use the DISU Package is entered or written as a one-dimensional array of size nodes.

The DISU VERTICES and CELL2D blocks are not required for all simulations. These blocks are required if the XT3D or the SAVE_SPECIFIC_DISCHARGE options are specified in the NPF Package. In general, it is recommended to include the VERTICES and CELL2D blocks.

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)> -- READARRAY
  BOT
    <bot(nodes)> -- READARRAY
  AREA
    <area(nodes)> -- READARRAY
  [IDOMAIN
    <idomain(nodes)> -- READARRAY]
END GRIDDATA
```

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

Block: OPTIONS

`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.

Block: DIMENSIONS

`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. NVERT and the accompanying VERTICES and CELL2D blocks should be specified for most simulations. If the XT3D or SAVE_SPECIFIC_DISCHARGE options are specified in the NPF Package, then this information is required.

Block: GRIDDATA

`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).

`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 or greater, the cell exists in the simulation. IDOMAIN values of -1 cannot be specified for the DISU Package.

Block: CONNECTIONDATA

`iac`—is the number of connections (plus 1) for each cell. The sum of all the entries in 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 divided such that every node and its connectivity list can be 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 converted to positive by the code.

`ihc`—is an index array indicating the direction between node n and all of its m connections. If IHC = 0 then 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 then cell n and cell m are connected in the horizontal direction. If IHC = 2 then 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 to m connection does not equal the value for HWVA for the corresponding n to m 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. The angle varies between zero and 360.0 degrees, where zero degrees points in the positive x-axis direction, and 90 degrees points in the positive y-axis direction. ANGLDEGX is only needed if horizontal anisotropy is specified in the NPF Package, if the XT3D option is used in the NPF Package, or if the SAVE_SPECIFIC_DISCHARGE option is specifed in the NPF Package. ANGLDEGX does not need to be specified if these conditions are not met. 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.

Block: VERTICES

`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.

Block: CELL2D

`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

Block: GRIDDATA

`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 flow budget are written to the Listing File at the end of every stress period. The default printout format for head is 10G11.4. The heads and overall flow budget are also written to the list file if the simulation terminates prematurely due to failed convergence.

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

Block: OPTIONS

`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.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [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

Block: OPTIONS

`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.

Block: CONTINUOUS

`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 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.

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]]
  [SAVE_SPECIFIC_DISCHARGE]
  [SAVE_SATURATION]
  [K22OVERK]
  [K33OVERK]
END OPTIONS
```

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

Explanation of Variables

Block: OPTIONS

`SAVE_FLOWS`—keyword to indicate that budget 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. XT3D requires additional information about the shapes of grid cells. If XT3D is active and the DISU Package is used, then the user will need to provide in the DISU Package the angldegx array in the CONNECTIONDATA block and the VERTICES and CELL2D blocks.

`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.

`SAVE_SPECIFIC_DISCHARGE`—keyword to indicate that x, y, and z components of specific discharge will be calculated at cell centers and written to the budget file, which is specified with "BUDGET SAVE FILE" in Output Control. If this option is activated, then additional information may be required in the discretization packages and the GWF Exchange package (if GWF models are coupled). Specifically, ANGLDEGX must be specified in the CONNECTIONDATA block of the DISU Package; ANGLDEGX must also be specified for the GWF Exchange as an auxiliary variable.

`SAVE_SATURATION`—keyword to indicate that cell saturation will be written to the budget file, which is specified with "BUDGET SAVE FILE" in Output Control. Saturation will be saved to the budget file as an auxiliary variable saved with the DATA-SAT text label. Saturation is a cell variable that ranges from zero to one and can be used by post processing programs to determine how much of a cell volume is saturated. If ICELLTYPE is 0, then saturation is always one.

`K22OVERK`—keyword to indicate that specified K22 is a ratio of K22 divided by K. If this option is specified, then the K22 array entered in the NPF Package will be multiplied by K after being read.

`K33OVERK`—keyword to indicate that specified K33 is a ratio of K33 divided by K. If this option is specified, then the K33 array entered in the NPF Package will be multiplied by K after being read.

Block: GRIDDATA

`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. For the common case in which the user would like to specify the horizontal hydraulic conductivity and the vertical hydraulic conductivity, then K should be assigned as the horizontal hydraulic conductivity, K33 should be assigned as the vertical hydraulic conductivity, and K22 and the three rotation angles should not be specified. When more sophisticated anisotropy is required, then K corresponds to the K11 hydraulic conductivity axis. All included cells (IDOMAIN > 0) must have a K value greater than zero.

`k22`—is the hydraulic conductivity of the second ellipsoid axis (or the ratio of K22/K if the K22OVERK option is specified); 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 set equal to K. For a regular MODFLOW grid (DIS Package is used) in which no rotation angles are specified, K22 is the hydraulic conductivity along columns in the y direction. For an unstructured DISU grid, the user must assign principal x and y axes and provide the angle for each cell face relative to the assigned x direction. All included cells (IDOMAIN > 0) must have a K22 value greater than zero.

`k33`—is the hydraulic conductivity of the third ellipsoid axis (or the ratio of K33/K if the K33OVERK option is specified); for an unrotated case, this is the vertical hydraulic conductivity. When anisotropy is applied, K33 corresponds to the K33 tensor component. All included cells (IDOMAIN > 0) must have a K33 value greater than zero.

`angle1`—is a rotation angle of the hydraulic conductivity tensor in degrees. The angle represents the first of three sequential rotations of the hydraulic conductivity ellipsoid. With the K11, K22, and K33 axes of the ellipsoid initially aligned with the x, y, and z coordinate axes, respectively, ANGLE1 rotates the ellipsoid about its K33 axis (within the x - y plane). A positive value represents counter-clockwise rotation when viewed from any point on the positive K33 axis, looking toward the center of the ellipsoid. A value of zero indicates that the K11 axis lies within the x - z plane. If ANGLE1 is not specified, default values of zero are assigned to ANGLE1, ANGLE2, and ANGLE3, in which case the K11, K22, and K33 axes are aligned with the x, y, and z axes, respectively.

`angle2`—is a rotation angle of the hydraulic conductivity tensor in degrees. The angle represents the second of three sequential rotations of the hydraulic conductivity ellipsoid. Following the rotation by ANGLE1 described above, ANGLE2 rotates the ellipsoid about its K22 axis (out of the x - y plane). An array can be specified for ANGLE2 only if ANGLE1 is also specified. A positive value of ANGLE2 represents clockwise rotation when viewed from any point on the positive K22 axis, looking toward the center of the ellipsoid. A value of zero indicates that the K11 axis lies within the x - y plane. If ANGLE2 is not specified, default values of zero are assigned to ANGLE2 and ANGLE3; connections that are not user-designated as vertical are assumed to be strictly horizontal

(that is, to have no z component to their orientation); and connection lengths are based on horizontal distances.

`angle3`—is a rotation angle of the hydraulic conductivity tensor in degrees. The angle represents the third of three sequential rotations of the hydraulic conductivity ellipsoid. Following the rotations by ANGLE1 and ANGLE2 described above, ANGLE3 rotates the ellipsoid about its K11 axis. An array can be specified for ANGLE3 only if ANGLE1 and ANGLE2 are also specified. An array must be specified for ANGLE3 if ANGLE2 is specified. A positive value of ANGLE3 represents clockwise rotation when viewed from any point on the positive K11 axis, looking toward the center of the ellipsoid. A value of zero indicates that the K22 axis lies within the x - y plane.

`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.

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

Block: OPTIONS

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

Block: DIMENSIONS

`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.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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 the hydraulic characteristic is negative, then the absolute value of HYDCHR acts as a multiplier to the conductance between the two model cells specified as containing the 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

Block: OPTIONS

`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.

Block: GRIDDATA

`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 a mixed formulation of unconfined and confined storage is used when head is below cell top.

`ss`—is specific storage (or the storage coefficient if STORAGECOEFFICIENT is specified as an option). Specific storage values must be greater than or equal to 0. If the CSUB Package is included in the GWF model, specific storage must be zero for every cell.

`sy`—is specific yield. Specific yield values must be greater than or equal to 0. Specific yield does not have to be specified if there are no convertible cells (ICONVERT=0 in every cell).

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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. If the CSUB Package is included in the GWF model, only the first and last stress period can be steady-state.

`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
```

Skeletal Storage, Compaction, and Subsidence (CSUB) Package

Input to the Skeletal Storage, Compaction, and Subsidence (CSUB) Package is read from the file that has type “CSUB6” in the Name File. If the CSUB Package is not included for a model, then storage changes resulting from compaction will not be calculated. Only one CSUB Package can be specified for a GWF model. Only the first and last stress period can be specified to be STEADY-STATE in the STO Package when the CSUB Package is being used in the GWF model. Also the specific storage (SS) must be specified to be zero in the STO Package for every cell.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [BOUNDNAMES]
  [PRINT_INPUT]
  [SAVE_FLOWS]
  [GAMMAW <gammaw>]
  [BETA <beta>]
  [HEAD_BASED]
  [INITIAL_PRECONSOLIDATION_HEAD]
  [NDELAYCELLS <ndelaycells>]
  [COMPRESSION_INDICES]
  [UPDATE_MATERIAL_PROPERTIES]
  [CELL_FRACTION]
  [SPECIFIED_INITIAL_INTERBED_STATE]
  [SPECIFIED_INITIAL_PRECONSOLIDATION_STRESS]
  [SPECIFIED_INITIAL_DELAY_HEAD]
  [EFFECTIVE_STRESS_LAG]
  [STRAIN_CSV_INTERBED FILEOUT <interbedstrain_filename>]
  [STRAIN_CSV_COARSE FILEOUT <coarsestrain_filename>]
  [COMPACTION FILEOUT <compaction_filename>]
  [COMPACTION_ELASTIC FILEOUT <elastic_compaction_filename>]
  [COMPACTION_INELASTIC FILEOUT <inelastic_compaction_filename>]
  [COMPACTION_INTERBED FILEOUT <interbed_compaction_filename>]
  [COMPACTION_COARSE FILEOUT <coarse_compaction_filename>]
  [ZDISPLACEMENT FILEOUT <zdisplacement_filename>]
  [PACKAGE_CONVERGENCE FILEOUT <package_convergence_filename>]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
END OPTIONS
```

```
BEGIN DIMENSIONS
  NINTERBEDS <ninterbeds>
  [MAXSIG0 <maxsig0>]
END DIMENSIONS
```

```
BEGIN GRIDDATA
  CG_SKE_CR
    <cg_ske_cr(nodes)> -- READARRAY
  CG_THETA
    <cg_theta(nodes)> -- READARRAY
  [SGM
    <sgm(nodes)> -- READARRAY]
  [SGS
    <sgs(nodes)> -- READARRAY]
END GRIDDATA
```

```
BEGIN PACKAGEDATA
  <icsubno> <cellid(ncelldim)> <cdelay> <pcs0> <thick_frac> <rnb> <ssv_cc> <sse_cr> <theta> <kv> <h0> [<boundname>]
  <icsubno> <cellid(ncelldim)> <cdelay> <pcs0> <thick_frac> <rnb> <ssv_cc> <sse_cr> <theta> <kv> <h0> [<boundname>]
```

```
  ...
END PACKAGEDATA
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <cellid(ncelldim)> <sig0>
  <cellid(ncelldim)> <sig0>
  ...
END PERIOD
```

All of the stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered, all of the stresses from the previous block are replaced with the stresses in the new PERIOD block. Note that this behavior is different from the advanced packages (MAW, SFR, LAK, and UZF). To turn off all of the stresses for a stress period, a PERIOD block must be specified with no entries. If a PERIOD block is not specified for the first stress period, then no stresses will be applied until the `iper` value of the first PERIOD block in the file.

Explanation of Variables

Block: OPTIONS

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

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

`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.

`gammaw`—unit weight of water. For freshwater, GAMMAW is 9806.65 Newtons/cubic meters or 62.48 lb/cubic foot in SI and English units, respectively. By default, GAMMAW is 9806.65 Newtons/cubic meters.

`beta`—compressibility of water. Typical values of BETA are 4.6512e-10 1/Pa or 2.2270e-8 lb/square foot in SI and English units, respectively. By default, BETA is 4.6512e-10 1/Pa.

`HEAD_BASED`—keyword to indicate the head-based formulation will be used to simulate coarse-grained aquifer materials and no-delay and delay interbeds. Specifying HEAD_BASED also specifies the INITIAL_PRECONSOLIDATION_HEAD option.

`INITIAL_PRECONSOLIDATION_HEAD`—keyword to indicate that preconsolidation heads will be specified for no-delay and delay interbeds in the PACKAGEDATA block. If the SPECIFIED_INITIAL_INTERBED_STATE option is specified in the OPTIONS block, user-specified preconsolidation heads in the PACKAGEDATA block are absolute values. Otherwise, user-specified preconsolidation heads in the PACKAGEDATA block are relative to steady-state or initial heads.

`ndelaycells`—number of nodes used to discretize delay interbeds. If not specified, then a default value of 19 is assigned.

`COMPRESSION_INDICES`—keyword to indicate that the recompression (CR) and compression (CC) indices are specified instead of the elastic specific storage (SSE) and inelastic specific storage (SSV) coefficients. If not specified, then elastic specific storage (SSE) and inelastic specific storage (SSV) coefficients must be specified.

UPDATE_MATERIAL_PROPERTIES—keyword to indicate that the thickness and void ratio of coarse-grained and interbed sediments (delay and no-delay) will vary during the simulation. If not specified, the thickness and void ratio of coarse-grained and interbed sediments will not vary during the simulation.

CELL_FRACTION—keyword to indicate that the thickness of interbeds will be specified in terms of the fraction of cell thickness. If not specified, interbed thicknness must be specified.

SPECIFIED_INITIAL_INTERBED_STATE—keyword to indicate that absolute preconsolidation stresses (heads) and delay bed heads will be specified for interbeds defined in the PACKAGEDATA block. The SPECIFIED_INITIAL_INTERBED_STATE option is equivalent to specifying the SPECIFIED_INITIAL_PRECONSOLITATION_STRESS and SPECIFIED_INITIAL_DELAY_HEAD. If SPECIFIED_INITIAL_INTERBED_STATE is not specified then preconsolidation stress (head) and delay bed head values specified in the PACKAGEDATA block are relative to simulated values of the first stress period if steady-state or initial stresses and GWF heads if the first stress period is transient.

SPECIFIED_INITIAL_PRECONSOLIDATION_STRESS—keyword to indicate that absolute preconsolidation stresses (heads) will be specified for interbeds defined in the PACKAGEDATA block. If SPECIFIED_INITIAL_PRECONSOLITATION_STRESS and SPECIFIED_INITIAL_INTERBED_STATE are not specified then preconsolidation stress (head) values specified in the PACKAGEDATA block are relative to simulated values if the first stress period is steady-state or initial stresses (heads) if the first stress period is transient.

SPECIFIED_INITIAL_DELAY_HEAD—keyword to indicate that absolute initial delay bed head will be specified for interbeds defined in the PACKAGEDATA block. If SPECIFIED_INITIAL_DELAY_HEAD and SPECIFIED_INITIAL_INTERBED_STATE are not specified then delay bed head values specified in the PACKAGEDATA block are relative to simulated values if the first stress period is steady-state or initial GWF heads if the first stress period is transient.

EFFECTIVE_STRESS_LAG—keyword to indicate the effective stress from the previous time step will be used to calculate specific storage values. This option can 1) help with convergence in models with thin cells and water table elevations close to land surface; 2) is identical to the approach used in the SUBWT package for MODFLOW-2005; and 3) is only used if the effective-stress formulation is being used. By default, current effective stress values are used to calculate specific storage values.

STRAIN_CSV_INTERBED—keyword to specify the record that corresponds to final interbed strain output.

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

interbedstrain_filename—name of the comma-separated-values output file to write final interbed strain information.

STRAIN_CSV_COARSE—keyword to specify the record that corresponds to final coarse-grained material strain output.

coarsestrain_filename—name of the comma-separated-values output file to write final coarse-grained material strain information.

COMPACTION—keyword to specify that record corresponds to the compaction.

compaction_filename—name of the binary output file to write compaction information.

COMPACTION_ELASTIC—keyword to specify that record corresponds to the elastic interbed compaction binary file.

elastic_compaction_filename—name of the binary output file to write elastic interbed compaction information.

COMPACTION_INELASTIC—keyword to specify that record corresponds to the inelastic interbed compaction binary file.

inelastic_compaction_filename—name of the binary output file to write inelastic interbed compaction information.

COMPACTION_INTERBED—keyword to specify that record corresponds to the interbed compaction binary file.

interbed_compaction_filename—name of the binary output file to write interbed compaction information.

COMPACTION_COARSE—keyword to specify that record corresponds to the elastic coarse-grained material compaction binary file.

coarse_compaction_filename—name of the binary output file to write elastic coarse-grained material compaction information.

ZDISPLACEMENT—keyword to specify that record corresponds to the z-displacement binary file.

zdisplacement_filename—name of the binary output file to write z-displacement information.

PACKAGE_CONVERGENCE—keyword to specify that record corresponds to the package convergence comma spaced values file.

package_convergence_filename—name of the comma spaced values output file to write package convergence 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 CSUB package. See the "Observation utility" section for instructions for preparing observation input files. Tables 30 and 31 lists observation type(s) supported by the CSUB package.

Block: DIMENSIONS

ninterbeds—is the number of CSUB interbed systems. More than 1 CSUB interbed systems can be assigned to a GWF cell; however, only 1 GWF cell can be assigned to a single CSUB interbed system.

maxsig0—is the maximum number of cells that can have a specified stress offset. More than 1 stress offset can be assigned to a GWF cell. By default, MAXSIG0 is 0.

Block: GRIDDATA

cg_ske_cr—is the initial elastic coarse-grained material specific storage or recompression index. The recompression index is specified if COMPRESSION_INDICES is specified in the OPTIONS block. Specified or calculated elastic coarse-grained material specific storage values are not adjusted from initial values if HEAD_BASED is specified in the OPTIONS block.

cg_theta—is the initial porosity of coarse-grained materials.

sgm—is the specific gravity of moist or unsaturated sediments. If not specified, then a default value of 1.7 is assigned.

sgs—is the specific gravity of saturated sediments. If not specified, then a default value of 2.0 is assigned.

Block: PACKAGEDATA

icsubno—integer value that defines the CSUB interbed number associated with the specified PACKAGEDATA data on the line. CSUBNO must be greater than zero and less than or equal to NINTERBEDS. CSUB information must be specified for every CSUB cell or the program will terminate with an error. The program will also terminate with an error if information for a CSUB interbed number is specified more than once.

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, CELLID is the node number for the cell.

cdelay—character string that defines the subsidence delay type for the interbed. Possible subsidence package CDELAY strings include: NODELAY–character keyword to indicate that delay will not be simulated in the interbed. DELAY–character keyword to indicate that delay will be simulated in the interbed.

pcs0—is the initial offset from the calculated initial effective stress or initial preconsolidation stress in the interbed, in units of height of a column of water. PCS0 is the initial preconsolidation stress if SPECIFIED_INITIAL_INTERBED_STATE or SPECIFIED_INITIAL_PRECONSOLIDATION_STRESS are specified in the OPTIONS block. If HEAD_BASED is specified in the OPTIONS block, PCS0 is the initial offset from the calculated initial head or initial preconsolidation head in the CSUB interbed and the initial preconsolidation stress is calculated from the calculated initial effective stress or calculated initial geostatic stress, respectively.

thick_frac—is the interbed thickness or cell fraction of the interbed. Interbed thickness is specified as a fraction of the cell thickness if CELL_FRACTION is specified in the OPTIONS block.

rnb—is the interbed material factor equivalent number of interbeds in the interbed system represented by the interbed. RNB must be greater than or equal to 1 if CDELAY is DELAY. Otherwise, RNB can be any value.

ssv_cc—is the initial inelastic specific storage or compression index of the interbed. The compression index is specified if COMPRESSION_INDICES is specified in the OPTIONS block. Specified or calculated interbed inelastic specific storage values are not adjusted from initial values if HEAD_BASED is specified in the OPTIONS block.

sse_cr—is the initial elastic coarse-grained material specific storage or recompression index of the interbed. The recompression index is specified if COMPRESSION_INDICES is specified in the OPTIONS block. Specified or calculated interbed elastic specific storage values are not adjusted from initial values if HEAD_BASED is specified in the OPTIONS block.

theta—is the initial porosity of the interbed.

kv—is the vertical hydraulic conductivity of the delay interbed. KV must be greater than 0 if CDELAY is DELAY. Otherwise, KV can be any value.

h0—is the initial offset from the head in cell cellid or the initial head in the delay interbed. H0 is the initial head in the delay bed if SPECIFIED_INITIAL_INTERBED_STATE or SPECIFIED_INITIAL_DELAY_HEAD are specified in the OPTIONS block. H0 can be any value if CDELAY is NODELAY.

`boundname`—name of the CSUB 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.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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, CELLID is the node number for the cell.

`sig0`—is the stress offset for the cell. SIG0 is added to the calculated geostatic stress for the cell. SIG0 is specified only if MAXSIG0 is specified to be greater than 0 in the DIMENSIONS 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.

Example Input File

```
BEGIN OPTIONS
  COMPRESSION_INDICES
  SPECIFIED_INITIAL_INTERBED_STATE
  BOUNDNAMES
  SAVE_FLOWS
END OPTIONS

BEGIN DIMENSIONS
  NINTERBEDS 4
  MAXSIG0 1
END DIMENSIONS

BEGIN GRIDDATA
  # compression indices of coarse grained aquifer materials
  cg_ske_cr LAYERED
    CONSTANT      0.01
    CONSTANT      0.01
    CONSTANT      0.01
    CONSTANT      0.01
  # porosity of coarse grained aquifer materials
  cg_theta LAYERED
    CONSTANT      0.45
    CONSTANT      0.45
    CONSTANT      0.45
    CONSTANT      0.45
  # specific gravity of saturated sediment
  SGS LAYERED
      CONSTANT 2.0
      CONSTANT 2.0
      CONSTANT 2.0
      CONSTANT 2.0
  # specific gravity of moist sediment
```

```
  SGM LAYERED
      CONSTANT 1.7
      CONSTANT 1.7
      CONSTANT 1.7
      CONSTANT 1.7
END GRIDDATA

BEGIN PACKAGEDATA
# icsubsno cellid cdelay   pcs0  thick_frac rnb ssv_cc sse_cr theta  kv  h0 boundname
         1 1 1 6  delay   15.0      0.450 1.0  0.25   0.01   0.45 0.1 15. nsystm0
         2 1 1 7  nodelay 15.0      0.450 1.0  0.25   0.01   0.45 0.0 0.0 nsystm1
         3 1 1 8  nodelay 15.0      0.450 1.0  0.25   0.01   0.45 0.0 0.0 nsystm1
         4 1 1 9  delay   15.0      0.450 1.0  0.25   0.01   0.45 0.1 15. nsystm2
END PACKAGEDATA

BEGIN PERIOD 1
# stress offset for stress period 1
  1 1 6   1700.00000000
END PERIOD
```

Available observation types

Subsidence Package observations include all of the terms that contribute to the continuity equation for each GWF cell. The data required for each CSUB Package observation type is defined in table 7. Negative and positive values for CSUB observations represent a loss from and gain to the GWF model, respectively.

Table 7. Available CSUB Package observation types.

Stress Package	Observation type	ID	ID2	Description
CSUB	csub	icsubno or boundname	–	Flow between the groundwater system and a interbed or group of interbeds.
CSUB	inelastic-csub	icsubno or boundname	–	Flow between the groundwater system and a interbed or group of interbeds from inelastic compaction.
CSUB	elastic-csub	icsubno or boundname	–	Flow between the groundwater system and a interbed or group of interbeds from elastic compaction.
CSUB	coarse-csub	cellid	–	Flow between the groundwater system and coarse-grained materials in a GWF cell.
CSUB	csub-cell	cellid	–	Flow between the groundwater system for all interbeds and coarse-grained materials in a GWF cell.
CSUB	wcomp-csub-cell	cellid	–	Flow between the groundwater system for all interbeds and coarse-grained materials in a GWF cell from water compressibility.
CSUB	sk	icsubno or boundname	–	Convertible interbed storativity in a interbed or group of interbeds. Convertible interbed storativity is inelastic interbed storativity if the current effective stress is greater than the preconsolidation stress. The NODATA value is reported for steady-state stress periods.

Table 7. Available CSUB Package observation types.—Continued

Stress Package	Observation type	ID	ID2	Description
CSUB	ske	icsubno or boundname	–	Elastic interbed storativity in a interbed or group of interbeds. The NODATA value is reported for steady-state stress periods.
CSUB	sk-cell	cellid	–	Convertible interbed and coarse-grained material storativity in a GWF cell. Convertible interbed storativity is inelastic interbed storativity if the current effective stress is greater than the preconsolidation stress. The NODATA value is reported for steady-state stress periods.
CSUB	ske-cell	cellid	–	Elastic interbed and coarse-grained material storativity in a GWF cell. The NODATA value is reported for steady-state stress periods.
CSUB	estress-cell	cellid	–	effective stress in a GWF cell.
CSUB	gstress-cell	cellid	–	geostatic stress in a GWF cell.
CSUB	interbed-compaction	icsubno or boundname	–	interbed compaction in a interbed or group of interbeds.
CSUB	inelastic-compaction	icsubno or boundname	–	inelastic interbed compaction in a interbed or group of interbeds.
CSUB	elastic-compaction	icsubno or boundname	–	elastic interbed compaction a interbed or group of interbeds.
CSUB	coarse-compaction	cellid	–	elastic compaction in coarse-grained materials in a GWF cell.
CSUB	inelastic-compaction-cell	cellid	–	inelastic compaction in all interbeds in a GWF cell.
CSUB	elastic-compaction-cell	cellid	–	elastic compaction in coarse-grained materials and all interbeds in a GWF cell.
CSUB	compaction-cell	cellid	–	total compaction in coarse-grained materials and all interbeds in a GWF cell.
CSUB	thickness	icsubno or boundname	–	thickness of a interbed or group of interbeds.
CSUB	coarse-thickness	cellid	–	thickness of coarse-grained materials in a GWF cell.
CSUB	thickness-cell	cellid	–	total thickness of coarse-grained materials and all interbeds in a GWF cell.
CSUB	theta	icsubno	–	porosity of a interbed .
CSUB	coarse-theta	cellid	–	porosity of coarse-grained materials in a GWF cell.

Table 7. Available CSUB Package observation types.—Continued

Stress Package	Observation type	ID	ID2	Description
CSUB	theta-cell	cellid	–	thickness-weighted porosity of coarse-grained materials and all interbeds in a GWF cell.
CSUB	delay-flowtop	icsubno	–	Flow between the groundwater system and a delay interbed across the top of the interbed.
CSUB	delay-flowbot	icsubno	–	Flow between the groundwater system and a delay interbed across the bottom of the interbed.
CSUB	delay-head	icsubno	idcellno	head in interbed delay cell idcellno (1 <= idcellno <= NDELAYCELLS). The NODATA value is reported for steady-state stress periods.
CSUB	delay-gstress	icsubno	idcellno	geostatic stress in interbed delay cell idcellno (1 <= idcellno <= NDELAYCELLS). The NODATA value is reported for steady-state stress periods.
CSUB	delay-estress	icsubno	idcellno	effective stress in interbed delay cell idcellno (1 <= idcellno <= NDELAYCELLS). The NODATA value is reported for steady-state stress periods.
CSUB	delay-preconstress	icsubno	idcellno	preconsolidation stress in interbed delay cell idcellno (1 <= idcellno <= NDELAY-CELLS). The NODATA value is reported for steady-state stress periods.
CSUB	delay-compaction	icsubno	idcellno	compaction in interbed delay cell idcellno (1 <= idcellno <= NDELAYCELLS).
CSUB	delay-thickness	icsubno	idcellno	thickness of interbed delay cell idcellno (1 <= idcellno <= NDELAYCELLS).
CSUB	delay-theta	icsubno	idcellno	porosity of interbed delay cell idcellno (1 <= idcellno <= NDELAYCELLS).
CSUB	preconstress-cell	cellid	–	preconsolidation stress in a GWF cell containing at least one interbed. The NODATA value is reported for steady-state stress periods.

Example Observation Input File

```
BEGIN CONTINUOUS FILEOUT my_model.csub.csv
 tcomp3       compaction-cell      1 1 7
```

```
  ibcensystm0  elastic-compaction   nsystm0
  ibcinsystm0  inelastic-compaction nsystm0
END CONTINUOUS
```

Buoyancy (BUY) Package

Input to the Buoyancy (BUY) Package is read from the file that has type "BUY6" in the Name File. If the BUY Package is included for a model, then the model will use the variable-density form of Darcy's Law for all flow calculations using the approach described by Langevin and others (2020). Only one BUY Package can be specified for a GWF model. The BUY Package can be coupled with one or more GWT Models so that fluid density is updated dynamically with one or more simulated concentration fields.

The BUY Package calculates fluid density using the following equation of state from Langevin and others (2008):

$$\rho = DENSEREF + \sum_{i=1}^{NRHOSPECIES} DRHODC_i \left(CONCENTRATION_i - CRHOREF_i\right) \qquad (2)$$

where ρ is the calculated density, $DENSEREF$ is the density of a reference fluid, typically taken to be freshwater at a temperature of 25 degrees Celsius; $NRHOSPECIES$ is the number of chemical species that contribute to the density calculation, $DRHODC_i$ is the parameter that describes how density changes as a function of concentration for chemical species i (i.e. the slope of a line that relates density to concentration), $CONCENTRATION_i$ is the concentration of species i, and $CRHOREF_i$ is the concentration of species i in the reference fluid, which is normally set to zero.

Stress Packages

For head-dependent stress packages, the BUY Package may require fluid density and elevation for each head-dependent boundary so that the model can use a variable-density form of Darcy's Law to calculate flow between the boundary and the aquifer. By default, the boundary elevation is set equal to the cell elevation. For water-table conditions, the cell elevation is calculated as bottom elevation plus half of saturation multiplied by the cell thickness. If desired, the user can more precisely locate the boundary elevation by specifying an auxiliary variable with the name "ELEVATION". The program will use the values in this column as the boundary elevation. A situation where this may be required is for river or general-head boundaries that are conceptualized as being on top of a model cell. In those cases, an ELEVATION column should be specified and the values set to the top of the cell or some other appropriate elevation that corresponds to where the boundary stage applies.

By default, the boundary density is set equal to DENSEREF, commonly specified as the density of freshwater; however, there are two other options for setting the density of a boundary package. The first is to assign an auxiliary variable with the name "DENSITY". If this auxiliary variable is detected, then the density value in this column will be assigned to the density for the boundary. Alternatively, a density value can be calculated for each boundary using the density equation of state and one or more concentrations provided as auxiliary variables. In this case, the user must assign one auxiliary variable for each AUXSPECIESNAME listed in the PACKAGEDATA block below. Thus, there must be NRHOSPECIES auxiliary variables, each with the identical name as those specified in PACKAGEDATA. The BUY Package will calculate the density for each boundary using these concentrations and the values specified for DENSEREF, DRHODC, and CRHOREF. If the boundary package contains an auxiliary variable named DENSITY and also contains AUXSPECIESNAME auxiliary variables, then the boundary density value will be assigned to the one in the DENSITY auxiliary variable.

A GWT Model can be used to calculate concentrations for the advanced stress packages (LAK, SFR, MAW, and UZF) if corresponding advanced transport packages are specified (LKT, SFT, MWT, and UZT). The advanced stress packages have an input option called FLOW_PACKAGE_AUXILIARY_NAME. When

activated, this option will result in the simulated concentration for a lake or other feature being copied from the advanced transport package into the auxiliary variable for the corresponding GWF stress package. This means that the density for a lake or stream, for example, can be dynamically updated during the simulation using concentrations from advanced transport packages that are fed into auxiliary variables in the advanced stress packages, and ultimately used by the BUY Package to calculate a fluid density using the equation of state. This concept also applies when multiple GWT Models are used simultaneously to simulate multiple species. In this case, multiple auxiliary variables are required for an advanced stress package, with each one representing a concentration from a different GWT Model.

Table 8. Description of density terms for stress packages.

Stress Package	Note
GHB	ELEVATION can be specified as an auxiliary variable. A DENSITY auxiliary variable or one or more auxiliary variables for calculating density in the equation of state can be specified
RIV	ELEVATION can be specified as an auxiliary variable. A DENSITY auxiliary variable or one or more auxiliary variables for calculating density in the equation of state can be specified
DRN	The drain formulation assumes that the drain boundary contains water of the same density as the discharging water; auxiliary variables have no affect on the drain calculation
LAK	Elevation for each lake-aquifer connection is determined based on lake bottom and adjacent cell elevations. A DENSITY auxiliary variable or one or more auxiliary variables for calculating density in the equation of state can be specified
SFR	Elevation for each sfr-aquifer connection is determined based on stream bottom and adjacent cell elevations. A DENSITY auxiliary variable or one or more auxiliary variables for calculating density in the equation of state can be specified
MAW	Elevation for each maw-aquifer connection is determined based on cell elevation. A DENSITY auxiliary variable or one or more auxiliary variables for calculating density in the equation of state can be specified
UZF	No density terms implemented

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [HHFORMULATION_RHS]
  [DENSEREF <denseref>]
  [DENSITY FILEOUT <densityfile>]
END OPTIONS
```

```
BEGIN DIMENSIONS
  NRHOSPECIES <nrhospecies>
END DIMENSIONS
```

```
BEGIN PACKAGEDATA
  <irhospec> <drhodc> <crhoref> <modelname> <auxspeciesname>
  <irhospec> <drhodc> <crhoref> <modelname> <auxspeciesname>
  ...
END PACKAGEDATA
```

Explanation of Variables

Block: OPTIONS

`HHFORMULATION_RHS`—use the variable-density hydraulic head formulation and add off-diagonal terms to the right-hand. This option will prevent the BUY Package from adding asymmetric terms to the flow matrix.

`denseref`—fluid reference density used in the equation of state. This value is set to 1000. if not specified as an option.

`DENSITY`—keyword to specify that record corresponds to density.

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

`densityfile`—name of the binary output file to write density information. The density file has the same format as the head file. Density values will be written to the density file whenever heads are written to the binary head file. The settings for controlling head output are contained in the Output Control option.

Block: DIMENSIONS

`nrhospecies`—number of species used in density equation of state. This value must be one or greater. The value must be one if concentrations are specified using the CONCENTRATION keyword in the PERIOD block below.

Block: PACKAGEDATA

`irhospec`—integer value that defines the species number associated with the specified PACKAGEDATA data on the line. IRHOSPECIES must be greater than zero and less than or equal to NRHOSPECIES. Information must be specified for each of the NRHOSPECIES species or the program will terminate with an error. The program will also terminate with an error if information for a species is specified more than once.

`drhodc`—real value that defines the slope of the density-concentration line for this species used in the density equation of state.

`crhoref`—real value that defines the reference concentration value used for this species in the density equation of state.

`modelname`—name of GWT model used to simulate a species that will be used in the density equation of state. This name will have no affect if the simulation does not include a GWT model that corresponds to this GWF model.

`auxspeciesname`—name of an auxiliary variable in a GWF stress package that will be used for this species to calculate a density value. If a density value is needed by the Buoyancy Package then it will use the concentration values in this AUXSPECIESNAME column in the density equation of state. For advanced stress packages (LAK, SFR, MAW, and UZF) that have an associated advanced

transport package (LKT, SFT, MWT, and UZT), the FLOW_PACKAGE_AUXILIARY_NAME option in the advanced transport package can be used to transfer simulated concentrations into the flow package auxiliary variable. In this manner, the Buoyancy Package can calculate density values for lakes, streams, multi-aquifer wells, and unsaturated zone flow cells using simulated concentrations.

Example Input File

```
BEGIN OPTIONS
  DENSEREF 1000.
END OPTIONS

BEGIN DIMENSIONS
  NRHOSPECIES 2
END DIMENSIONS

BEGIN PACKAGEDATA
  #ISPEC DRHODC  CRHOREF MODELNAME AUXSPECIESNAME
      1    0.7      0.    GWT-1        SALINITY
      2  -0.375     25.    GWT-2     TEMPERATURE
END PACKAGEDATA
```

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.

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
```

All of the stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered, all of the stresses from the previous block are replaced with the stresses in the new PERIOD block. Note that this behavior is different from the advanced packages (MAW, SFR, LAK, and UZF). To turn off all of the stresses for a stress period, a PERIOD block must be specified with no entries. If a PERIOD block is not specified for the first stress period, then no stresses will be applied until the `iper` value of the first PERIOD block in the file.

Explanation of Variables

Block: OPTIONS

`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. Tables 30 and 31 lists observation type(s) supported by the constant-head package.

Block: DIMENSIONS

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

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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, CELLID is the node number for the cell.

head—is the head at 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 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 9. Negative and positive values for an observation represent a loss from and gain to the GWF model, respectively.

Table 9. 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 CONTINUOUS 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 CONTINUOUS

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.

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
```

All of the stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered, all of the stresses from the previous block are replaced with the stresses in the new PERIOD block. Note that this behavior is different from the advanced packages (MAW, SFR, LAK, and UZF). To turn off all of the stresses for a stress period, a PERIOD block must be specified with no entries. If a PERIOD block is not specified for the first stress period, then no stresses will be applied until the `iper` value of the first PERIOD block in the file.

Explanation of Variables

Block: OPTIONS

`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. Tables 30 and 31 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.

Block: DIMENSIONS

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

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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, 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_figure
       5   28   50 -12000   187.3         15.0  well_field
END PERIOD

#Use a list of wells in ASCII file wells_sp6.txt for stress period 6.
#Use these wells until the end of the simulation.
BEGIN PERIOD 6
  OPEN/CLOSE wells_sp6.txt
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 10. 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 10. 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.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [AUXILIARY <auxiliary(naux)>]
  [AUXMULTNAME <auxmultname>]
  [AUXDEPTHNAME <auxdepthname>]
  [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
```

All of the stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered, all of the stresses from the previous block are replaced with the stresses in the new PERIOD block. Note that this behavior is different from the advanced packages (MAW, SFR, LAK, and UZF). To turn off all of the stresses for a stress period, a PERIOD block must be specified with no entries. If a PERIOD block is not specified for the first stress period, then no stresses will be applied until the `iper` value of the first PERIOD block in the file.

Explanation of Variables

Block: OPTIONS

`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.

`auxdepthname`—name of a variable listed in AUXILIARY that defines the depth at which drainage discharge will be scaled. If a positive value is specified for the AUXDEPTHNAME AUXILIARY variable, then ELEV is the elevation at which the drain starts to discharge and ELEV + DDRN (assuming DDRN is the AUXDEPTHNAME variable) is the elevation when the drain conductance (COND) scaling factor is 1. If a negative drainage depth value is specified for DDRN, then ELEV + DDRN is the elevation at which the drain starts to discharge and ELEV is the elevation when the conductance (COND) scaling factor is 1. A linear- or cubic-scaling is used to scale the drain conductance (COND) when the Standard or Newton-Raphson Formulation is used, respectively.

`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. Tables 30 and 31 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.

Block: DIMENSIONS

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

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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, 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 11. 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 11. 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.

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
```

All of the stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered, all of the stresses from the previous block are replaced with the stresses in the new PERIOD block. Note that this behavior is different from the advanced packages (MAW, SFR, LAK, and UZF). To turn off all of the stresses for a stress period, a PERIOD block must be specified with no entries. If a PERIOD block is not specified for the first stress period, then no stresses will be applied until the `iper` value of the first PERIOD block in the file.

Explanation of Variables

Block: OPTIONS

`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. Tables 30 and 31 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.

Block: DIMENSIONS

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

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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, 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 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 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
  TS6 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 12. 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 12. 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.

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
```

All of the stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered, all of the stresses from the previous block are replaced with the stresses in the new PERIOD block. Note that this behavior is different from the advanced packages (MAW, SFR, LAK, and UZF). To turn off all of the stresses for a stress period, a PERIOD block must be specified with no entries. If a PERIOD block is not specified for the first stress period, then no stresses will be applied until the `iper` value of the first PERIOD block in the file.

Explanation of Variables

Block: OPTIONS

`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. Tables 30 and 31 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.

Block: DIMENSIONS

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

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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, 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)
  TS6 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 13. 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 13. 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.

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
```

All of the stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered, all of the stresses from the previous block are replaced with the stresses in the new PERIOD block. Note that this behavior is different from the advanced packages (MAW, SFR, LAK, and UZF). To turn off all of the stresses for a stress period, a PERIOD block must be specified with no entries. If a PERIOD block is not specified for the first stress period, then no stresses will be applied until the `iper` value of the first PERIOD block in the file.

Explanation of Variables

Block: OPTIONS

`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. Tables 30 and 31 lists observation type(s) supported by the Recharge package.

Block: DIMENSIONS

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

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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, 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. 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 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
  TS6 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 14. Negative and positive values for an observation represent a loss from and gain to the GWF model, respectively.

Table 14. 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.

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)> -- READARRAY]
  RECHARGE
    <recharge(ncol*nrow; ncpl)> -- READARRAY
  [AUX
    <aux(ncol*nrow; ncpl)> -- READARRAY]
END PERIOD
```

All of the stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered, all of the stresses from the previous block are replaced with the stresses in the new PERIOD block. Note that this behavior is different from the advanced packages (MAW, SFR, LAK, and UZF). To turn off all of the stresses for a stress period, a PERIOD block must be specified with no entries. If a PERIOD block is not specified for the first stress period, then no stresses will be applied until the `iper` value of the first PERIOD block in the file.

Explanation of Variables

Block: OPTIONS

`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. Tables 30 and 31 lists observation type(s) supported by the Recharge package.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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. If IRCH is specified, it must be specified as the first variable in the PERIOD block or MODFLOW will terminate with an error.

`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
  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 evapotranspiration is the default, and is described here. Instructions for specifying array-based evapotranspiration 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
```

All of the stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered, all of the stresses from the previous block are replaced with the stresses in the new PERIOD block. Note that this behavior is different from the advanced packages (MAW, SFR, LAK, and UZF). To turn off all of the stresses for a stress period, a PERIOD block must be specified with no entries. If a PERIOD block is not specified for the first stress period, then no stresses will be applied until the `iper` value of the first PERIOD block in the file.

Explanation of Variables

Block: OPTIONS

`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. Tables 30 and 31 lists observation type(s) supported by the Evapotranspiration package.

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

Block: DIMENSIONS

`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.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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, CELLID is the node number for the cell.

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

`rate`—is the maximum ET flux rate (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.

`depth`—is the ET extinction depth (L). 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.

`pxdp`—is the proportion of the ET extinction depth at the bottom of a segment (dimensionless). 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.

`petm`—is the proportion of the maximum ET flux rate at the bottom of a segment (dimensionless). 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.

`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. 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 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
  TS6 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 15. Negative and positive values for an observation represent a loss from and gain to the GWF model, respectively.

Table 15. 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)> -- READARRAY]
  SURFACE
    <surface(ncol*nrow; ncpl)> -- READARRAY
  RATE
    <rate(ncol*nrow; ncpl)> -- READARRAY
  DEPTH
    <depth(ncol*nrow; ncpl)> -- READARRAY
  AUX(IAUX)
    <aux(iaux)(ncol*nrow; ncpl)> -- READARRAY
END PERIOD
```

All of the stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered, all of the stresses from the previous block are replaced with the stresses in the new PERIOD block. Note that this behavior is different from the advanced packages (MAW, SFR, LAK, and UZF). To turn off all of the stresses for a stress period, a PERIOD block must be specified with no entries. If a PERIOD block is not specified for the first stress period, then no stresses will be applied until the `iper` value of the first PERIOD block in the file.

Explanation of Variables

Block: OPTIONS

`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. Tables 30 and 31 lists observation type(s) supported by the Evapotranspiration package.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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. If IEVT is specified, it must be specified as the first variable in the PERIOD block or MODFLOW will terminate with an error.

`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
  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.

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]
  [FLOW_CORRECTION]
  [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
```

All of the advanced stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered only the wells specified in the new period block will be changed. A well not specified in the new period block will continue to behave according to its specification in the previous PERIOD block. Note that this behavior is different from the simple stress packages (CHD, WEL, DRN, RIV, GHB, RCH and EVT), in which any stress not specified in a new PERIOD block will be removed. To turn off

all of the advanced stresses for a stress period, a PERIOD block must be specified with settings that deactivate the wells. If a PERIOD block is not specified for the first stress period, then no stresses will be applied.

Explanation of Variables

Block: OPTIONS

`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 head 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.

`FLOW_CORRECTION`—keyword that activates flow corrections in cases where the head in a multi-aquifer well is below the bottom of the screen for a connection or the head in a convertible cell connected to a multi-aquifer well is below the cell bottom. When flow corrections are activated, unit head gradients are used to calculate the flow between a multi-aquifer well and a connected GWF cell. By default, flow corrections are not made.

`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. Tables 30 and 31 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.

Block: DIMENSIONS

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

Block: PACKAGEDATA

`wellno`—integer value that defines the well number associated with the specified PACKAGEDATA data on the line. WELLNO must be greater than zero and less than or equal to NMAWWELLS. Multi-aquifer well information must be specified for every multi-aquifer well or the program will terminate with an error. The program will also terminate with an error if information for a multi-aquifer well is specified more than once.

`radius`—radius for the multi-aquifer well. The program will terminate with an error if the radius is less than or equal to zero.

`bottom`—bottom elevation of the multi-aquifer well. If CONDEQN is SPECIFIED, THIEM, SKIN, or COMPOSITE, BOTTOM is set to the cell bottom in the lowermost GWF cell connection in cases where the specified well bottom is above the bottom of this GWF cell. If CONDEQN is MEAN, BOTTOM is set to the lowermost GWF cell connection screen bottom in cases where the specified

well bottom is above this value. The bottom elevation defines the lowest well head that will be simulated when the NEWTON UNDER_RELAXATION option is specified in the GWF model name file. The bottom elevation is also used to calculate volumetric storage in the well.

`strt`—starting head for the multi-aquifer well. The program will terminate with an error if the starting head is less than the specified well bottom.

`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. THIEM–character keyword to indicate the multi-aquifer well saturated conductance will be calculated using the Thiem equation, which considers the cell top and bottom, aquifer hydraulic conductivity, and effective cell and well radius. SKIN–character keyword to indicate that the multi-aquifer well saturated conductance will be calculated using the cell top and bottom, aquifer and screen hydraulic conductivity, and well and skin radius. CUMULATIVE–character keyword to indicate that the multi-aquifer well saturated conductance will be calculated using a combination of the Thiem and SKIN equations. MEAN–character keyword to indicate the multi-aquifer well saturated conductance will be calculated using the aquifer and screen top and bottom, aquifer and screen hydraulic conductivity, and well and skin radius. The CUMULATIVE conductance equation is identical to the SKIN LOSSTYPE in the Multi-Node Well (MNW2) package for MODFLOW-2005. The program will terminate with an error condition if CONDEQN is SKIN or CUMULATIVE and the calculated saturated conductance is less than zero; if an error condition occurs, it is suggested that the THEIM or MEAN conductance equations be used for these multi-aquifer wells.

`ngwfnodes`—integer value that defines the number of GWF nodes connected to this (WELLNO) multi-aquifer well. 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.

Block: CONNECTIONDATA

`wellno`—integer value that defines the well number associated with the specified CONNECTIONDATA data on the line. WELLNO must be greater than zero and less than or equal to NMAWWELLS. Multi-aquifer well connection information must be specified for every multi-aquifer well connection to the GWF model (NGWFNODES) or the program will terminate with an error. The program will also terminate with an error if connection information for a multi-aquifer well connection to the GWF model is specified more than once.

`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, CELLID is the node number for the cell. One or more screened intervals can be connected to the same CELLID if CONDEQN for a well is MEAN. The program will terminate with an error if MAW wells using SPECIFIED, THIEM, SKIN, or CUMULATIVE conductance equations have more than one connection to the same CELLID.

`scrn_top`—value that defines the top elevation of the screen for the multi-aquifer well connection. If CONDEQN is SPECIFIED, THIEM, SKIN, or COMPOSITE, SCRN_TOP can be any value and is set to the top of the cell. If CONDEQN is MEAN, SCRN_TOP is set to the multi-aquifer well connection cell top if the specified value is greater than the cell top. The program will terminate with an error if the screen top is less than the screen bottom.

`scrn_bot`—value that defines the bottom elevation of the screen for the multi-aquifer well connection. If CONDEQN is SPECIFIED, THIEM, SKIN, or COMPOSITE, SCRN_BOT can be any value is set to the bottom of the cell. If CONDEQN is MEAN, SCRN_BOT is set to the multi-aquifer well connection cell bottom if the specified value is less than the cell bottom. The program will terminate with an error if the screen bottom is greater than the screen top.

`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). If CONDEQN is SPECIFIED, HK_SKIN must be greater than or equal to zero. HK_SKIN can be any value if CONDEQN is THIEM. Otherwise, HK_SKIN must be greater than zero. If CONDEQN is SKIN, the contrast between the cell transmissivity (the product of geometric mean horizontal hydraulic conductivity and the cell thickness) and the well transmissivity (the product of HK_SKIN and the screen thicknesses) must be greater than one in node CELLID or the program will terminate with an error condition; if an error condition occurs, it is suggested that the HK_SKIN be reduced to a value less than K11 and K22 in node CELLID or the THEIM or MEAN conductance equations be used for these multi-aquifer wells.

`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 THIEM. If CONDEQN is SKIN, CUMULATIVE, or MEAN, the program will terminate with an error if RADIUS_SKIN is less than or equal to the RADIUS for the multi-aquifer well.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD block.

`wellno`—integer value that defines the well number associated with the specified PERIOD data on the line. 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> <fwrlen>
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).

fwrlen—length used to reduce the conductance of the flowing well. When the head in the well drops below the well top plus the reduction length, then the conductance is reduced. This reduction length can be used to improve the stability of simulations with flowing wells so that there is not an abrupt change in flowing well rates.

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. The program will terminate with an error if WELL_HEAD is less than the bottom of the well.

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 can be applied to extraction wells (RATE $<$ 0) or injection 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 injection 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. RATE_SCALING can be used for extraction or injection wells. For extraction wells, the extraction rate will start to decrease once the head in the well lowers to a level equal to the pump elevation plus the scaling length. If the head in the well drops below the pump elevation, then the extraction rate is calculated to be zero. For an injection well, the injection rate will begin to decrease once the head in the well rises above the specified pump elevation. If the head in the well rises above the pump elevation plus the scaling length, then the injection rate will be set to zero.

`pump_elevation`—is the elevation of the multi-aquifer well pump (PUMP_ELEVATION). PUMP_ELEVATION should not be less than the bottom elevation (BOTTOM) of the multi-aquifer well.

`scaling_length`—height above the pump elevation (SCALING_LENGTH). If the simulated well head is below this elevation (pump elevation plus the scaling length), then the pumping rate is reduced.

`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
  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 thiem       2 pwell
       2   0.25 -100.0 9.14 thiem       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
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
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 16. 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 16. 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.
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.

Table 16. Available MAW Package observation types.—Continued

Stress Package	Observation type	ID	ID2	Description
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
  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.

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.

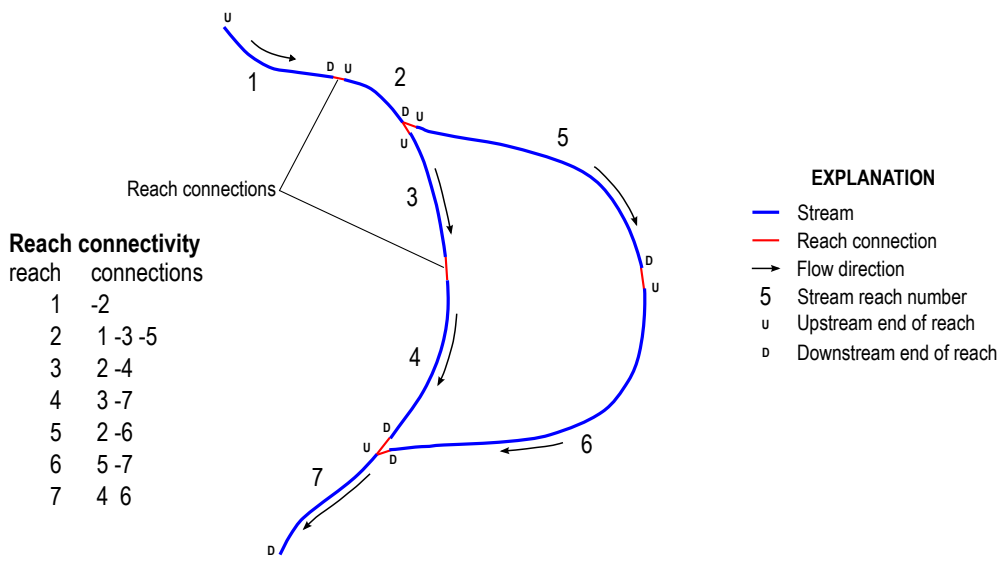

Figure 2. Simple stream network having seven reaches with a junction having two reaches, a confluence of two reaches, and the resulting reach connectivity. Downstream connections for a reach must include the reach as an upstream connection for all downstream connections to the reach. Downstream connections 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>]
  [PACKAGE_CONVERGENCE FILEOUT <package_convergence_filename>]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
```

```
  [MOVER]
  [MAXIMUM_PICARD_ITERATIONS <maximum_picard_iterations>]
  [MAXIMUM_ITERATIONS <maximum_iterations>]
  [MAXIMUM_DEPTH_CHANGE <maximum_depth_change>]
  [UNIT_CONVERSION <unit_conversion>]
END OPTIONS
```

```
BEGIN DIMENSIONS
  NREACHES <nreaches>
END DIMENSIONS
```

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

All of the advanced stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered only the reaches specified in the new period block will be changed. A reach not specified in the new period block will continue to behave according to its specification in the previous PERIOD block. Note that this behavior is different from the simple stress packages (CHD, WEL, DRN, RIV, GHB, RCH and EVT), in which any stress not specified in a new PERIOD block will be removed. To turn off all of the advanced stresses for a stress period, a PERIOD block must be specified with settings that deactivate the reaches. If a PERIOD block is not specified for the first stress period, then no stresses will be applied.

Explanation of Variables

Block: OPTIONS

`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.

PACKAGE_CONVERGENCE—keyword to specify that record corresponds to the package convergence comma spaced values file.

package_convergence_filename—name of the comma spaced values output file to write package convergence 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. Tables 30 and 31 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_picard_iterations—value that defines the maximum number of Streamflow Routing picard iterations allowed when solving for reach stages and flows as part of the GWF formulate step. Picard iterations are used to minimize differences in SFR package results between subsequent GWF picard (non-linear) iterations as a result of non-optimal reach numbering. If reaches are numbered in order, from upstream to downstream, MAXIMUM_PICARD_ITERATIONS can be set to 1 to reduce model run time. By default, MAXIMUM_PICARD_ITERATIONS is equal to 100.

`maximum_iterations`—value that defines the maximum number of Streamflow Routing Newton-Raphson iterations allowed for a reach. By default, MAXIMUM_ITERATIONS 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.

Block: DIMENSIONS

`nreaches`—integer value specifying the number of stream reaches. There must be NREACHES entries in the PACKAGEDATA block.

Block: PACKAGEDATA

`rno`—integer value that defines the reach number associated with the specified PACKAGEDATA data on the line. RNO must be greater than zero and less than or equal to NREACHES. Reach information must be specified for every reach or the program will terminate with an error. The program will also terminate with an error if information for a reach is specified more than once.

`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. If a value of zero is specified for NCON an entry for RNO is still required in the subsequent CONNECTIONDATA block.

`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. 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.

`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.

Block: CONNECTIONDATA

`rno`—integer value that defines the reach number associated with the specified CONNECTIONDATA data on the line. RNO must be greater than zero and less than or equal to NREACHES. Reach connection information must be specified for every reach or the program will terminate with an error. The program will also terminate with an error if connection information for a reach is specified more than once.

`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 NREACHES. IC should not be specified when NCON is zero but must be specified otherwise.

Block: DIVERSIONS

`rno`—integer value that defines the reach number associated with the specified DIVERSIONS data on the line. RNO must be greater than zero and less than or equal to NREACHES. Reach diversion information must be specified for every reach with a NDV value greater than 0 or the program will terminate with an error. The program will also terminate with an error if diversion information for a given reach diversion is specified more than once.

`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 NREACHES. 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 $\leq$ 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.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD block.

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

`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> <divflow>
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 fractions 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 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. 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. If the volumetric evaporation rate for a reach exceeds the sources of water to the reach (upstream and specified inflows, rainfall, and runoff but excluding groundwater leakage into the reach) the volumetric evaporation rate is limited to the sources of water to the reach. 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. If the volumetric runoff rate for a reach is negative and exceeds inflows to the reach (upstream and specified inflows, and rainfall but excluding groundwater leakage into the reach) the volumetric runoff rate is limited to inflows to the reach and the volumetric evaporation rate for the reach is set to zero. By default, runoff rates are zero for each reach.

`DIVERSION`—keyword to indicate diversion record.

`idv`—an integer value specifying which diversion of reach RNO that DIVFLOW is being specified for. Must be less or equal to ndv for the current reach (RNO).

`divflow`—real or character value that defines the volumetric diversion (DIVFLOW) 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.

`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
  NREACHES 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 17. 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 17. 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.
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.

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>]
  [PACKAGE_CONVERGENCE FILEOUT <package_convergence_filename>]
  [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>
  <number> <laksetting>
  <number> <laksetting>
  ...
END PERIOD
```

All of the advanced stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered only the lakes specified in the new period block will be changed. A lake not specified in the new period block will continue to behave according to its specification in the previous PERIOD block. Note that this behavior is different from the simple stress packages (CHD, WEL, DRN, RIV, GHB, RCH and EVT), in which any stress not specified in a new PERIOD block will be removed. To turn off all of the advanced stresses for a stress period, a PERIOD block must be specified with settings that deactivate the lakes. If a PERIOD block is not specified for the first stress period, then no stresses will be applied.

Explanation of Variables

Block: OPTIONS

`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.

PACKAGE_CONVERGENCE—keyword to specify that record corresponds to the package convergence comma spaced values file.

package_convergence_filename—name of the comma spaced values output file to write package convergence 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. Tables 30 and 31 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.

Block: DIMENSIONS

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.

Block: PACKAGEDATA

lakeno—integer value that defines the lake number associated with the specified PACKAGEDATA data on the line. LAKENO must be greater than zero and less than or equal to NLAKES. Lake information must be specified for every lake or the program will terminate with an error. The program will also terminate with an error if information for a lake is specified more than once.

strt—real value that defines the starting stage for the lake.

`nlakeconn`—integer value that defines the number of GWF cells connected to this (LAKENO) lake. There can only be one vertical lake connection to each GWF cell. 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.

Block: CONNECTIONDATA

`lakeno`—integer value that defines the lake number associated with the specified CONNECTIONDATA data on the line. LAKENO must be greater than zero and less than or equal to NLAKES. Lake connection information must be specified for every lake connection to the GWF model (NLAKECONN) or the program will terminate with an error. The program will also terminate with an error if connection information for a lake connection to the GWF model is specified more than once.

`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, 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 (NLAKECONN = 1) and there must be a lake table associated with each embedded lake.

`bedleak`—character string or real value that defines the bed leakance for the lake-GWF connection. BEDLEAK must be greater than or equal to zero or specified to be NONE. If BEDLEAK is specified to be NONE, the lake-GWF connection conductance is solely a function of aquifer properties in the connected GWF cell and lakebed sediments are assumed to be absent.

`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. CONLENN 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.

Block: TABLES

`lakeno`—integer value that defines the lake number associated with the specified TABLES data on the line. LAKENO must be greater than zero and less than or equal to NLAKES. The program will terminate with an error if table information for a lake is specified more than once or the number of specified tables is less than NTABLES.

`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 TAB6_FILENAME file includes the number of entries in the file and the relation between stage, volume, and surface area 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. Instructions for creating the TAB6_FILENAME input file are provided in Lake Table Input File section.

Block: OUTLETS

`outletno`—integer value that defines the outlet number associated with the specified OUTLETS data on the line. OUTLETNO must be greater than zero and less than or equal to NOUTLETS. Outlet information must be specified for every outlet or the program will terminate with an error. The program will also terminate with an error if information for a outlet is specified more than once.

`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.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD block.

`number`—integer value that defines the lake or outlet number associated with the specified PERIOD data on the line. NUMBER must be greater than zero and less than or equal to NLAKES for a lake number and less than or equal to NOUTLETS for an outlet number.

`laksetting`—line of information that is parsed into a keyword and values. Keyword values that can be used to start the LAKSETTING string include both keywords for lake settings and keywords for outlet settings. Keywords for lake settings include: STATUS, STAGE, RAINFALL, EVAPORATION, RUNOFF, INFLOW, WITHDRAWAL, and AUXILIARY. Keywords for outlet settings include RATE, INVERT, WIDTH, SLOPE, and ROUGH.

```
STATUS <status>
STAGE <stage>
RAINFALL <rainfall>
EVAPORATION <evaporation>
RUNOFF <runoff>
INFLOW <inflow>
WITHDRAWAL <withdrawal>
RATE <rate>
INVERT <invert>
WIDTH <width>
SLOPE <slope>
ROUGH <rough>
AUXILIARY <auxname> <auxval>
```

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

`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 (LT^{-1}) 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 (LT^{-1}) 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 (L^3T^{-1}) 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.

`inflow`—real or character value that defines the volumetric inflow rate (L^3T^{-1}) 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. By default, inflow rates are zero for each lake.

`withdrawal`—real or character value that defines the maximum withdrawal rate (L^3T^{-1}) 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.

`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.

`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 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.

`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.

`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
  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    ctype bedleak belev telev dx width
      1     1     1   7      6  HORIZONTAL    0.1     0     0 500   500
      1     2     1   8      6  HORIZONTAL    0.1     0     0 500   500
      1     3     1   9      6  HORIZONTAL    0.1     0     0 500   500
      1     4     1  10      6  HORIZONTAL    0.1     0     0 500   500
      1     5     1  11      6  HORIZONTAL    0.1     0     0 500   500
      1     6     1   6      7  HORIZONTAL    0.1     0     0 500   500
      1     7     2   7      7    VERTICAL    0.1     0     0   0     0
      1     8     2   8      7    VERTICAL    0.1     0     0   0     0
      1     9     2   8      7  HORIZONTAL    0.1     0     0 250   500
      1    10     2   9      7    VERTICAL    0.1     0     0   0     0
      1    11     2   9      7  HORIZONTAL    0.1     0     0 250   500
      1    12     2  10      7    VERTICAL    0.1     0     0   0     0
      1    13     2  10      7  HORIZONTAL    0.1     0     0 250   500
      1    14     2  11      7    VERTICAL    0.1     0     0   0     0
      1    15     1  12      7  HORIZONTAL    0.1     0     0 500   500
      1    16     1   6      8  HORIZONTAL    0.1     0     0 500   500
      1    17     2   7      8    VERTICAL    0.1     0     0   0     0
      1    18     2   7      8  HORIZONTAL    0.1     0     0 250   500
      1    19     3   8      8    VERTICAL    0.1     0     0   0     0
      1    20     3   9      8    VERTICAL    0.1     0     0   0     0
      1    21     3  10      8    VERTICAL    0.1     0     0   0     0
      1    22     2  11      8    VERTICAL    0.1     0     0   0     0
      1    23     2  11      8  HORIZONTAL    0.1     0     0 250   500
      1    24     1  12      8  HORIZONTAL    0.1     0     0 500   500
      1    25     1   6      9  HORIZONTAL    0.1     0     0 500   500
      1    26     2   7      9    VERTICAL    0.1     0     0   0     0
```

```
      1    27     2   7      9  HORIZONTAL    0.1     0     0 250   500
      1    28     3   8      9    VERTICAL    0.1     0     0   0     0
      1    29     3   9      9    VERTICAL    0.1     0     0   0     0
      1    30     3  10      9    VERTICAL    0.1     0     0   0     0
      1    31     2  11      9    VERTICAL    0.1     0     0   0     0
      1    32     2  11      9  HORIZONTAL    0.1     0     0 250   500
      1    33     1  12      9  HORIZONTAL    0.1     0     0 500   500
      1    34     1   6     10  HORIZONTAL    0.1     0     0 500   500
      1    35     2   7     10    VERTICAL    0.1     0     0   0     0
      1    36     2   7     10  HORIZONTAL    0.1     0     0 250   500
      1    37     3   8     10    VERTICAL    0.1     0     0   0     0
      1    38     3   9     10    VERTICAL    0.1     0     0   0     0
      1    39     3  10     10    VERTICAL    0.1     0     0   0     0
      1    40     2  11     10    VERTICAL    0.1     0     0   0     0
      1    41     2  11     10  HORIZONTAL    0.1     0     0 250   500
      1    42     1  12     10  HORIZONTAL    0.1     0     0 500   500
      1    43     1   6     11  HORIZONTAL    0.1     0     0 500   500
      1    44     2   7     11    VERTICAL    0.1     0     0   0     0
      1    45     2   8     11    VERTICAL    0.1     0     0   0     0
      1    46     2   8     11  HORIZONTAL    0.1     0     0 250   500
      1    47     2   9     11    VERTICAL    0.1     0     0   0     0
      1    48     2   9     11  HORIZONTAL    0.1     0     0 250   500
      1    49     2  10     11    VERTICAL    0.1     0     0   0     0
      1    50     2  10     11  HORIZONTAL    0.1     0     0 250   500
      1    51     2  11     11    VERTICAL    0.1     0     0   0     0
      1    52     1  12     11  HORIZONTAL    0.1     0     0 500   500
      1    53     1   7     12  HORIZONTAL    0.1     0     0 500   500
      1    54     1   8     12  HORIZONTAL    0.1     0     0 500   500
      1    55     1   9     12  HORIZONTAL    0.1     0     0 500   500
      1    56     1  10     12  HORIZONTAL    0.1     0     0 500   500
      1    57     1  11     12  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
END PERIOD

BEGIN PERIOD 100
  1 STATUS CONSTANT
  1 STAGE 110.
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 18. 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 18. Available LAK Package observation types.

Stress Package	Observation type	ID	ID2	Description
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.
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.

Table 18. Available LAK Package observation types.—Continued

Stress Package	Observation type	ID	ID2	Description
LAK	constant	lakeno or boundname	–	Simulated constant-flow rate for a lake or group of lakes.
LAK	outlet	outletno or boundname	–	Simulated 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
  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.

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

Block: DIMENSIONS

`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 columns in the lake table. There must be NCOL columns of data in the TABLE block. For lakes with HORIZONTAL and/or VERTICAL CTYPE connections, NCOL must be equal to 3. For lakes with EMBEDDEDH or EMBEDDEDV CTYPE connections, NCOL must be equal to 4.

Block: TABLE

`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.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [AUXILIARY <auxiliary(naux)>]
  [AUXMULTNAME <auxmultname>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [BUDGET FILEOUT <budgetfile>]
  [PACKAGE_CONVERGENCE FILEOUT <package_convergence_filename>]
  [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
  <iuzno> <cellid(ncelldim)> <landflag> <ivertcon> <surfdep> <vks> <thtr> <thts> <thti> <eps> [<boundname>]
  <iuzno> <cellid(ncelldim)> <landflag> <ivertcon> <surfdep> <vks> <thtr> <thts> <thti> <eps> [<boundname>]
  ...
END PACKAGEDATA
```

FOR ANY STRESS PERIOD

```
BEGIN PERIOD <iper>
  <iuzno> <finf> <pet> <extdp> <extwc> <ha> <hroot> <rootact> [<aux(naux)>]
  <iuzno> <finf> <pet> <extdp> <extwc> <ha> <hroot> <rootact> [<aux(naux)>]
  ...
END PERIOD
```

All of the advanced stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered only the UZF cells specified in the new period block will be changed. A UZF cell not specified in the new period block will continue to behave according to its specification in the previous PERIOD block. Note that this behavior is different from the simple stress packages (CHD, WEL, DRN, RIV, GHB, RCH and EVT), in which any stress not specified in a new PERIOD block will be removed. To turn off all of the advanced stresses for a stress period, a PERIOD block must be specified with settings that deactivate the UZF cells. If a PERIOD block is not specified for the first stress period, then no stresses will be applied.

Explanation of Variables

Block: OPTIONS

`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.

`PACKAGE_CONVERGENCE`—keyword to specify that record corresponds to the package convergence comma spaced values file.

`package_convergence_filename`—name of the comma spaced values output file to write package convergence 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. Tables 30 and 31 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 omitting 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 using a capillary pressure based formulation. Capillary pressure is calculated using the Brooks-Corey retention function.

Block: DIMENSIONS

`nuzfcells`—is the number of UZF cells. More than one UZF cell can be assigned to a GWF cell; however, only one GWF cell can be assigned to a single UZF cell. If more than one UZF cell is assigned to a GWF cell, then an auxiliary variable should be used to reduce the surface area of the UZF cell with the AUXMULTNAME option.

`ntrailwaves`—is the number of trailing waves. A recommended value of 7 can be used for NTRAILWAVES. This value can be increased to lower mass balance error in the unsaturated zone.

`nwavesets`—is the number of wave sets. A recommended value of 40 can be used for NWAVESETS. This value can be increased if more waves are required to resolve variations in water content within the unsaturated zone.

Block: PACKAGEDATA

`iuzno`—integer value that defines the UZF cell number associated with the specified PACKAGEDATA data on the line. IUZNO must be greater than zero and less than or equal to NUZFCELLS. UZF information must be specified for every UZF cell or the program will terminate with an error. The program will also terminate with an error if information for a UZF cell is specified more than once.

`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, 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 the water table before the cell bottom, then water is added to the GWF cell instead of flowing to the 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.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD block.

`iuzno`—integer value that defines the UZF cell number associated with the specified PERIOD data on the line.

`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 1.0 1.0 0.05 0.35 0.1 4.0
  2  1 1 2 1 1.0 1.0 0.05 0.35 0.1 4.0
  3  1 1 3 1 1.0 1.0 0.05 0.35 0.1 4.0
  4  1 1 4 1 1.0 1.0 0.05 0.35 0.1 4.0
  5  1 1 5 1 1.0 1.0 0.05 0.35 0.1 4.0
  6  1 1 6 1 1.0 1.0 0.05 0.35 0.1 4.0
  7  1 1 7 1 1.0 1.0 0.05 0.35 0.1 4.0
  8  1 1 8 1 1.0 1.0 0.05 0.35 0.1 4.0
  9  1 1 9 1 1.0 1.0 0.05 0.35 0.1 4.0
  10 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 19. 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 19. Available UZF Package observation types.

Stress Package	Observation type	ID	ID2	Description
UZF	uzf-gwrch	iuzno 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	iuzno 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	iuzno 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	iuzno or boundname	–	Simulated groundwater evapotranspiration calculated by the UZF package for a UZF cell or a group of UZF cells.
UZF	infiltration	iuzno 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	iuzno or boundname	–	Inflow into a UZF cell from the MVR package for a UZF cell or a group of UZF cells.
UZF	rej-inf	iuzno or boundname	–	Simulated rejected infiltration calculated by the UZF package for a UZF cell or a group of UZF cells.

Table 19. Available UZF Package observation types.—Continued

Stress Package	Observation type	ID	ID2	Description
UZF	rej-inf-to-mvr	iuzno 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	iuzno or boundname	–	Simulated unsaturated evapotranspiration calculated by the UZF package for a UZF cell or a group of UZF cells.
UZF	storage	iuzno or boundname	–	Simulated storage flow rate for a UZF package cell or a group of UZF cells.
UZF	net-infiltration	iuzno or boundname	–	Simulated net infiltration rate for a UZF package cell or a group of UZF cells.
UZF	water-content	iuzno 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.

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
```

All of the mover information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered, all of the movers from the previous block are replaced with the movers in the new PERIOD block. Note that this behavior is different from the other advanced packages (MAW, SFR, LAK, and UZF). To turn off all of the movers for a stress period, a PERIOD block must be specified with no entries. If a PERIOD block is not specified for the first stress period, then no movers will be applied until the `iper` value of the first PERIOD block in the file.

Explanation of Variables

Block: OPTIONS

`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.

Block: DIMENSIONS

`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.

Block: PACKAGES

`mname`—name of model containing the package. Model names are assigned by the user in the simulation name file.

`pname`—is the name of a package that may be included in a subsequent stress period block. The package name is assigned in the name file for the GWF Model. Package names are optionally provided in the name file. If they are not provided by the user, then packages are assigned a default value, which is the package acronym followed by a hyphen and the package number. For example, the first Drain Package is named DRN-1. The second Drain Package is named DRN-2, and so forth.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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. For the standard boundary packages, the provider identifier is the number of the boundary as it is listed in the package input file. (Note that the order of these boundaries may change by stress period, which must be accounted for in the Mover Package.) So the first well has an identifier of one. The second is two, and so forth. For the advanced packages, the identifier is the reach number (SFR Package), well number (MAW Package), or UZF cell number. For the Lake Package, ID1 is the lake outlet number. Thus, outflows from a single lake can be routed to different streams, for example.

`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. The receiver identifier is the reach number (SFR Package), Lake number (LAK Package), well number (MAW Package), or UZF cell number.

`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 mirror the options defined for the cprior variable in the SFR package, with the term "FACTOR" being functionally equivalent to the "FRACTION" option for cprior.

`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.

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

Block: OPTIONS

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.

Block: DIMENSIONS

`numgnc`—is the number of GNC entries.

`numalphaj`—is the number of contributing factors.

Block: GNCDATA

`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. The sum of ALPHASJ should be less than one. This is because one minus the sum of ALPHASJ is equal to the alpha term (alpha n in equation 4-61 of the GWF Model report) that is multiplied by the head in cell n.

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.

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

Block: OPTIONS

`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, where a value of zero indicates that a normal vector points directly along the positive x axis. The connection face normal is a normal vector on the cell face shared between the cell in model 1 and the cell in model 2 pointing away from the model 1 cell. Additional information on "ANGLDEGX" is provided in the description of the DISU Package. If an auxiliary variable with the name "CDIST" is found, then this information will be used as the straight-line connection distance, including the vertical component, between the two cell centers. Both ANGLDEGX and CDIST are required if specific discharge is calculated for either of the groundwater models.

`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 package name 1 and package name 2 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 30 lists observation type(s) supported by the GWF-GWF package.

Block: DIMENSIONS

`nexg`—keyword and integer value specifying the number of GWF-GWF exchanges.

Block: EXCHANGEDATA

`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 1 and the its shared face with cell 2.

`cl2`—is the distance between the center of cell 2 and the its shared face with cell 1.

`hwva`—is the horizontal width of the flow connection between cell 1 and cell 2 if IHC $>$ 0, or it is the area perpendicular to flow of the vertical connection between cell 1 and cell 2 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 20. For `flow-ja-face` observation types, negative and positive values represent a loss from and gain to the `cellid` specified for ID, respectively.

Table 20. 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
  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
```

Groundwater Transport (GWT) Model Input

The GWT Model simulates three-dimensional transport of a single solute species in flowing groundwater. The GWT Model solves the solute transport equation using numerical methods and a generalized control-volume finite-difference approach, which can be used with regular MODFLOW grids (DIS Package) or with unstructured grids (DISV and DISU Packages). The GWT Model is designed to work with most of the new capabilities released with the GWF Model, including the Newton flow formulation, unstructured grids, advanced packages, and the movement of water between packages. The GWF and GWT Models operate simultaneously during a MODFLOW 6 simulation to represent coupled groundwater flow and solute transport. The GWT Model can also run separately from a GWF Model by reading the heads and flows saved by a previously run GWF Model. The GWT model is also capable of working with the flows from another groundwater flow model, as long as the flows from that model can be written in the correct form to flow and head files.

The purpose of the GWT Model is to calculate changes in solute concentration in both space and time. Solute concentrations within an aquifer can change in response to multiple solute transport processes. These processes include (1) advective transport of solute with flowing groundwater, (2) the combined hydrodynamic dispersion processes of velocity-dependent mechanical dispersion and chemical diffusion, (3) sorbtion of solutes by the aquifer matrix either by adsorption to individual solid grains or by absorbtion into solid grains, (4) transfer of solute into very low permeability aquifer material (called an immobile domain) where it can be stored and later released, (5) first- or zero-order solute decay or production in response to chemical or biological reactions, (6) mixing with fluids from groundwater sources and sinks, and (7) direct addition of solute mass.

With the present implementation, there can be multiple domains and multiple phases. There is a single mobile domain, which normally consists of flowing groundwater, and there can be one or more immobile domains. The GWT Model simulates the dissolved phase of chemical constituents in both the mobile and immobile domains. The dissolved phase is also referred to in this report as the aqueous phase. If sorbtion is represented, then the GWT Model also simulates the solid phase of the chemical constituent in both the mobile and immobile domains. The dissolved and solid phases of the chemical constituent are tracked in the different domains by the GWT Model and can be reported as output as requested by the user.

This section describes the data files for a MODFLOW 6 Groundwater Transport (GWT) Model. A GWT Model is added to the simulation by including a GWT entry in the MODELS block of the simulation name file. There are three types of spatial discretization approaches that can be used with the GWT Model: DIS, DISV, and DISU. The input instructions for these three packages are not described here in this section on GWT Model input; input instructions for these three packages are described in the section on GWF Model input.

The GWT 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 GWT Model Listing File is a key file to which the GWT model output is written. As MODFLOW 6 runs, information about the GWT Model is written to the GWT 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 GWT Model Instructions.

The GWT 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.

For the GWT Model, “flows” (unless stated otherwise) represent solute mass “flow” in mass per time, rather than groundwater flow.

Information for Existing Solute Transport Modelers

The MODFLOW 6 GWT Model contains most of the functionality of MODFLOW-GWT, MT3DMS, MT3D-USGS and MODFLOW-USG. The following list summarizes major differences between the GWT Model in MODFLOW 6 and previous MODFLOW-based solute transport programs.

1. The GWT Model simulates transport of a single chemical species; however, because MODFLOW 6 allows for multiple models of the same type to be included in a single simulation, multiple species can be represented by using multiple GWT Models.

2. There is no specialized flow and transport link file (Zheng and others, 2001) used to pass the simulated groundwater flows to the transport model. Instead, simulated flows from the GWF Model are passed in memory to the GWT Model while the program is running. Alternatively, the GWT Model can read binary flow and head files saved by the GWF Model while it is running. If the user intends to simulate transport through the advanced stress packages and Water Mover Package, then flows from these advanced packages must also be saved to binary files. Names for these binary files are provided as input to the FMI Package.

3. The GWT Model is based on a generalized control-volume finite-difference method, which means that solute transport can be simulated using regular MODFLOW grids consisting of layers, rows, and columns, or solute transport can be simulated using unstructured grids.

4. Advection can be simulated using central-in-space weighting, upstream weighting, or an implicit second-order TVD scheme. The GWT model does not have the Method of Characteristics (particle-based approaches) or an explicit TVD scheme. Consequently, the GWT Model may require a higher level of spatial discretization than other transport models that use higher order terms for advection dominated systems. This can be an important limitation for some problems, which require the preservation of sharp solute fronts.

5. Variable-density flow and transport can be simulated by including a GWF Model and a GWT Model in the same MODFLOW 6 simulation. The Buoyancy Package should be activated for the GWF Model so that fluid density is calculated as a function of simulated concentration. If more than one chemical species is represented then the Buoyancy Package allows the simulated concentration for each of them to be used in the density equation of state. Langevin and others (2020) describe the hydraulic-head formation that is implemented in the Buoyancy Package for variable-density groundwater flow and present the results from MODFLOW 6 variable-density simulations. The variable-density capabilities available in MODFLOW 6 replicate and extend the capabilities available in SEAWAT to include the Newton flow formulation and unstructured grids, for example.

6. The GWT model includes the MST and IST Packages. These two package collectively comprise the capabilities of the MT3DMS Reactions Package.

7. The MST Package contains the linear, Freundlich, and Langmuir isotherms for representing sorption. The IST Packages contains only the linear isotherm for representation of sorption.

8. The GWT model was designed so that the user can specify as many immobile domains and necessary to represent observed contaminant transport patterns and solute breakthrough curves. The effects of an immobile domain are represented using the Immobile Storage and Transfer (IST) Package, and the user can specify as many IST Packages as necessary.

9. Although there is GWF-GWF Exchange, a GWT-GWT Exchange has not yet been developed to connect multiple transport models, as might be done in a nested grid configuration.

10. There is no option to automatically run the GWT Model to steady state using a single time step. This is an option available in MT3DMS (Zheng, 2010). Steady state conditions must be determined by running the transport model under transient conditions until concentrations stabilize.

11. The GWT Model described in this report is capable of simulating solute transport in the advanced stress packages of MODFLOW 6, including the Lake, Streamflow Routing, Multi-Aquifer Well and Unsaturated Zone Transport Packages. The present implementation simulates solute advection between package features, such as between two stream reaches, but dispersive transport is not represented. Likewise, solute transport between the advanced packages and the aquifer occurs only through advection.

12. The GWT Model has not yet been programmed to work with the Skeletal Storage, Compaction, and Subsidence (CSUB) Package for the GWF Model.

13. There are many other differences between the MODFLOW 6 GWT Model and other solute transport models that work with MODFLOW, especially with regards to program design and input and output. Descriptions for the GWT input and output are described here.

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.

Solute Mass Budget

A summary of all inflow (sources) and outflow (sinks) of solute mass is called a mass budget. MODFLOW 6 calculates a mass 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 mass to the flow system. The solute mass budget is printed to the GWT Model Listing File for selected time steps.

Time Stepping

For the present implementation of the GWT Model, all terms in the solute transport equation are solved implicitly. With the implicit approach applied to the transport equation, it is possible to take relatively large time steps and efficiently obtain a stable solution. If the time steps are too large, however, accuracy of the model results will suffer, so there is usually some compromise required between the desired level of accuracy and length of the time step. An assessment of accuracy can be performed by simply running simulations with shorter time steps and comparing results.

In MODFLOW 6 time step lengths are controlled by the user and specified in the Temporal Discretization (TDIS) input file. When the flow model and transport model are included in the same simulation, then the length of the time step specified in TDIS is used for both models. If the GWT Model runs in a separate simulation from the GWT Model, then the time steps used for the transport model can be different, and likely shorter, than the time steps used for the flow solution. Instructions for specifying time steps are described in the TDIS section of this user guide; additional information on GWF and GWT configurations are in the Flow Model Interface section.

GWT Model Name File

The GWT Model Name File specifies the options and packages that are active for a GWT 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]
END OPTIONS
```

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

Explanation of Variables

Block: OPTIONS

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

Block: PACKAGES

`ftype`—is the file type, which must be one of the following character values shown in table 21. 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 GWT 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 21. Ftype values described in this report. The `Pname` column indicates whether or not a package name can be provided in the name file. The capability to provide a package name also indicates that the GWT Model can have more than one package of that Ftype.

Ftype	Input File Description	`Pname`
DIS6	Rectilinear Discretization Input File	
DISV6	Discretization by Vertices Input File	
DISU6	Unstructured Discretization Input File	
FMI6	Flow Model Interface Package	
IC6	Initial Conditions Package	
OC6	Output Control Option	
ADV6	Advection Package	
DSP6	Dispersion Package	
SSM6	Source and Sink Mixing Package	
MST6	Mobile Storage and Transfer Package	
IST6	Immobile Storage and Transfer Package	*
CNC6	Constant Concentration Package	*
SRC6	Mass Source Loading Package	*
LKT6	Lake Transport Package	*
SFT6	Streamflow Transport Package	*
MWT6	Multi-Aquifer Well Transport Package	*
UZT6	Unsaturated Zone Transport Package	*
MVT6	Mover Transport Package	
OBS6	Observations Option	

Example Input File

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

BEGIN PACKAGES
  DIS6      transport.dis
  IC6       transport.ic
  MST6      transport.mst
  ADV6      transport.adv
  DSP6      transport.dsp
  SSM6      transport.ssm
  CNC6      transport01.cnc LEFT
  CNC6      transport02.cnc RIGHT
```

```
  SRC6      transport01.src LAY1
  SRC6      transport02.src LAY2
  SRC6      transport03.src LAY3
  IST6      transport01.ist CLAY
  IST6      transport02.ist SILT
  OC6       transport.oc
END PACKAGES
```

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 GWT model.

Structure of Blocks

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

Explanation of Variables

Block: GRIDDATA

`strt`—is the initial (starting) concentration—that is, concentration at the beginning of the GWT Model simulation. STRT must be specified for all GWT Model simulations. One value is read for every model cell.

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 Concentration layer 1
    CONSTANT  0.0  Initial Concentration layer 2
END GRIDDATA
```

Output Control (OC) Option

Input to the Output Control Option of the Groundwater Transport 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 concentrations are printed to the listing file and/or written to a separate binary output file. Under the default, concentration and overall transport budget are written to the Listing File at the end of every stress period. The default printout format for concentrations is 10G11.4. The concentrations and overall transport budget are also written to the list file if the simulation terminates prematurely due to failed convergence.

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

For the PRINT and SAVE options of concentration, there is no option to specify individual layers. Whenever the concentration array is printed or saved, all layers are printed or saved.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [BUDGET FILEOUT <budgetfile>]
  [CONCENTRATION FILEOUT <concentrationfile>]
  [CONCENTRATION 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

Block: OPTIONS

`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.

`CONCENTRATION`—keyword to specify that record corresponds to concentration.

`concentrationfile`—name of the output file to write conc 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.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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 CONCENTRATION.

`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
  CONCENTRATION FILEOUT transport.ucn
  CONCENTRATION PRINT_FORMAT COLUMNS 15 WIDTH 7 DIGITS 2 FIXED
END OPTIONS

BEGIN PERIOD 1
  PRINT BUDGET ALL
  SAVE CONCENTRATION ALL
  PRINT CONCENTRATION ALL
END PERIOD
```

Observation (OBS) Utility for a GWT Model

GWT Model observations include the simulated groundwater concentration (`concentration`), and the mass flow, with units of mass per time, between two connected cells (`flow-ja-face`). The data required for each GWT Model observation type is defined in table 22. For `flow-ja-face` observation types, negative and positive values represent a loss from and gain to the `cellid` specified for ID, respectively.

Structure of Blocks

FOR EACH SIMULATION

```
BEGIN OPTIONS
  [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

Block: OPTIONS

`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.

Block: CONTINUOUS

`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 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.

Table 22. Available GWT model observation types.

Model	Observation type	ID	ID2	Description
GWT	concentration	cellid	–	Concentration at a specified cell.
GWT	flow-ja-face	cellid	cellid	Mass flow in dimensions of mass per time between two adjacent cells. The mass flow rate includes the contributions from both advection and dispersion if those packages are active

Example Observation Input File

An example GWT Model observation file is shown below.

```
BEGIN OPTIONS
  DIGITS 10
  PRINT_INPUT
END OPTIONS

BEGIN CONTINUOUS FILEOUT my_model.gwt.conc.csv
# obsname  obstype        ID
  L1       CONCENTRATION  1 51 51 # concs at lay 1 row 51 col 51
  L2       CONCENTRATION  2 51 51 # concs at lay 2 row 51 col 51
END CONTINUOUS

BEGIN CONTINUOUS FILEOUT my_model.gwt.mflow.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
```

Advection (ADV) Package

Advection (ADV) Package information is read from the file that is specified by “ADV6” as the file type. Only one ADV Package can be specified for a GWT model.

Structure of Blocks

```
BEGIN OPTIONS
  [SCHEME <scheme>]
END OPTIONS
```

Explanation of Variables

Block: OPTIONS

`scheme`—scheme used to solve the advection term. Can be upstream, central, or TVD. If not specified, upstream weighting is the default weighting scheme.

Example Input File

```
BEGIN OPTIONS
  SCHEME UPSTREAM
END OPTIONS
```

Dispersion (DSP) Package

Dispersion (DSP) Package information is read from the file that is specified by "DSP6" as the file type. Only one DSP Package can be specified for a GWT model. The DSP Package is based on the mathematical formulation presented for the XT3D option of the NPF Package available to represent full three-dimensional anisotropy in groundwater flow. XT3D can be computationally expensive and can be turned off to use a simplified and approximate form of the dispersion equations. For most problems, however, XT3D will be required to accurately represent dispersion.

Structure of Blocks

```
BEGIN OPTIONS
  [XT3D_OFF]
  [XT3D_RHS]
END OPTIONS
```

```
BEGIN GRIDDATA
  [DIFFC [LAYERED]
    <diffc(nodes)> -- READARRAY]
  [ALH [LAYERED]
    <alh(nodes)> -- READARRAY]
  [ALV [LAYERED]
    <alv(nodes)> -- READARRAY]
  [ATH1 [LAYERED]
    <ath1(nodes)> -- READARRAY]
  [ATH2 [LAYERED]
    <ath2(nodes)> -- READARRAY]
  [ATV [LAYERED]
    <atv(nodes)> -- READARRAY]
END GRIDDATA
```

Explanation of Variables

Block: OPTIONS

`XT3D_OFF`—deactivate the xt3d method and use the faster and less accurate approximation. This option may provide a fast and accurate solution under some circumstances, such as when flow aligns with the model grid, there is no mechanical dispersion, or when the longitudinal and transverse dispersivities are equal. This option may also be used to assess the computational demand of the XT3D approach by noting the run time differences with and without this option on.

`XT3D_RHS`—add xt3d terms to right-hand side, when possible. This option uses less memory, but may require more iterations.

Block: GRIDDATA

`diffc`—effective molecular diffusion coefficient.

`alh`—longitudinal dispersivity in horizontal direction. If flow is strictly horizontal, then this is the longitudinal dispersivity that will be used. If flow is not strictly horizontal or strictly vertical, then the longitudinal dispersivity is a function of both ALH and ALV. If mechanical dispersion is represented (by specifying any dispersivity values) then this array is required.

`alv`—longitudinal dispersivity in vertical direction. If flow is strictly vertical, then this is the longitudinal dispsersivity value that will be used. If flow is not strictly horizontal or strictly vertical, then the longitudinal dispersivity is a function of both ALH and ALV. If this value is not specified and mechanical dispersion is represented, then this array is set equal to ALH.

`ath1`—transverse dispersivity in horizontal direction. This is the transverse dispersivity value for the second ellipsoid axis. If flow is strictly horizontal and directed in the x direction (along a row for a regular grid), then this value controls spreading in the y direction. If mechanical dispersion is represented (by specifying any dispersivity values) then this array is required.

`ath2`—transverse dispersivity in horizontal direction. This is the transverse dispersivity value for the third ellipsoid axis. If flow is strictly horizontal and directed in the x direction (along a row for a regular grid), then this value controls spreading in the z direction. If this value is not specified and mechanical dispersion is represented, then this array is set equal to ATH1.

`atv`—transverse dispersivity when flow is in vertical direction. If flow is strictly vertical and directed in the z direction, then this value controls spreading in the x and y directions. If this value is not specified and mechanical dispersion is represented, then this array is set equal to ATH2.

Example Input File

```
BEGIN OPTIONS
END OPTIONS

BEGIN GRIDDATA
  DIFFC
    CONSTANT 1.e-9
  ALH
    CONSTANT 1.
  ALV
    CONSTANT 1.
  ATH1
    CONSTANT 0.1
  ATH2
    CONSTANT 0.1
  ATV
    CONSTANT 0.1
END GRIDDATA
```

Source and Sink Mixing (SSM) Package

Source and Sink Mixing (SSM) Package information is read from the file that is specified by "SSM6" as the file type. Only one SSM Package can be specified for a GWT model.

The SSM Package is used to add or remove solute mass from GWT model cells based on inflows and outflows from GWF stress packages. If a GWF stress package provides flow into a model cell, that flow can be assigned a user-specified concentration. This user-specified concentration must be entered as an auxiliary variable in the flow package. In the SOURCES block below, the user provides the name of the package and the name of the auxiliary variable containing concentration values for each boundary. As described below for srctype, there are multiple options for defining this behavior.

If the user does not enter a record for a GWF stress package in the SOURCES block, then inflow to the GWT model is assigned a concentration value of zero. For negative flow rates in GWF stress packages (sinks), the sink concentration is set to the calculated cell concentration.

Structure of Blocks

```
BEGIN OPTIONS
  [PRINT_FLOWS]
  [SAVE_FLOWS]
END OPTIONS
```

```
BEGIN SOURCES
  <pname> <srctype> <auxname>
  <pname> <srctype> <auxname>
  ...
END SOURCES
```

Explanation of Variables

Block: OPTIONS

`PRINT_FLOWS`—keyword to indicate that the list of SSM 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 SSM flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

Block: SOURCES

`pname`—name of the package for which an auxiliary variable contains a source concentration.

`srctype`—keyword indicating how concentration will be assigned for sources and sinks. Keyword must be specified as either AUX or AUXMIXED. For both options the user must provide an auxiliary variable in the corresponding flow package. The auxiliary variable must have the same name as the AUXNAME value that follows. If the AUX keyword is specified, then the auxiliary variable specified by the user will be assigned as the concenration value for groundwater sources (flows with a positive sign). For negative flow rates (sinks), groundwater will be withdrawn from the cell at the simulated concentration of the cell. The AUXMIXED option provides an alternative method

for how to determine the concentration of sinks. If the cell concentration is larger than the user-specified auxiliary concentration, then the concentration of groundwater withdrawn from the cell will be assigned as the user-specified concentration. Alternatively, if the user-specified auxiliary concentration is larger than the cell concentration, then groundwater will be withdrawn at the cell concentration. Thus, the AUXMIXED option is designed to work with the Evapotranspiration (EVT) and Recharge (RCH) Packages where water may be withdrawn at a concentration that is less than the cell concentration.

`auxname`—name of the auxiliary variable in the package PNAME. This auxiliary variable must exist and be specified by the user in that package. The values in this auxiliary variable will be used to set the concentration associated with the flows for that boundary package.

Example Input File

```
BEGIN SOURCES
# pname  srctype       auxname
  WEL-1      AUX  CONCENTRATION
  LAK-1      AUX  CONCENTRATION
  EVT-1 AUXMIXED       ETCONC
END SOURCES
```

Mobile Storage and Transfer (MST) Package

Mobile Storage and Transfer (MST) Package information is read from the file that is specified by "MST6" as the file type. Only one MST Package can be specified for a GWT model.

Structure of Blocks

```
BEGIN OPTIONS
  [SAVE_FLOWS]
  [FIRST_ORDER_DECAY]
  [ZERO_ORDER_DECAY]
  [SORPTION <sorption>]
END OPTIONS
```

```
BEGIN GRIDDATA
  POROSITY [LAYERED]
    <porosity(nodes)> -- READARRAY
  [DECAY [LAYERED]
    <decay(nodes)> -- READARRAY]
  [DECAY_SORBED [LAYERED]
    <decay_sorbed(nodes)> -- READARRAY]
  [BULK_DENSITY [LAYERED]
    <bulk_density(nodes)> -- READARRAY]
  [DISTCOEF [LAYERED]
    <distcoef(nodes)> -- READARRAY]
  [SP2 [LAYERED]
    <sp2(nodes)> -- READARRAY]
END GRIDDATA
```

Explanation of Variables

Block: OPTIONS

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

`FIRST_ORDER_DECAY`—is a text keyword to indicate that first-order decay will occur. Use of this keyword requires that DECAY and DECAY_SORBED (if sorbtion is active) are specified in the GRIDDATA block.

`ZERO_ORDER_DECAY`—is a text keyword to indicate that zero-order decay will occur. Use of this keyword requires that DECAY and DECAY_SORBED (if sorbtion is active) are specified in the GRIDDATA block.

`sorption`—is a text keyword to indicate that sorption will be activated. Valid sorption options include LINEAR, FREUNDLICH, and LANGMUIR. Use of this keyword requires that BULK_DENSITY and DISTCOEF are specified in the GRIDDATA block. If sorption is specified as FREUNDLICH or LANGMUIR then SP2 is also required in the GRIDDATA block.

Block: GRIDDATA

`porosity`—is the aquifer porosity.

`decay`—is the rate coefficient for first or zero-order decay for the aqueous phase of the mobile domain. A negative value indicates solute production. The dimensions of decay for first-order decay is one over time. The dimensions of decay for zero-order decay is mass per length cubed per time. decay

will have no affect on simulation results unless either first- or zero-order decay is specified in the options block.

`decay_sorbed`—is the rate coefficient for first or zero-order decay for the sorbed phase of the mobile domain. A negative value indicates solute production. The dimensions of decay_sorbed for first-order decay is one over time. The dimensions of decay_sorbed for zero-order decay is mass of solute per mass of aquifer per time. If decay_sorbed is not specified and both decay and sorbtion are active, then the program will terminate with an error. decay_sorbed will have no affect on simulation results unless the SORPTION keyword and either first- or zero-order decay are specified in the options block.

`bulk_density`—is the bulk density of the aquifer in mass per length cubed. bulk_density is not required unless the SORBTION keyword is specified.

`distcoef`—is the distribution coefficient for the equilibrium-controlled linear sorption isotherm in dimensions of length cubed per mass. distcoef is not required unless the SORPTION keyword is specified.

`sp2`—is the exponent for the Freundlich isotherm and the sorption capacity for the Langmuir isotherm.

Example Input File

```
BEGIN OPTIONS
  SORBTION
  FIRST_ORDER_DECAY
END OPTIONS

BEGIN GRIDDATA
  POROSITY
    CONSTANT 0.1
  DECAY
    CONSTANT 0.001
  DECAY_SORBED
    CONSTANT 0.001
  BULK_DENSITY
    CONSTANT 1.
  DISTCOEF
    CONSTANT 0.01
END GRIDDATA
```

Immobile Storage and Transfer (IST) Package

Immobile Storage and Transfer (IMD) Package information is read from the file that is specified by "IST6" as the file type. Any number of IST Packages can be specified for a single GWT model. This allows the user to specify triple porosity systems, or systems with as many immobile domains as necessary.

Structure of Blocks

```
BEGIN OPTIONS
  [SAVE_FLOWS]
  [SORPTION]
  [FIRST_ORDER_DECAY]
  [ZERO_ORDER_DECAY]
  [CIM FILEOUT <cimfile>]
  [CIM PRINT_FORMAT COLUMNS <columns> WIDTH <width> DIGITS <digits> <format>]
END OPTIONS
```

```
BEGIN GRIDDATA
  [CIM [LAYERED]
    <cim(nodes)> -- READARRAY]
  THETAIM [LAYERED]
    <thetaim(nodes)> -- READARRAY
  ZETAIM [LAYERED]
    <zetaim(nodes)> -- READARRAY
  [DECAY [LAYERED]
    <decay(nodes)> -- READARRAY]
  [DECAY_SORBED [LAYERED]
    <decay_sorbed(nodes)> -- READARRAY]
  BULK_DENSITY [LAYERED]
    <bulk_density(nodes)> -- READARRAY
  DISTCOEF [LAYERED]
    <distcoef(nodes)> -- READARRAY
END GRIDDATA
```

Explanation of Variables

Block: OPTIONS

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

`SORPTION`—is a text keyword to indicate that sorption will be activated. Use of this keyword requires that BULK_DENSITY and DISTCOEF are specified in the GRIDDATA block. The linear sorption isotherm is the only isotherm presently supported in the IST Package.

`FIRST_ORDER_DECAY`—is a text keyword to indicate that first-order decay will occur. Use of this keyword requires that DECAY and DECAY_SORBED (if sorbtion is active) are specified in the GRIDDATA block.

`ZERO_ORDER_DECAY`—is a text keyword to indicate that zero-order decay will occur. Use of this keyword requires that DECAY and DECAY_SORBED (if sorbtion is active) are specified in the GRIDDATA block.

`CIM`—keyword to specify that record corresponds to immobile concentration.

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

`cimfile`—name of the output file to write immobile concentrations. This file is a binary file that has the same format and structure as a binary head and concentration file. The value for the text variable written to the file is CIM. Immobile domain concentrations will be written to this file at the same interval as mobile domain concentrations are saved, as specified in the GWT Model Output Control file.

`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.

Block: GRIDDATA

`cim`—initial concentration of the immobile domain in mass per length cubed. If CIM is not specified, then it is assumed to be zero.

`thetaim`—porosity of the immobile domain specified as the volume of immobile pore space per total volume (dimensionless).

`zetaim`—mass transfer rate coefficient between the mobile and immobile domains, in dimensions of per time.

`decay`—is the rate coefficient for first or zero-order decay for the aqueous phase of the immobile domain. A negative value indicates solute production. The dimensions of decay for first-order decay is one over time. The dimensions of decay for zero-order decay is mass per length cubed per time. decay will have no affect on simulation results unless either first- or zero-order decay is specified in the options block.

`decay_sorbed`—is the rate coefficient for first or zero-order decay for the sorbed phase of the immobile domain. A negative value indicates solute production. The dimensions of decay_sorbed for first-order decay is one over time. The dimensions of decay_sorbed for zero-order decay is mass of solute per mass of aquifer per time. If decay_sorbed is not specified and both decay and sorbtion are active, then the program will terminate with an error. decay_sorbed will have no affect on simulation results unless the SORPTION keyword and either first- or zero-order decay are specified in the options block.

`bulk_density`—is the bulk density of the aquifer in mass per length cubed. bulk_density will have no affect on simulation results unless the SORBTION keyword is specified in the options block.

`distcoef`—is the distribution coefficient for the equilibrium-controlled linear sorption isotherm in dimensions of length cubed per mass. distcoef will have no affect on simulation results unless the SORBTION keyword is specified in the options block.

Example Input File

```
BEGIN OPTIONS
  SORBTION
  FIRST_ORDER_DECAY
  CIM FILEOUT gwtmodel.imd1.ucn
END OPTIONS
```

```
BEGIN GRIDDATA
  ZETAIM
    CONSTANT 0.01
  THETAIM
    CONSTANT 0.025
  BULK_DENSITY
    CONSTANT     0.25000000
  DISTCOEF
    CONSTANT     0.01000000
  DECAY
    CONSTANT     0.01000000
  DECAY_SORBED
    CONSTANT     0.01000000
END GRIDDATA
```

Constant Concentration (CNC) Package

Constant Concentration (CNC) Package information is read from the file that is specified by "CNC6" as the file type. Any number of CNC Packages can be specified for a single GWT model, but the same cell cannot be designated as a constant concentration by more than one CNC entry.

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)> <conc> [<aux(naux)>] [<boundname>]
  <cellid(ncelldim)> <conc> [<aux(naux)>] [<boundname>]
  ...
END PERIOD
```

All of the stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered, all of the stresses from the previous block are replaced with the stresses in the new PERIOD block. Note that this behavior is different from the advanced packages (MAW, SFR, LAK, and UZF). To turn off all of the stresses for a stress period, a PERIOD block must be specified with no entries. If a PERIOD block is not specified for the first stress period, then no stresses will be applied until the `iper` value of the first PERIOD block in the file.

Explanation of Variables

Block: OPTIONS

`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 concentration value.

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

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

`PRINT_FLOWS`—keyword to indicate that the list of constant concentration 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 concentration 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 Concentration package. See the "Observation utility" section for instructions for preparing observation input files. Tables 30 and 31 lists observation type(s) supported by the Constant Concentration package.

Block: DIMENSIONS

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

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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, CELLID is the node number for the cell.

`conc`—is the constant concentration value. 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 constant concentration. The values of auxiliary variables must be present for each constant concentration. 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 concentration 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_FLOWS
  PRINT_INPUT
  SAVE_FLOWS
END OPTIONS

BEGIN DIMENSIONS
  MAXBOUND 1
END DIMENSIONS

BEGIN PERIOD 1
  1 1 1 1.0
END PERIOD
```

Available observation types

CNC Package observations are limited to the simulated constant concentration mass flow rate (`cnc`). The data required for the CNC Package observation type is defined in table 23. Negative and positive values for an observation represent a loss from and gain to the GWT model, respectively.

Table 23. Available CNC Package observation types.

Model	Observation type	ID	ID2	Description
CNC	cnc	cellid or boundname	–	Mass flow between the groundwater system and a constant-concentration boundary or a group of cells with constant-concentration boundaries.

Example Observation Input File

```
BEGIN OPTIONS
  DIGITS 8
  PRINT_INPUT
END OPTIONS

BEGIN CONTINUOUS FILEOUT my_model.cnc01.csv
# obsname obstype  ID
  cnc_2_1 CNC      1 1 2
  cnc_2_2 CNC      1 2 2
  cnc_2_3 CNC      1 3 2
  cnc_2_4 CNC      1 4 2
END CONTINUOUS
```

```
BEGIN CONTINUOUS FILEOUT my_model.chd02.csv
# obsname    obstype  ID
  cnc_3_flow CNC      CNC_1_3
END CONTINUOUS
```

Mass Source Loading (SRC) Package

Input to the Mass Source Loading (SRC) Package is read from the file that has type "SRC6" in the Name File. Any number of SRC Packages can be specified for a single groundwater transport model.

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)> <smassrate> [<aux(naux)>] [<boundname>]
  <cellid(ncelldim)> <smassrate> [<aux(naux)>] [<boundname>]
  ...
END PERIOD
```

All of the stress package information in the PERIOD block will continue to apply for subsequent stress periods until the end of the simulation, or until another PERIOD block is encountered. When a new PERIOD block is encountered, all of the stresses from the previous block are replaced with the stresses in the new PERIOD block. Note that this behavior is different from the advanced packages (MAW, SFR, LAK, and UZF). To turn off all of the stresses for a stress period, a PERIOD block must be specified with no entries. If a PERIOD block is not specified for the first stress period, then no stresses will be applied until the `iper` value of the first PERIOD block in the file.

Explanation of Variables

Block: OPTIONS

`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 mass loading rate.

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

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

`PRINT_FLOWS`—keyword to indicate that the list of mass source 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 mass source 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 Mass Source package. See the "Observation utility" section for instructions for preparing observation input files. Tables 30 and 31 lists observation type(s) supported by the Mass Source package.

Block: DIMENSIONS

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

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD 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, CELLID is the node number for the cell.

`smassrate`—is the mass source loading rate. A positive value indicates addition of solute mass and a negative value indicates removal of solute mass. 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 mass source. The values of auxiliary variables must be present for each mass source. 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 mass source 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_FLOWS
  PRINT_INPUT
  SAVE_FLOWS
END OPTIONS

BEGIN DIMENSIONS
  MAXBOUND 1
END DIMENSIONS

BEGIN PERIOD 1
  1 1 1 1.0
END PERIOD
```

Available observation types

Mass Source Loading Package observations include the simulated source loading rates (`src`). The data required for each SRC Package observation type is defined in table 24. The `src` observation is equal to the simulated mass source loading rate. Negative and positive values for an observation represent a loss from and gain to the GWT model, respectively.

Table 24. Available SRC Package observation types.

Stress Package	Observation type	ID	ID2	Description
SRC	src	cellid or boundname	–	Mass source loading rate between the groundwater system and a mass source loading boundary or a group of boundaries.

Example Observation Input File

```
BEGIN OPTIONS
  DIGITS 7
  PRINT_INPUT
END OPTIONS

BEGIN CONTINUOUS FILEOUT my_model.src.obs.csv
# obsname         obstype ID
  src-7-102-17    SRC     7  102  17
  src-7-102-17    SRC     CW_1
  sources         SRC     sources
END CONTINUOUS
```

Streamflow Transport (SFT) Package

Streamflow Transport (SFT) Package information is read from the file that is specified by "SFT6" as the file type. There can be as many SFT Packages as necessary for a GWT model. Each SFT Package is designed to work with flows from a corresponding GWF SFR Package. By default MODFLOW 6 uses the SFT package name to determine which SFR Package corresponds to the SFT Package. Therefore, the package name of the SFT Package (as specified in the GWT name file) must match with the name of the corresponding SFR Package (as specified in the GWF name file). Alternatively, the name of the flow package can be specified using the FLOW_PACKAGE_NAME keyword in the options block. The GWT SFT Package cannot be used without a corresponding GWF SFR Package.

The SFT Package does not have a dimensions block; instead, dimensions for the SFT Package are set using the dimensions from the corresponding SFR Package. For example, the SFR Package requires specification of the number of reaches (NREACHES). SFT sets the number of reaches equal to NREACHES. Therefore, the PACKAGEDATA block below must have NREACHES entries in it.

Structure of Blocks

```
BEGIN OPTIONS
  [FLOW_PACKAGE_NAME <flow_package_name>]
  [AUXILIARY <auxiliary(naux)>]
  [FLOW_PACKAGE_AUXILIARY_NAME <flow_package_auxiliary_name>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_CONCENTRATION]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [CONCENTRATION FILEOUT <concfile>]
  [BUDGET FILEOUT <budgetfile>]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
END OPTIONS
```

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

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

Explanation of Variables

Block: OPTIONS

`flow_package_name`—keyword to specify the name of the corresponding flow package. If not specified, then the corresponding flow package must have the same name as this advanced transport package (the name associated with this package in the GWT name file).

`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.

`flow_package_auxiliary_name`—keyword to specify the name of an auxiliary variable in the corresponding flow package. If specified, then the simulated concentrations from this advanced transport package will be copied into the auxiliary variable specified with this name. Note that the flow package must have an auxiliary variable with this name or the program will terminate with an error. If the flows for this advanced transport package are read from a file, then this option will have no affect.

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

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

`PRINT_CONCENTRATION`—keyword to indicate that the list of 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 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 reach flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

`CONCENTRATION`—keyword to specify that record corresponds to concentration.

`concfile`—name of the binary output file to write concentration 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 SFT package. See the "Observation utility" section for instructions for preparing observation input files. Tables 30 and 31 lists observation type(s) supported by the SFT package.

Block: PACKAGEDATA

`rno`—integer value that defines the reach number associated with the specified PACKAGEDATA data on the line. RNO must be greater than zero and less than or equal to NREACHES. Reach information must be specified for every reach or the program will terminate with an error. The program will also terminate with an error if information for a reach is specified more than once.

`strt`—real value that defines the starting concentration for the reach.

`aux`—represents the values of the auxiliary variables for each reach. The values of auxiliary variables must be present for each 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 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.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD block.

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

`reachsetting`—line of information that is parsed into a keyword and values. Keyword values that can be used to start the REACHSETTING string include: STATUS, CONCENTRATION, RAINFALL, EVAPORATION, RUNOFF, and AUXILIARY. These settings are used to assign the concentration of associated with the corresponding flow terms. Concentrations cannot be specified for all flow terms. For example, the Streamflow Package supports a "DIVERSION" flow term. Diversion water will be routed using the calculated concentration of the reach.

```
STATUS <status>
CONCENTRATION <concentration>
RAINFALL <rainfall>
EVAPORATION <evaporation>
RUNOFF <runoff>
INFLOW <inflow>
AUXILIARY <auxname> <auxval>
```

`status`—keyword option to define reach status. STATUS can be ACTIVE, INACTIVE, or CONSTANT. By default, STATUS is ACTIVE, which means that concentration will be calculated for the reach. If a reach is inactive, then there will be no solute mass fluxes into or out of the reach and the inactive value will be written for the reach concentration. If a reach is constant, then the concentration for the reach will be fixed at the user specified value.

`concentration`—real or character value that defines the concentration for the reach. The specified CONCENTRATION is only applied if the reach is a constant concentration 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.

`rainfall`—real or character value that defines the rainfall solute concentration (ML^{-3}) for 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.

evaporation—real or character value that defines the concentration of evaporated water (ML^{-3}) for the reach. If this concentration value is larger than the simulated concentration in the reach, then the evaporated water will be removed at the same concentration as 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.

runoff—real or character value that defines the concentration of runoff (ML^{-3}) for the reach. 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.

inflow—real or character value that defines the concentration of inflow (ML^{-3}) for the reach. 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.

Example Input File

```
BEGIN OPTIONS
  AUXILIARY aux1 aux2
  BOUNDNAMES
  PRINT_INPUT
  PRINT_CONCENTRATION
  PRINT_FLOWS
  SAVE_FLOWS
  CONCENTRATION FILEOUT gwt_sft_02.sft.bin
  BUDGET FILEOUT gwt_sft_02.sft.bud
  OBS6 FILEIN gwt_sft_02.sft.obs
END OPTIONS

BEGIN PACKAGEDATA
# L            STRT            aux1            aux2    bname
  1      0.00000000     99.00000000    999.00000000  REACH1
  2      0.00000000     99.00000000    999.00000000  REACH2
  3      0.00000000     99.00000000    999.00000000  REACH3
END PACKAGEDATA

BEGIN PERIOD 1
  1  STATUS ACTIVE
  2  STATUS ACTIVE
  3  STATUS ACTIVE
END PERIOD 1
```

Available observation types

Streamflow Transport Package observations include reach concentration and all of the terms that contribute to the continuity equation for each reach. Additional SFT Package observations include mass flow

rates for individual reaches, or groups of reaches. The data required for each SFT Package observation type is defined in table 25. Negative and positive values for `sft` observations represent a loss from and gain to the GWT model, respectively. For all other flow terms, negative and positive values represent a loss from and gain from the SFT package, respectively.

Table 25. Available SFT Package observation types.

Stress Package	Observation type	ID	ID2	Description
SFT	concentration	rno or boundname	–	Reach concentration. If boundname is specified, boundname must be unique for each reach.
SFT	flow-ja-face	rno or boundname	rno or –	Mass flow between two reaches. If a boundname is specified for ID1, then the result is the total mass flow for all reaches. If a boundname is specified for ID1 then ID2 is not used.
SFT	storage	rno or boundname	–	Simulated mass storage flow rate for a reach or group of reaches.
SFT	constant	rno or boundname	–	Simulated mass constant-flow rate for a reach or group of reaches.
SFT	from-mvr	rno or boundname	–	Simulated mass inflow into a reach or group of reaches from the MVT package. Mass inflow is calculated as the product of provider concentration and the mover flow rate.
SFT	to-mvr	rno or boundname	–	Mass outflow from a reach, or a group of reaches 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 reach is observed.
SFT	sft	rno or boundname	`iconn` or –	Mass flow rate for a reach or group of reaches and its aquifer connection(s). If boundname is not specified for ID, then the simulated reach-aquifer flow rate at a specific reach connection is observed. In this case, ID2 must be specified and is the connection number `iconn` for reach `rno`.
SFT	rainfall	rno or boundname	–	Rainfall rate applied to a reach or group of reaches multiplied by the rainfall concentration.
SFT	evaporation	rno or boundname	–	Simulated evaporation rate from a reach or group of reaches multiplied by the evaporation concentration.

Table 25. Available SFT Package observation types.—Continued

Stress Package	Observation type	ID	ID2	Description
SFT	runoff	rno or boundname	–	Runoff rate applied to a reach or group of reaches multiplied by the runoff concentration.
SFT	ext-inflow	rno or boundname	–	Mass inflow into a reach or group of reaches calculated as the external inflow rate multiplied by the inflow concentration.
SFT	ext-outflow	rno or boundname	–	External outflow from a reach or group of reaches to an external boundary. If boundname is not specified for ID, then the external outflow from a specific reach is observed. In this case, ID is the reach rno.

Example Observation Input File

```
BEGIN options
  DIGITS 7
  PRINT_INPUT
END options

BEGIN continuous FILEOUT gwt_lkt02.lkt.obs.csv
  sft-1-conc CONCENTRATION 1
  sft-1-extinflow EXT-INFLOW 1
  sft-1-rain RAINFALL 1
  sft-1-roff RUNOFF 1
  sft-1-evap EVAPORATION 1
  sft-1-stor STORAGE 1
  sft-1-const CONSTANT 1
  sft-1-gwt1 SFT 1  1
  sft-1-gwt2 SFT 1  2
  sft-2-gwt1 SFT 2  1
  sft-1-mylake1 SFT MYREACHES
  sft-1-fjf FLOW-JA-FACE 1 2
  sft-2-fjf FLOW-JA-FACE 2 1
  sft-3-fjf FLOW-JA-FACE 2 3
  sft-4-fjf FLOW-JA-FACE 3 2
  sft-5-fjf FLOW-JA-FACE MYREACH1
  sft-6-fjf FLOW-JA-FACE MYREACH2
  sft-7-fjf FLOW-JA-FACE MYREACH3
END continuous
```

Lake Transport (LKT) Package

Lake Transport (LKT) Package information is read from the file that is specified by "LKT6" as the file type. There can be as many LKT Packages as necessary for a GWT model. Each LKT Package is designed to work with flows from a single corresponding GWF LAK Package. By default MODFLOW 6 uses the LKT package name to determine which LAK Package corresponds to the LKT Package. Therefore, the package name of the LKT Package (as specified in the GWT name file) must match with the name of the corresponding LAK Package (as specified in the GWF name file). Alternatively, the name of the flow package can be specified using the FLOW_PACKAGE_NAME keyword in the options block. The GWT LKT Package cannot be used without a corresponding GWF LAK Package.

The LKT Package does not have a dimensions block; instead, dimensions for the LKT Package are set using the dimensions from the corresponding LAK Package. For example, the LAK Package requires specification of the number of lakes (NLAKES). LKT sets the number of lakes equal to NLAKES. Therefore, the PACKAGEDATA block below must have NLAKES entries in it.

Structure of Blocks

```
BEGIN OPTIONS
  [FLOW_PACKAGE_NAME <flow_package_name>]
  [AUXILIARY <auxiliary(naux)>]
  [FLOW_PACKAGE_AUXILIARY_NAME <flow_package_auxiliary_name>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_CONCENTRATION]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [CONCENTRATION FILEOUT <concfile>]
  [BUDGET FILEOUT <budgetfile>]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
END OPTIONS
```

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

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

Explanation of Variables

Block: OPTIONS

`flow_package_name`—keyword to specify the name of the corresponding flow package. If not specified, then the corresponding flow package must have the same name as this advanced transport package (the name associated with this package in the GWT name file).

`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.

`flow_package_auxiliary_name`—keyword to specify the name of an auxiliary variable in the corresponding flow package. If specified, then the simulated concentrations from this advanced transport package will be copied into the auxiliary variable specified with this name. Note that the flow package must have an auxiliary variable with this name or the program will terminate with an error. If the flows for this advanced transport package are read from a file, then this option will have no affect.

`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_CONCENTRATION`—keyword to indicate that the list of lake concentration 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_CONCENTRATION is specified, then concentration 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.

`CONCENTRATION`—keyword to specify that record corresponds to concentration.

`concfile`—name of the binary output file to write concentration 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 LKT package. See the "Observation utility" section for instructions for preparing observation input files. Tables 30 and 31 lists observation type(s) supported by the LKT package.

Block: PACKAGEDATA

`lakeno`—integer value that defines the lake number associated with the specified PACKAGEDATA data on the line. LAKENO must be greater than zero and less than or equal to NLAKES. Lake information must be specified for every lake or the program will terminate with an error. The program will also terminate with an error if information for a lake is specified more than once.

`strt`—real value that defines the starting concentration for the lake.

`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.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD block.

`lakeno`—integer value that defines the lake number associated with the specified PERIOD 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, CONCENTRATION, RAINFALL, EVAPORATION, RUNOFF, and AUXILIARY. These settings are used to assign the concentration of associated with the corresponding flow terms. Concentrations cannot be specified for all flow terms. For example, the Lake Package supports a "WITHDRAWAL" flow term. If this withdrawal term is active, then water will be withdrawn from the lake at the calculated concentration of the lake.

```
STATUS <status>
CONCENTRATION <concentration>
RAINFALL <rainfall>
EVAPORATION <evaporation>
RUNOFF <runoff>
EXT-INFLOW <ext-inflow>
AUXILIARY <auxname> <auxval>
```

`status`—keyword option to define lake status. STATUS can be ACTIVE, INACTIVE, or CONSTANT. By default, STATUS is ACTIVE, which means that concentration will be calculated for the lake. If a lake is inactive, then there will be no solute mass fluxes into or out of the lake and the inactive value will be written for the lake concentration. If a lake is constant, then the concentration for the lake will be fixed at the user specified value.

`concentration`—real or character value that defines the concentration for the lake. The specified CONCENTRATION is only applied if the lake is a constant concentration 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 solute concentration (ML^{-3}) for the 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.

`evaporation`—real or character value that defines the concentration of evaporated water (ML^{-3}) for the lake. If this concentration value is larger than the simulated concentration in the lake, then the evaporated water will be removed at the same concentration as the 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.

`runoff`—real or character value that defines the concentration of runoff (ML^{-3}) 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.

`ext-inflow`—real or character value that defines the concentration of external inflow (ML^{-3}) for the lake. Value must be greater than or equal to zero. 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.

`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
  AUXILIARY aux1 aux2
  BOUNDNAMES
  PRINT_INPUT
  PRINT_CONCENTRATION
  PRINT_FLOWS
  SAVE_FLOWS
  CONCENTRATION FILEOUT gwt_lkt_02.lkt.bin
  BUDGET FILEOUT gwt_lkt_02.lkt.bud
  OBS6 FILEIN gwt_lkt_02.lkt.obs
END OPTIONS

BEGIN PACKAGEDATA
# L            STRT            aux1            aux2    bname
  1      0.00000000     99.00000000    999.00000000 MYLAKE1
  2      0.00000000     99.00000000    999.00000000 MYLAKE2
  3      0.00000000     99.00000000    999.00000000 MYLAKE3
END PACKAGEDATA

BEGIN PERIOD 1
  1  STATUS ACTIVE
  2  STATUS ACTIVE
  3  STATUS ACTIVE
END PERIOD 1
```

Available observation types

Lake Transport Package observations include lake concentration and all of the terms that contribute to the continuity equation for each lake. Additional LKT Package observations include mass flow rates for individ-

ual outlets, lakes, or groups of lakes (`outlet`). The data required for each LKT Package observation type is defined in table 26. Negative and positive values for `lkt` observations represent a loss from and gain to the GWT model, respectively. For all other flow terms, negative and positive values represent a loss from and gain from the LKT package, respectively.

Table 26. Available LKT Package observation types.

Stress Package	Observation type	ID	ID2	Description
LKT	concentration	lakeno or boundname	–	Lake concentration. If boundname is specified, boundname must be unique for each lake.
LKT	flow-ja-face	lakeno or boundname	lakeno or –	Mass flow between two lakes connected by an outlet. If more than one outlet is used to connect the same two lakes, then the mass flow for only the first outlet can be observed. If a boundname is specified for ID1, then the result is the total mass flow for all outlets for a lake. If a boundname is specified for ID1 then ID2 is not used.
LKT	storage	lakeno or boundname	–	Simulated mass storage flow rate for a lake or group of lakes.
LKT	constant	lakeno or boundname	–	Simulated mass constant-flow rate for a lake or group of lakes.
LKT	from-mvr	lakeno or boundname	–	Simulated mass inflow into a lake or group of lakes from the MVT package. Mass inflow is calculated as the product of provider concentration and the mover flow rate.
LKT	to-mvr	outletno or boundname	–	Mass 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, which must be between 1 and NOUTLETS.
LKT	lkt	lakeno or boundname	`iconn` or –	Mass 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` for lake `lakeno`.
LKT	rainfall	lakeno or boundname	–	Rainfall rate applied to a lake or group of lakes multiplied by the rainfall concentration.

Table 26. Available LKT Package observation types.—Continued

Stress Package	Observation type	ID	ID2	Description
LKT	evaporation	lakeno or boundname	–	Simulated evaporation rate from a lake or group of lakes multiplied by the evaporation concentration.
LKT	runoff	lakeno or boundname	–	Runoff rate applied to a lake or group of lakes multiplied by the runoff concentration.
LKT	ext-inflow	lakeno or boundname	–	Mass inflow into a lake or group of lakes calculated as the external inflow rate multiplied by the inflow concentration.
LKT	withdrawal	lakeno or boundname	–	Specified withdrawal rate from a lake or group of lakes multiplied by the simulated lake concentration.
LKT	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.

Example Observation Input File

```
BEGIN options
  DIGITS 7
  PRINT_INPUT
END options

BEGIN continuous FILEOUT gwt_lkt02.lkt.obs.csv
  lkt-1-conc CONCENTRATION 1
  lkt-1-extinflow EXT-INFLOW 1
  lkt-1-rain RAINFALL 1
  lkt-1-roff RUNOFF 1
  lkt-1-evap EVAPORATION 1
  lkt-1-wdrl WITHDRAWAL 1
  lkt-1-stor STORAGE 1
  lkt-1-const CONSTANT 1
  lkt-1-gwt1 LKT 1  1
  lkt-1-gwt2 LKT 1  2
  lkt-2-gwt1 LKT 2  1
  lkt-1-mylake1 LKT MYLAKE1
  lkt-1-fjf FLOW-JA-FACE 1 2
  lkt-2-fjf FLOW-JA-FACE 2 1
  lkt-3-fjf FLOW-JA-FACE 2 3
  lkt-4-fjf FLOW-JA-FACE 3 2
  lkt-5-fjf FLOW-JA-FACE MYLAKE1
  lkt-6-fjf FLOW-JA-FACE MYLAKE2
  lkt-7-fjf FLOW-JA-FACE MYLAKE3
END continuous
```

Multi-Aquifer Well Transport (MWT) Package

Multi-Aquifer Well Transport (MWT) Package information is read from the file that is specified by "MWT6" as the file type. There can be as many MWT Packages as necessary for a GWT model. Each MWT Package is designed to work with flows from a corresponding GWF MAW Package. By default MODFLOW 6 uses the MWT package name to determine which MAW Package corresponds to the MWT Package. Therefore, the package name of the MWT Package (as specified in the GWT name file) must match with the name of the corresponding MAW Package (as specified in the GWF name file). Alternatively, the name of the flow package can be specified using the FLOW_PACKAGE_NAME keyword in the options block. The GWT MWT Package cannot be used without a corresponding GWF MAW Package.

The MWT Package does not have a dimensions block; instead, dimensions for the MWT Package are set using the dimensions from the corresponding MAW Package. For example, the MAW Package requires specification of the number of wells (NMAWWELLS). MWT sets the number of wells equal to NMAWWELLS. Therefore, the PACKAGEDATA block below must have NMAWWELLS entries in it.

Structure of Blocks

```
BEGIN OPTIONS
  [FLOW_PACKAGE_NAME <flow_package_name>]
  [AUXILIARY <auxiliary(naux)>]
  [FLOW_PACKAGE_AUXILIARY_NAME <flow_package_auxiliary_name>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_CONCENTRATION]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [CONCENTRATION FILEOUT <concfile>]
  [BUDGET FILEOUT <budgetfile>]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
END OPTIONS
```

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

```
BEGIN PERIOD <iper>
  <mawno> <mwtsetting>
  <mawno> <mwtsetting>
  ...
END PERIOD
```

Explanation of Variables

Block: OPTIONS

`flow_package_name`—keyword to specify the name of the corresponding flow package. If not specified, then the corresponding flow package must have the same name as this advanced transport package (the name associated with this package in the GWT name file).

`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.

`flow_package_auxiliary_name`—keyword to specify the name of an auxiliary variable in the corresponding flow package. If specified, then the simulated concentrations from this advanced transport package will be copied into the auxiliary variable specified with this name. Note that the flow package must have an auxiliary variable with this name or the program will terminate with an error. If the flows for this advanced transport package are read from a file, then this option will have no affect.

`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_CONCENTRATION`—keyword to indicate that the list of well concentration 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_CONCENTRATION is specified, then concentration are printed for the last time step of each stress period.

`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.

`CONCENTRATION`—keyword to specify that record corresponds to concentration.

`concfile`—name of the binary output file to write concentration 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 MWT package. See the "Observation utility" section for instructions for preparing observation input files. Tables 30 and 31 lists observation type(s) supported by the MWT package.

Block: PACKAGEDATA

`mawno`—integer value that defines the well number associated with the specified PACKAGEDATA data on the line. MAWNO must be greater than zero and less than or equal to NMAWWELLS. Well information must be specified for every well or the program will terminate with an error. The program will also terminate with an error if information for a well is specified more than once.

`strt`—real value that defines the starting concentration for the well.

`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.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD block.

`mawno`—integer value that defines the well number associated with the specified PERIOD data on the line. MAWNO must be greater than zero and less than or equal to NMAWWELLS.

`mwtsetting`—line of information that is parsed into a keyword and values. Keyword values that can be used to start the MWTSETTING string include: STATUS, CONCENTRATION, RAINFALL, EVAPORATION, RUNOFF, and AUXILIARY. These settings are used to assign the concentration of associated with the corresponding flow terms. Concentrations cannot be specified for all flow terms. For example, the Multi-Aquifer Well Package supports a "WITHDRAWAL" flow term. If this withdrawal term is active, then water will be withdrawn from the well at the calculated concentration of the well.

```
STATUS <status>
CONCENTRATION <concentration>
RATE <rate>
AUXILIARY <auxname> <auxval>
```

`status`—keyword option to define well status. STATUS can be ACTIVE, INACTIVE, or CONSTANT. By default, STATUS is ACTIVE, which means that concentration will be calculated for the well. If a well is inactive, then there will be no solute mass fluxes into or out of the well and the inactive value will be written for the well concentration. If a well is constant, then the concentration for the well will be fixed at the user specified value.

`concentration`—real or character value that defines the concentration for the well. The specified CONCENTRATION is only applied if the well is a constant concentration well. 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.

`rate`—real or character value that defines the injection solute concentration (ML^{-3}) for the well. 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.

Example Input File

```
BEGIN OPTIONS
  AUXILIARY aux1 aux2
  BOUNDNAMES
  PRINT_INPUT
  PRINT_CONCENTRATION
  PRINT_FLOWS
  SAVE_FLOWS
  CONCENTRATION FILEOUT gwt_mwt_02.mwt.bin
  BUDGET FILEOUT gwt_mwt_02.mwt.bud
  OBS6 FILEIN gwt_mwt_02.mwt.obs
END OPTIONS

BEGIN PACKAGEDATA
# L            STRT            aux1            aux2    bname
  1      0.00000000     99.00000000    999.00000000 MYWELL1
  2      0.00000000     99.00000000    999.00000000 MYWELL2
  3      0.00000000     99.00000000    999.00000000 MYWELL3
END PACKAGEDATA

BEGIN PERIOD 1
  1  STATUS ACTIVE
  2  STATUS ACTIVE
  3  STATUS ACTIVE
END PERIOD 1
```

Available observation types

Multi-Aquifer Well Transport Package observations include well concentration and all of the terms that contribute to the continuity equation for each well. Additional MWT Package observations include mass flow rates for individual wells, or groups of wells; the well volume (`volume`); and the conductance for a well-aquifer connection conductance (`conductance`). The data required for each MWT Package observation type is defined in table 27. Negative and positive values for `mwt` observations represent a loss from and gain to the GWT model, respectively. For all other flow terms, negative and positive values represent a loss from and gain from the MWT package, respectively.

Table 27. Available MWT Package observation types.

Stress Package	Observation type	ID	ID2	Description
MWT	concentration	mawno or boundname	–	Well concentration. If boundname is specified, boundname must be unique for each well.
MWT	storage	mawno or boundname	–	Simulated mass storage flow rate for a well or group of wells.
MWT	constant	mawno or boundname	–	Simulated mass constant-flow rate for a well or group of wells.

Table 27. Available MWT Package observation types.—Continued

Stress Package	Observation type	ID	ID2	Description
MWT	from-mvr	mawno or boundname	–	Simulated mass inflow into a well or group of wells from the MVT package. Mass inflow is calculated as the product of provider concentration and the mover flow rate.
MWT	mwt	mawno or boundname	`iconn` or –	Mass flow rate for a well or group of wells and its aquifer connection(s). If boundname is not specified for ID, then the simulated well-aquifer flow rate at a specific well connection is observed. In this case, ID2 must be specified and is the connection number `iconn` for well `mawno`.
MWT	rate	mawno or boundname	–	Simulated mass flow rate for a well or group of wells.
MWT	fw-rate	mawno or boundname	–	Simulated mass flow rate for a flowing well or group of flowing wells.
MWT	rate-to-mvr	well or boundname	–	Simulated mass flow rate that is sent to the MVT Package for a well or group of wells.
MWT	fw-rate-to-mvr	well or boundname	–	Simulated mass flow rate that is sent to the MVT Package from a flowing well or group of flowing wells.

Example Observation Input File

```
BEGIN options
  DIGITS 12
  PRINT_INPUT
END options

BEGIN continuous FILEOUT gwt_mwt_02.mwt.obs.csv
  mwt1mwt MWT  1  1
  mwt2mwt MWT  2  1
  mwt3mwt MWT  3  1
  mwt4mwt MWT  4  1
  mwt1conc CONCENTRATION 1
  mwt2conc CONCENTRATION 2
  mwt3conc CONCENTRATION 3
  mwt4conc CONCENTRATION 4
  mwt1stor STORAGE 1
  mwt2stor STORAGE 2
  mwt3stor STORAGE 3
  mwt4stor STORAGE 4
  mwt1cnst CONSTANT 1
  mwt2cnst CONSTANT 2
  mwt3cnst CONSTANT 3
```

```
  mwt4cnst CONSTANT 4
  mwt1fmvr FROM-MVR 1
  mwt2fmvr FROM-MVR 2
  mwt3fmvr FROM-MVR 3
  mwt4fmvr FROM-MVR 4
  mwt1rate RATE  1
  mwt2rate RATE  2
  mwt3rate RATE  3
  mwt4rate RATE  4
  mwt1rtmv RATE-TO-MVR 1
  mwt2rtmv RATE-TO-MVR 2
  mwt3rtmv RATE-TO-MVR 3
  mwt4rtmv RATE-TO-MVR 4
  mwt1fwrt FW-RATE 1
  mwt2fwrt FW-RATE 2
  mwt3fwrt FW-RATE 3
  mwt4fwrt FW-RATE 4
  mwt1frtm FW-RATE-TO-MVR 1
  mwt2frtm FW-RATE-TO-MVR 2
  mwt3frtm FW-RATE-TO-MVR 3
  mwt4frtm FW-RATE-TO-MVR 4
END continuous FILEOUT gwt_mwt_02.mwt.obs.csv
```

Unsaturated Zone Transport (UZT) Package

Unsaturated Zone Transport (UZT) Package information is read from the file that is specified by "UZT6" as the file type. There can be as many UZT Packages as necessary for a GWT model. Each UZT Package is designed to work with flows from a corresponding GWF UZF Package. By default MODFLOW 6 uses the UZT package name to determine which UZF Package corresponds to the UZT Package. Therefore, the package name of the UZT Package (as specified in the GWT name file) must match with the name of the corresponding UZF Package (as specified in the GWF name file). Alternatively, the name of the flow package can be specified using the FLOW_PACKAGE_NAME keyword in the options block. The GWT UZT Package cannot be used without a corresponding GWF UZF Package.

The UZT Package does not have a dimensions block; instead, dimensions for the UZT Package are set using the dimensions from the corresponding UZF Package. For example, the UZF Package requires specification of the number of cells (NUZFCELLS). UZT sets the number of UZT cells equal to NUZFCELLS. Therefore, the PACKAGEDATA block below must have NUZFCELLS entries in it.

Structure of Blocks

```
BEGIN OPTIONS
  [FLOW_PACKAGE_NAME <flow_package_name>]
  [AUXILIARY <auxiliary(naux)>]
  [FLOW_PACKAGE_AUXILIARY_NAME <flow_package_auxiliary_name>]
  [BOUNDNAMES]
  [PRINT_INPUT]
  [PRINT_CONCENTRATION]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [CONCENTRATION FILEOUT <concfile>]
  [BUDGET FILEOUT <budgetfile>]
  [TS6 FILEIN <ts6_filename>]
  [OBS6 FILEIN <obs6_filename>]
END OPTIONS
```

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

```
BEGIN PERIOD <iper>
  <uzfno> <uztsetting>
  <uzfno> <uztsetting>
  ...
END PERIOD
```

Explanation of Variables

Block: OPTIONS

`flow_package_name`—keyword to specify the name of the corresponding flow package. If not specified, then the corresponding flow package must have the same name as this advanced transport package (the name associated with this package in the GWT name file).

`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.

`flow_package_auxiliary_name`—keyword to specify the name of an auxiliary variable in the corresponding flow package. If specified, then the simulated concentrations from this advanced transport package will be copied into the auxiliary variable specified with this name. Note that the flow package must have an auxiliary variable with this name or the program will terminate with an error. If the flows for this advanced transport package are read from a file, then this option will have no affect.

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

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

`PRINT_CONCENTRATION`—keyword to indicate that the list of UZF cell concentration 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_CONCENTRATION is specified, then concentration are printed for the last time step of each stress period.

`PRINT_FLOWS`—keyword to indicate that the list of unsaturated zone 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 unsaturated zone flow terms will be written to the file specified with "BUDGET FILEOUT" in Output Control.

`CONCENTRATION`—keyword to specify that record corresponds to concentration.

`concfile`—name of the binary output file to write concentration 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 UZT package. See the "Observation utility" section for instructions for preparing observation input files. Tables 30 and 31 lists observation type(s) supported by the UZT package.

Block: PACKAGEDATA

`uzfno`—integer value that defines the UZF cell number associated with the specified PACKAGEDATA data on the line. UZFNO must be greater than zero and less than or equal to NUZFCELLS. Unsaturated zone flow information must be specified for every UZF cell or the program will terminate with

an error. The program will also terminate with an error if information for a UZF cell is specified more than once.

`strt`—real value that defines the starting concentration for the unsaturated zone flow cell.

`aux`—represents the values of the auxiliary variables for each unsaturated zone flow. The values of auxiliary variables must be present for each unsaturated zone flow. 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 unsaturated zone flow 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.

Block: 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 or equal to 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 PERIOD block. The information specified in the PERIOD block will continue to apply for all subsequent stress periods, unless the program encounters another PERIOD block.

`uzfno`—integer value that defines the UZF cell number associated with the specified PERIOD data on the line. UZFNO must be greater than zero and less than or equal to NUZFCELLS.

`uztsetting`—line of information that is parsed into a keyword and values. Keyword values that can be used to start the UZTSETTING string include: STATUS, CONCENTRATION, INFILTRATION, UZET, and AUXILIARY. These settings are used to assign the concentration of associated with the corresponding flow terms. Concentrations cannot be specified for all flow terms.

```
STATUS <status>
CONCENTRATION <concentration>
INFILTRATION <infiltration>
UZET <uzet>
AUXILIARY <auxname> <auxval>
```

`status`—keyword option to define UZF cell status. STATUS can be ACTIVE, INACTIVE, or CONSTANT. By default, STATUS is ACTIVE, which means that concentration will be calculated for the UZF cell. If a UZF cell is inactive, then there will be no solute mass fluxes into or out of the UZF cell and the inactive value will be written for the UZF cell concentration. If a UZF cell is constant, then the concentration for the UZF cell will be fixed at the user specified value.

`concentration`—real or character value that defines the concentration for the unsaturated zone flow cell. The specified CONCENTRATION is only applied if the unsaturated zone flow cell is a constant concentration cell. 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.

`infiltration`—real or character value that defines the infiltration solute concentration (ML^{-3}) for the UZF cell. 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.

`uzet`—real or character value that defines the concentration of unsaturated zone evapotranspiration water (ML^{-3}) for the UZF cell. If this concentration value is larger than the simulated concentration in the UZF cell, then the unsaturated zone ET water will be removed at the same concentration as the UZF cell. 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.

Example Input File

```
BEGIN OPTIONS
  AUXILIARY aux1  aux2
  BOUNDNAMES
  PRINT_INPUT
  PRINT_CONCENTRATION
  PRINT_FLOWS
  SAVE_FLOWS
  CONCENTRATION FILEOUT gwt_02.uzt.bin
  BUDGET FILEOUT gwt_02.uzt.bud
  OBS6  FILEIN  gwt_02.uzt.obs
END OPTIONS

BEGIN PACKAGEDATA
# L            STRT            aux1            aux2    bname
  1      0.00000000     99.00000000    999.00000000 MYUZFCELL1
  2      0.00000000     99.00000000    999.00000000 MYUZFCELL2
  3      0.00000000     99.00000000    999.00000000 MYUZFCELL3
END PACKAGEDATA

BEGIN PERIOD 1
  1  STATUS  ACTIVE
  2  STATUS  ACTIVE
  3  STATUS  ACTIVE
END PERIOD 1
```

Available observation types

Unsaturated Zone Transport Package observations include UZF cell concentration and all of the terms that contribute to the continuity equation for each UZF cell. Additional UZT Package observations include mass flow rates for individual UZF cells, or groups of UZF cells. The data required for each UZT Package observation type is defined in table 28. Negative and positive values for `uzt` observations represent a loss from and gain to the GWT model, respectively. For all other flow terms, negative and positive values represent a loss from and gain from the UZT package, respectively.

Table 28. Available UZT Package observation types.

Stress Package	Observation type	ID	ID2	Description
UZT	concentration	uztno or boundname	–	uzt cell concentration. If boundname is specified, boundname must be unique for each uzt cell.
UZT	flow-ja-face	uztno or boundname	uztno or –	Mass flow between two uzt cells. If a boundname is specified for ID1, then the result is the total mass flow for all uzt cells. If a boundname is specified for ID1 then ID2 is not used.
UZT	storage	uztno or boundname	–	Simulated mass storage flow rate for a uzt cell or group of uzt cells.
UZT	from-mvr	uztno or boundname	–	Simulated mass inflow into a uzt cell or group of uzt cells from the MVT package. Mass inflow is calculated as the product of provider concentration and the mover flow rate.
UZT	uzt	uztno or boundname	`iconn` or –	Mass flow rate for a uzt cell or group of uzt cells and its aquifer connection(s). If boundname is not specified for ID, then the simulated uzt-aquifer flow rate at a specific uzt cell connection is observed. In this case, ID2 must be specified and is the connection number for the uzt cell.
UZT	infiltration	uztno or boundname	–	Infiltration rate applied to a uzt cell or group of uzt cells multiplied by the infiltration concentration.
UZT	rej-inf	uztno or boundname	–	Rejected infiltration rate applied to a uzt cell or group of uzt cells multiplied by the infiltration concentration.
UZT	uzet	uztno or boundname	–	Unsaturated zone evapotranspiration rate applied to a uzt cell or group of uzt cells multiplied by the uzt cell concentration.
UZT	infiltration	uztno or boundname	–	Rejected infiltration rate applied to a uzt cell or group of uzt cells multiplied by the infiltration concentration that is sent to the mover package.

Example Observation Input File

```
BEGIN options
```

```
  DIGITS 7
  PRINT_INPUT
END options

BEGIN continuous FILEOUT gwt_02.uzt.obs.csv
  mwt-1-conc CONCENTRATION 1
  mwt-1-stor STORAGE 1
  mwt-1-gwt1 UZT 1  1
  mwt-1-gwt2 UZT 2  2
  mwt-2-gwt1 UZT 3  3
END continuous
```

Flow Model Interface (FMI) Package

Flow Model Interface (FMI) Package information is read from the file that is specified by "FMI6" as the file type. The FMI Package is optional, but if provided, only one FMI Package can be specified for a GWT model.

For most simulations, the GWT Model needs groundwater flows for every cell in the model grid, for all boundary conditions, and for other terms, such as the flow of water in or out of storage. The FMI Package is the interface between the GWT Model and simulated groundwater flows provided by a corresponding GWF Model that is running concurrently within the simulation or from binary budget files that were created from a previous GWF model run. The following are several different FMI simulation cases:

- Flows are provided by a corresponding GWF Model running in the same simulation—in this case, all groundwater flows are calculated by the corresponding GWF Model and provided through FMI to the transport model. This is a common use case in which the user wants to run the flow and transport models as part of a single simulation. The GWF and GWT models must be part of a GWF-GWT Exchange that is listed in mfsim.nam. If a GWF-GWT Exchange is specified by the user, then the user does not need to specify an FMI Package input file for the simulation, unless an FMI option is needed. If a GWF-GWT Exchange is specified and the FMI Package is specified, then the PACKAGEDATA block below is not read or used.

- There is no groundwater flow and the user is interested only in the effects of diffusion, sorption, and decay or production—in this case, FMI should not be provided in the GWT name file and the GWT model should not be listed in any GWF-GWT Exchanges in mfsim.nam. In this case, all groundwater flows are assumed to be zero and cells are assumed to be fully saturated. The SSM Package should not be activated in this case, because there can be no sources or sinks of water. Likewise, none of the advanced transport packages (LKT, SFT, MWT, and UZT) should be specified in the GWT name file. This type of model simulation without an FMI Package is included as an option to represent diffusion, sorption, and decay or growth in the absence of any groundwater flow.

- Flows are provided from a previous GWF model simulation—in this case FMI should be provided in the GWT name file and the head and budget files should be listed in the FMI options block. In this case, FMI reads the simulated head and flows from these files and makes them available to the transport model. There are some additional considerations when the heads and flows are provided from binary files.

  - The binary budget file must contain the simulated flows for all of the packages that were included in the GWF model run. Saving of flows can be activated for all packages by specifying "SAVE_FLOWS" as an option in the GWF name file. The GWF Output Control Package must also have "SAVE BUGET ALL" specified. The easiest way to ensure that all flows and heads are saved is to use the following simple form of a GWF Output Control file:

    ```
    BEGIN OPTIONS
      HEAD FILEOUT mymodel.hds
      BUDGET FILEOUT mymodel.bud
    END OPTIONS

    BEGIN PERIOD 1
      SAVE HEAD ALL
      SAVE BUDGET ALL
    END PERIOD
    ```

  - The binary budget file must have the same number of budget terms listed for each time step. This will always be the case when the binary budget file is created by MODFLOW 6.

- The advanced flow packages (LAK, SFR, MAW, and UZF) all have options for saving a detailed budget file the describes all of the flows for each lake, reach, well, or UZF cell. These budget files can also be used as input to FMI if a corresponding advanced transport package is needed, such as LKT, SFT, MWT, and UZT. If the Water Mover Package is also specified for the GWF Model, then the the budget file for the Water Mover Package will also need to be specified as input to this FMI Package.
- The binary heads file must have heads saved for all layers in the model. This will always be the case when the binary head file is created by MODFLOW 6. This was not always the case as previous MODFLOW versions allowed different save options for each layer.
- If the binary budget and head files have more than one time step for a single stress period, then the budget and head information must be contained within the binary file for every time step in the simulation stress period.
- The binary budget and head files must correspond in terms of information stored for each time step and stress period.
- If the binary budget and head files have information provided for only the first time step of each stress period, then this information will be used for all time steps in the GWT model run for that stress period. This makes it possible to provide flows, for example, from a steady state GWF stress period and have those flows used for all steps in the GWT simulation. With this option, it is possible to have smaller time steps in the GWT model than the time steps used in the GWF model. Note that this cannot be done when the GWF and GWT models are run in the same simulation, because in that case, both models are solved for each time step in the stress period, as listed in the TDIS Package. This option for reading flows from a previous GWF simulation may offer an efficient alternative to running both models in the same simulation, but it comes at the cost of having potentially very large budget files.

Determination of which FMI use case to invoke requires careful consideration of the different advantages and disadvantages of each case. For example, running GWT and GWF in the same simulation can often be faster because GWF flows are passed through memory to the GWT model instead of being written to files. The disadvantage of this approach is that the same time step lengths must be used for both GWF and GWT. Ultimately, it should be relatively straightforward to test different ways in which GWF and GWT interact and select the use case most appropriate for the particular problem.

Structure of Blocks

```
BEGIN OPTIONS
  [FLOW_IMBALANCE_CORRECTION]
END OPTIONS
```

```
BEGIN PACKAGEDATA
  <flowtype> FILEIN <fname>
  <flowtype> FILEIN <fname>
  ...
END PACKAGEDATA
```

Explanation of Variables

Block: OPTIONS

FLOW_IMBALANCE_CORRECTION—correct for an imbalance in flows by assuming that any residual flow error comes in or leaves at the concentration of the cell. When this option is activated, the GWT Model budget written to the listing file will contain two additional entries: FLOW-ERROR and FLOW-CORRECTION. These two entries will be equal but opposite in sign. The FLOW-CORRECTION term is a mass flow that is added to offset the error caused by an imprecise flow balance. If these terms are not relatively small, the flow model should be rerun with stricter convergence tolerances.

Block: PACKAGEDATA

flowtype—is the word GWFBUDGET, GWFHEAD, GWFMOVER or the name of an advanced GWF stress package. If GWFBUDGET is specified, then the corresponding file must be a budget file from a previous GWF Model run. If an advanced GWF stress package name appears then the corresponding file must be the budget file saved by a LAK, SFR, MAW or UZF Package.

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

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

Example Input File

```
BEGIN OPTIONS
  FLOW_IMBALANCE_CORRECTION
END OPTIONS

BEGIN PACKAGEDATA
  GWFBUDGET FILEIN ../flow/mygwfmodel.bud
  GWFHEAD FILEIN ../flow/mygwfmodel.hds
  GWFMOVER FILEIN ../flow/mygwfmodel.hds
  LAK-1 FILEIN   ../flow/mygwfmodel.lak.bud
  SFR-1 FILEIN   ../flow/mygwfmodel.sfr.bud
  MAW-1 FILEIN   ../flow/mygwfmodel.maw.bud
  UZF-1 FILEIN   ../flow/mygwfmodel.uzf.bud
  LAK-2 FILEIN   ../flow/mygwfmodel-2.lak.bud
END PACKAGEDATA
```

Mover Transport (MVT) Package

Mover Transport (MVT) Package information is read from the file that is specified by "MVT6" as the file type. Only one MVT Package can be specified for a GWT model.

The MVT Package is used to route solute mass according to flows from the GWF Water Mover (MVR) Package. This MVT Package must be activated by the user if the MVR Package was active for the GWF Model. Flows from the GWF MVR Package must be available to the GWT model either through activation of a GWF-GWT Exchange or through specification of "GWFMOVER" in the PACKAGEDATA block of the GWT FMI Package.

Structure of Blocks

```
BEGIN OPTIONS
  [PRINT_INPUT]
  [PRINT_FLOWS]
  [SAVE_FLOWS]
  [BUDGET FILEOUT <budgetfile>]
END OPTIONS
```

Explanation of Variables

Block: OPTIONS

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

`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.

`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.

Example Input File

```
BEGIN OPTIONS
  PRINT_INPUT
  PRINT_FLOWS
  SAVE_FLOWS
  BUDGET FILEOUT mygwtmodel.mvt.bud
END OPTIONS
```

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_OUTER_OUTPUT FILEOUT <outer_csvfile>]
  [CSV_INNER_OUTPUT FILEOUT <inner_csvfile>]
  [NO_PTC [<no_ptc_option>]]
END OPTIONS
```

```
BEGIN NONLINEAR
  OUTER_DVCLOSE <outer_dvclose>
  OUTER_MAXIMUM <outer_maximum>
  [UNDER_RELAXATION <under_relaxation>]
  [UNDER_RELAXATION_GAMMA <under_relaxation_gamma>]
  [UNDER_RELAXATION_THETA <under_relaxation_theta>]
  [UNDER_RELAXATION_KAPPA <under_relaxation_kappa>]
  [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_DVCLOSE <inner_dvclose>
  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

Block: OPTIONS

`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 non-linear 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_OUTER_OUTPUT`—keyword to specify that the record corresponds to the comma separated values outer iteration convergence output.

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

`outer_csvfile`—name of the ascii comma separated values output file to write maximum dependent-variable (for example, head) change convergence information at the end of each outer iteration for each time step.

`CSV_INNER_OUTPUT`—keyword to specify that the record corresponds to the comma separated values solver convergence output.

`inner_csvfile`—name of the ascii comma separated values output file to write solver convergence information. Comma separated values output includes maximum dependent-variable (for example, 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.

`NO_PTC`—is a flag that is used to disable pseudo-transient continuation (PTC). Option only applies to steady-state stress periods for models using the Newton-Raphson formulation. For many problems, PTC can significantly improve convergence behavior for steady-state simulations, and for this reason it is active by default. In some cases, however, PTC can worsen the convergence behavior, especially when the initial conditions are similar to the solution. When the initial conditions are similar to, or exactly the same as, the solution and convergence is slow, then the NO_PTC FIRST option should be used to deactivate PTC for the first stress period. The NO_PTC ALL option should also be used in order to compare convergence behavior with other MODFLOW versions, as PTC is only available in MODFLOW 6.

`no_ptc_option`—is an optional keyword that is used to define options for disabling pseudo-transient continuation (PTC). FIRST is an optional keyword to disable PTC for the first stress period, if steady-state and one or more model is using the Newton-Raphson formulation. ALL is an optional keyword to disable PTC for all steady-state stress periods for models using the Newton-Raphson formulation. If NO_PTC_OPTION is not specified, the NO_PTC ALL option is used.

Block: NONLINEAR

`outer_dvclose`—real value defining the dependent-variable (for example, head) change criterion for convergence of the outer (nonlinear) iterations, in units of the dependent-variable (for example, length for head). When the maximum absolute value of the dependent-variable change at all nodes during an iteration is less than or equal to OUTER_DVCLOSE, iteration stops. Commonly,

OUTER_DVCLOSE equals 0.01. The keyword, OUTER_HCLOSE can be still be specified instead of OUTER_DVCLOSE for backward compatibility with previous versions of MODFLOW 6 but eventually OUTER_HCLOSE will be deprecated and specification of OUTER_HCLOSE will cause MODFLOW 6 to terminate with an error.

`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. Under-relaxation is also known as dampening, and is used to reduce the size of the calculated dependent variable before proceeding to the next outer iteration. Under-relaxation can be an effective tool for highly nonlinear models when there are large and often counteracting changes in the calculated dependent variable between successive outer iterations. By default under-relaxation is not used. NONE - under-relaxation is not used (default). SIMPLE - Simple under-relaxation scheme with a fixed relaxation factor (UNDER_RELAXATION_GAMMA) is used. COOLEY - Cooley under-relaxation scheme is used. DBD - delta-bar-delta under-relaxation is used. Note that the under-relaxation schemes are often used in conjunction with problems that use the Newton-Raphson formulation, however, experience has indicated that they also work well non-Newton problems, such as those with the wet/dry options of MODFLOW 6.

`under_relaxation_gamma`—real value defining either the relaxation factor for the SIMPLE scheme or the history or memory term factor of the Cooley and delta-bar-delta algorithms. For the SIMPLE scheme, a value of one indicates that there is no under-relaxation and the full head change is applied. This value can be gradually reduced from one as a way to improve convergence; for well behaved problems, using a value less than one can increase the number of outer iterations required for convergence and needlessly increase run times. UNDER_RELAXATION_GAMMA must be greater than zero for the SIMPLE scheme or the program will terminate with an error. For the Cooley and delta-bar-delta schemes, UNDER_RELAXATION_GAMMA is a memory term that can range between zero and 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_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 dependent-variable (for example, 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 dependent-variable (for example, 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_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.

Block: LINEAR

`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_dvclose`—real value defining the dependent-variable (for example, head) change criterion for convergence of the inner (linear) iterations, in units of the dependent-variable (for example, length for head). When the maximum absolute value of the dependent-variable change at all nodes during an iteration is less than or equal to INNER_DVCLOSE, the matrix solver assumes convergence. Commonly, INNER_DVCLOSE is set equal to or an order of magnitude less than the OUTER_DVCLOSE value specified for the NONLINEAR block. The keyword, INNER_HCLOSE can be still be specified instead of INNER_DVCLOSE for backward compatibility with previous versions of MODFLOW 6 but eventually INNER_HCLOSE will be deprecated and specification of INNER_HCLOSE will cause MODFLOW 6 to terminate with an error.

`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 resid-

ual 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. STRICT–an optional keyword that is used to specify that INNER_RCLOSE represents a infinity-Norm (absolute convergence criteria) and that the dependent-variable (for example, head) and flow convergence criteria must be met on the first inner iteration (this criteria is equivalent to the criteria used by the MODFLOW-2005 PCG package (Hill, 1990)). 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 0.1 times the number of active cells 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_DVCLOSE 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 dependent-variable (for example, 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 29. The values defined for the `simple` complexity option are assigned if the `COMPLEXITY` keyword is not specified in the `OPTIONS` block.

Table 29. IMS variable values for the available complexity options..

Nonlinear Variable	default/simple	moderate	complex
OUTER_DVCLOSE	0.001	0.01	0.1
OUTER_MAXIMUM	25	50	100
UNDER_RELAXATION	NONE	DBD	DBD
UNDER_RELAXATION_THETA	1.0	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_DVCLOSE	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 ALL
  COMPLEXITY MODERATE
END OPTIONS

BEGIN NONLINEAR
  OUTER_DVCLOSE 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_DVCLOSE 1.0E-4
  INNER_RCLOSE 0.001
  LINEAR_ACCELERATION BICGSTAB
  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
  [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

Block: OPTIONS

`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.

Block: CONTINUOUS

`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 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

GWF Observations

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 20). All available observation types are repeated in Table 30 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 30. Available observation types for the GWF Model.

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.

Internal Package	Observation type	ID	ID2	Description
CSUB	csub	icsubno or boundname	–	Flow between the groundwater system and a interbed or group of interbeds.
CSUB	inelastic-csub	icsubno or boundname	–	Flow between the groundwater system and a interbed or group of interbeds from inelastic compaction.
CSUB	elastic-csub	icsubno or boundname	–	Flow between the groundwater system and a interbed or group of interbeds from elastic compaction.
CSUB	coarse-csub	cellid	–	Flow between the groundwater system and coarse-grained materials in a GWF cell.
CSUB	csub-cell	cellid	–	Flow between the groundwater system for all interbeds and coarse-grained materials in a GWF cell.
CSUB	wcomp-csub-cell	cellid	–	Flow between the groundwater system for all interbeds and coarse-grained materials in a GWF cell from water compressibility.
CSUB	sk	icsubno or boundname	–	Convertible interbed storativity in a interbed or group of interbeds. Convertible interbed storativity is inelastic interbed storativity if the current effective stress is greater than the preconsolidation stress. The NODATA value is reported for steady-state stress periods.
CSUB	ske	icsubno or boundname	–	Elastic interbed storativity in a interbed or group of interbeds. The NODATA value is reported for steady-state stress periods.

Table 30. Available observation types.—Continued

Internal Package	Observation types	ID	ID2	Description
CSUB	sk-cell	cellid	–	Convertible interbed and coarse-grained material storativity in a GWF cell. Convertible interbed storativity is inelastic interbed storativity if the current effective stress is greater than the preconsolidation stress. The NODATA value is reported for steady-state stress periods.
CSUB	ske-cell	cellid	–	Elastic interbed and coarse-grained material storativity in a GWF cell. The NODATA value is reported for steady-state stress periods.
CSUB	estress-cell	cellid	–	effective stress in a GWF cell.
CSUB	gstress-cell	cellid	–	geostatic stress in a GWF cell.
CSUB	interbed-compaction	icsubno or boundname	–	interbed compaction in a interbed or group of interbeds.
CSUB	inelastic-compaction	icsubno or boundname	–	inelastic interbed compaction in a interbed or group of interbeds.
CSUB	elastic-compaction	icsubno or boundname	–	elastic interbed compaction a interbed or group of interbeds.
CSUB	coarse-compaction	cellid	–	elastic compaction in coarse-grained materials in a GWF cell.
CSUB	inelastic-compaction-cell	cellid	–	inelastic compaction in all interbeds in a GWF cell.
CSUB	elastic-compaction-cell	cellid	–	elastic compaction in coarse-grained materials and all interbeds in a GWF cell.
CSUB	compaction-cell	cellid	–	total compaction in coarse-grained materials and all interbeds in a GWF cell.
CSUB	thickness	icsubno or boundname	–	thickness of a interbed or group of interbeds.
CSUB	coarse-thickness	cellid	–	thickness of coarse-grained materials in a GWF cell.
CSUB	thickness-cell	cellid	–	total thickness of coarse-grained materials and all interbeds in a GWF cell.
CSUB	theta	icsubno	–	porosity of a interbed .
CSUB	coarse-theta	cellid	–	porosity of coarse-grained materials in a GWF cell.
CSUB	theta-cell	cellid	–	thickness-weighted porosity of coarse-grained materials and all interbeds in a GWF cell.
CSUB	delay-flowtop	icsubno	–	Flow between the groundwater system and a delay interbed across the top of the interbed.

Table 30. Available observation types.—Continued

Internal Package	Observation types	ID	ID2	Description
CSUB	delay-flowbot	icsubno	–	Flow between the groundwater system and a delay interbed across the bottom of the interbed.
CSUB	delay-head	icsubno	idcellno	head in interbed delay cell idcellno (1 <= idcellno <= NDELAYCELLS). The NODATA value is reported for steady-state stress periods.
CSUB	delay-gstress	icsubno	idcellno	geostatic stress in interbed delay cell idcellno (1 <= idcellno <= NDELAYCELLS). The NODATA value is reported for steady-state stress periods.
CSUB	delay-estress	icsubno	idcellno	effective stress in interbed delay cell idcellno (1 <= idcellno <= NDELAYCELLS). The NODATA value is reported for steady-state stress periods.
CSUB	delay-preconstress	icsubno	idcellno	preconsolidation stress in interbed delay cell idcellno (1 <= idcellno <= NDELAYCELLS). The NODATA value is reported for steady-state stress periods.
CSUB	delay-compaction	icsubno	idcellno	compaction in interbed delay cell idcellno (1 <= idcellno <= NDELAYCELLS).
CSUB	delay-thickness	icsubno	idcellno	thickness of interbed delay cell idcellno (1 <= idcellno <= NDELAYCELLS).
CSUB	delay-theta	icsubno	idcellno	porosity of interbed delay cell idcellno (1 <= idcellno <= NDELAYCELLS).
CSUB	preconstress-cell	cellid	–	preconsolidation stress in a GWF cell containing at least one interbed. The NODATA value is reported for steady-state stress periods.

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.

Table 30. Available observation types.—Continued

Stress Package	Observation types	ID	ID2	Description
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.
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.

Table 30. Available observation types.—Continued

Stress Package	Observation types	ID	ID2	Description
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.
LAK	outlet	outletno or boundname	–	Simulated 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`.

Table 30. Available observation types.—Continued

Stress Package	Observation types	ID	ID2	Description
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.
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.

Table 30. Available observation types.—Continued

Stress Package	Observation types	ID	ID2	Description
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.
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.

Table 30. Available observation types.—Continued

Stress Package	Observation types	ID	ID2	Description
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	iuzno 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	iuzno 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	iuzno 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	iuzno or boundname	–	Simulated groundwater evapotranspiration calculated by the UZF package for a UZF cell or a group of UZF cells.
UZF	infiltration	iuzno 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	iuzno or boundname	–	Inflow into a UZF cell from the MVR package for a UZF cell or a group of UZF cells.
UZF	rej-inf	iuzno 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	iuzno 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	iuzno or boundname	–	Simulated unsaturated evapotranspiration calculated by the UZF package for a UZF cell or a group of UZF cells.
UZF	storage	iuzno or boundname	–	Simulated storage flow rate for a UZF package cell or a group of UZF cells.
UZF	net-infiltration	iuzno or boundname	–	Simulated net infiltration rate for a UZF package cell or a group of UZF cells.

Table 30. Available observation types.—Continued

Stress Package	Observation types	ID	ID2	Description
UZF	water-content	iuzno 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.

Exchange	Observation type	ID	ID2	Description
GWF-GWF	flow-ja-face	cellid	cellid	Flow rate for specified exchange.

GWT Observations

Observations are available for GWT models and GWT stress packages. Available observation types have been listed for each package that supports observations (tables 22 to 28). All available observation types are repeated in Table 31 for convenience.

The sign convention adopted for transport 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, negative and positive values represent a loss from and gain to the GWT model, respectively. For advanced transport packages (Package = LKT, MWT, SFT, and UZT), negative and positive values for exchanges with the GWT model (Observation type = lkt, mwt, sft, and uzt) represent a loss from and gain to the GWT 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 31. Available observation types for the GWT Model.

Model	Observation types	ID	ID2	Description
GWT	concentration	cellid	–	Concentration at a specified cell.
GWT	flow-ja-face	cellid	cellid	Mass flow in dimensions of mass per time between two adjacent cells. The mass flow rate includes the contributions from both advection and dispersion if those packages are active

Stress Package	Observation type	ID	ID2	Description
CNC	cnc	cellid or boundname	–	Mass flow between the groundwater system and a constant-concentration boundary or a group of cells with constant-concentration boundaries.
SRC	src	cellid or boundname	–	Mass source loading rate between the groundwater system and a mass source loading boundary or a group of boundaries.
SFT	concentration	rno or boundname	–	Reach concentration. If boundname is specified, boundname must be unique for each reach.
SFT	flow-ja-face	rno or boundname	rno or –	Mass flow between two reaches. If a boundname is specified for ID1, then the result is the total mass flow for all reaches. If a boundname is specified for ID1 then ID2 is not used.
SFT	storage	rno or boundname	–	Simulated mass storage flow rate for a reach or group of reaches.
SFT	constant	rno or boundname	–	Simulated mass constant-flow rate for a reach or group of reaches.
SFT	from-mvr	rno or boundname	–	Simulated mass inflow into a reach or group of reaches from the MVT package. Mass inflow is calculated as the product of provider concentration and the mover flow rate.
SFT	to-mvr	rno or boundname	–	Mass outflow from a reach, or a group of reaches 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 reach is observed.
SFT	sft	rno or boundname	`iconn` or –	Mass flow rate for a reach or group of reaches and its aquifer connection(s). If boundname is not specified for ID, then the simulated reach-aquifer flow rate at a specific reach connection is observed. In this case, ID2 must be specified and is the connection number `iconn` for reach `rno`.
SFT	rainfall	rno or boundname	–	Rainfall rate applied to a reach or group of reaches multiplied by the rainfall concentration.
SFT	evaporation	rno or boundname	–	Simulated evaporation rate from a reach or group of reaches multiplied by the evaporation concentration.

Table 31. Available observation types.—Continued

Stress Package	Observation types	ID	ID2	Description
SFT	runoff	rno or boundname	–	Runoff rate applied to a reach or group of reaches multiplied by the runoff concentration.
SFT	ext-inflow	rno or boundname	–	Mass inflow into a reach or group of reaches calculated as the external inflow rate multiplied by the inflow concentration.
SFT	ext-outflow	rno or boundname	–	External outflow from a reach or group of reaches to an external boundary. If boundname is not specified for ID, then the external outflow from a specific reach is observed. In this case, ID is the reach rno.
LKT	concentration	lakeno or boundname	–	Lake concentration. If boundname is specified, boundname must be unique for each lake.
LKT	flow-ja-face	lakeno or boundname	lakeno or –	Mass flow between two lakes connected by an outlet. If more than one outlet is used to connect the same two lakes, then the mass flow for only the first outlet can be observed. If a boundname is specified for ID1, then the result is the total mass flow for all outlets for a lake. If a boundname is specified for ID1 then ID2 is not used.
LKT	storage	lakeno or boundname	–	Simulated mass storage flow rate for a lake or group of lakes.
LKT	constant	lakeno or boundname	–	Simulated mass constant-flow rate for a lake or group of lakes.
LKT	from-mvr	lakeno or boundname	–	Simulated mass inflow into a lake or group of lakes from the MVT package. Mass inflow is calculated as the product of provider concentration and the mover flow rate.
LKT	to-mvr	outletno or boundname	–	Mass 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, which must be between 1 and NOUTLETS.

Table 31. Available observation types.—Continued

Stress Package	Observation types	ID	ID2	Description
LKT	lkt	lakeno or boundname	`iconn` or –	Mass 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` for lake `lakeno`.
LKT	rainfall	lakeno or boundname	–	Rainfall rate applied to a lake or group of lakes multiplied by the rainfall concentration.
LKT	evaporation	lakeno or boundname	–	Simulated evaporation rate from a lake or group of lakes multiplied by the evaporation concentration.
LKT	runoff	lakeno or boundname	–	Runoff rate applied to a lake or group of lakes multiplied by the runoff concentration.
LKT	ext-inflow	lakeno or boundname	–	Mass inflow into a lake or group of lakes calculated as the external inflow rate multiplied by the inflow concentration.
LKT	withdrawal	lakeno or boundname	–	Specified withdrawal rate from a lake or group of lakes multiplied by the simulated lake concentration.
LKT	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.
MWT	concentration	mawno or boundname	–	Well concentration. If boundname is specified, boundname must be unique for each well.
MWT	storage	mawno or boundname	–	Simulated mass storage flow rate for a well or group of wells.
MWT	constant	mawno or boundname	–	Simulated mass constant-flow rate for a well or group of wells.
MWT	from-mvr	mawno or boundname	–	Simulated mass inflow into a well or group of wells from the MVT package. Mass inflow is calculated as the product of provider concentration and the mover flow rate.

Table 31. Available observation types.—Continued

Stress Package	Observation types	ID	ID2	Description
MWT	mwt	mawno or boundname	`iconn` or –	Mass flow rate for a well or group of wells and its aquifer connection(s). If boundname is not specified for ID, then the simulated well-aquifer flow rate at a specific well connection is observed. In this case, ID2 must be specified and is the connection number `iconn` for well `mawno`.
MWT	rate	mawno or boundname	–	Simulated mass flow rate for a well or group of wells.
MWT	fw-rate	mawno or boundname	–	Simulated mass flow rate for a flowing well or group of flowing wells.
MWT	rate-to-mvr	well or boundname	–	Simulated mass flow rate that is sent to the MVT Package for a well or group of wells.
MWT	fw-rate-to-mvr	well or boundname	–	Simulated mass flow rate that is sent to the MVT Package from a flowing well or group of flowing wells.
UZT	concentration	uztno or boundname	–	uzt cell concentration. If boundname is specified, boundname must be unique for each uzt cell.
UZT	flow-ja-face	uztno or boundname	uztno or –	Mass flow between two uzt cells. If a boundname is specified for ID1, then the result is the total mass flow for all uzt cells. If a boundname is specified for ID1 then ID2 is not used.
UZT	storage	uztno or boundname	–	Simulated mass storage flow rate for a uzt cell or group of uzt cells.
UZT	from-mvr	uztno or boundname	–	Simulated mass inflow into a uzt cell or group of uzt cells from the MVT package. Mass inflow is calculated as the product of provider concentration and the mover flow rate.
UZT	uzt	uztno or boundname	`iconn` or –	Mass flow rate for a uzt cell or group of uzt cells and its aquifer connection(s). If boundname is not specified for ID, then the simulated uzt-aquifer flow rate at a specific uzt cell connection is observed. In this case, ID2 must be specified and is the connection number for the uzt cell.
UZT	infiltration	uztno or boundname	–	Infiltration rate applied to a uzt cell or group of uzt cells multiplied by the infiltration concentration.

Table 31. Available observation types.—Continued

Stress Package	Observation types	ID	ID2	Description
UZT	rej-inf	uztno or boundname	–	Rejected infiltration rate applied to a uzt cell or group of uzt cells multiplied by the infiltration concentration.
UZT	uzet	uztno or boundname	–	Unsaturated zone evapotranspiration rate applied to a uzt cell or group of uzt cells multiplied by the uzt cell concentration.
UZT	infiltration	uztno or boundname	–	Rejected infiltration rate applied to a uzt cell or group of uzt cells multiplied by the infiltration concentration that is sent to the mover package.

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 READARRAY array reading utility is used to read the array. 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 READARRAY 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 TIMEARRAYSERIES 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 Binary Output Files for the Groundwater Flow (GWF) and Groundwater Transport (GWT) Models

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 structure was designed to be self-documenting so that it can evolve if necessary. 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 is not written to the binary grid file, but is shown here to explain the file content. The binary grid file is written for the GWF Model, but is not written for the GWT Model.

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 or SAVE_SPECIFIC_DISCHARGE options are specified in the NPF Package. 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" for the GWF Model and "CONCENTRATION" for the GWT Model. 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).

Advanced Flow and Transport Packages

The dependent variable can be saved to a binary file for the LAK, SFR, and MAW Packages of the GWF Model and for the LKT, SFT, MWT, and UZT Packages of the GWT Model. Table 32 shows the text identifier and description of the dependent variable for these packages.

Table 32. Dependent variable written for advanced flow and transport packages.

Model/Package	TEXT	Description
GWF/LAK	STAGE	Simulated lake stage
GWF/SFR	STAGE	Simulated stream reach stage
GWF/MAW	HEAD	Simulated well head
GWT/LKT	CONCENTRATION	Simulated lake concentration
GWT/SFT	CONCENTRATION	Simulated stream reach concentration
GWT/MWT	CONCENTRATION	Simulated well concentration
GWT/UZT	CONCENTRATION	Simulated unsaturated zone cell concentration

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`).

Model Budget Files

MODFLOW 6 can optionally write a budget file, also referred to as a cell-by-cell flow file. The budget file is written in a binary format that can be post-processed using other software programs, such as ZONEBUDGET. The budget file for the MODFLOW 6 models, such as the GWF and GWT Models, contains intercell water and solute 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.

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 (corresponding to Record 3 for IMETH=1), 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
```

Variations for Discretization Types

The format for the GWF and GWT Model budget files 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 33).

Table 33. 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 for the GWF Model

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 34 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 34 are flows to or a from a GWF Model cell. As described previously, 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. If the model has an active Skeletal Storage, Compaction, and Subsidence Package, then CSUB-CGELASTIC and CSUB-WATERCOMP are written to the budget file using IMETH=1.

The remaining flow terms in table 34 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 34, 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 34. 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)
CSUB-CGELASTIC	1	coarse-grained elastic storage from CSUB Package; array of size (NCELLS)
CSUB-WATERCOMP	1	water compressibility from CSUB Package; array of size (NCELLS)
CSUB-ELASTIC	6	interbed elastic storage from CSUB package; list of size(NINTERBEDS)
CSUB-INELASTIC	6	interbed inelastic storage from CSUB package; list of size(NINTERBEDS)
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

Table 34. Types of information that may be contained in the GWF Model budget file.

Flow Type (TEXT)	Method Code (IMETH)	Description
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
DATA-SPDIS	6	specific discharge at the cell center. The x, y, and z components are stored in auxiliary variables called “qx”, “qy”, and “qz”, respectively. The flow value written for each cell is zero. The “DATA” prefix on the text identifier can be used by post-processors to recognize that the record does not contain a cell flow budget term.
DATA-SAT	6	cell saturation. The cell saturation is stored in an auxiliary variable called “sat”. The flow value written for each cell is zero. The “DATA” prefix on the text identifier can be used by post-processors to recognize that the record does not contain a cell flow budget term. The cell saturation can be used by post-processors to determine how much of the cell is saturated without having to know the value for ICELLTYPE or the value for head. If a cell is marked as confined (ICELLTYPE=0) then saturation is always one. If ICELLTYPE is one, then saturation ranges between zero and one. For Newton GWF simulations, saturation is zero if the head is below the cell bottom.

GWF Model CSUB Package

For each stress period, time step, and compaction data type that is saved to the CSUB Package binary output files as `IMETH=1` budget file type. The compaction data that are written to the CSUB Package binary files are summarized in Tables 35.

Table 35. Data written to the CSUB Package compaction binary output files.

Flow Type (TEXT)	Method Code (IMETH)	Description
`CSUB-COMPACTION`	1	total compaction for cell; array of size (NCELLS)
`CSUB-INELASTIC`	1	inelastic compaction for cell; array of size (NCELLS)
`CSUB-ELASTIC`	1	elastic compaction for cell; array of size (NCELLS)
`CSUB-INTERBED`	1	interbed compaction for cell; array of size (NCELLS)
`CSUB-COARSE`	1	coarse-grained compaction for cell; array of size (NCELLS)
`CSUB-ZDISPLACE`	1	z-displacement for cell; z-displacement of the upper most model cells represents subsidance at land-surface; array of size (NCELLS)

GWF Model 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 36 to 39, respectively.

Table 36. 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`).

Flow term	IMETH	NDAT / NLIST	Description
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).
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 number LAKEIN for the connected outlet 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 37. 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).
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).

Flow term	IMETH	NDAT / NLIST	Description
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 38. 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 / $\sum_{n=1}^{\mathtt{maxbound}} \mathtt{nconn}_n$	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).
STORAGE	6	2 / maxbound	Calculated storage changes for each reach. This value is always zero for the present implementation. The water volume in the reach (VOLUME) is saved as an auxiliary data item for this flow term. 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).

Flow term	IMETH	NDAT / NLIST	Description
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 39. 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).

Budget File Contents for the GWT Model

The type of information that is written to the budget file for a GWT Model depends on the packages used for the model and whether or not the save flags are set. Table 40 contains a list of the types of information that may be contained in a GWT Model budget file. In all cases, the flows in table 40 are solute mass flows (in mass per time) to or a from a GWT Model cell. Intercell flows are written as FLOW-JA-FACE using IMETH=1.

The remaining flow terms in table 40 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 GWT Model budget file, TXT1ID1 is the name of the GWT Model and TXT2ID1 is also the name of the GWT Model. These text identifiers describe what is contained in ID1. For the GWT Model budget file, ID1 is the cell or node number in the GWT Model grid. The second set of text identifiers refer to the information in ID2. Unless noted otherwise in the description in table 40, TXT1ID2 is the name of the GWT Model, TXT2ID2 is the name of the package, and ID2 is the bound number in the package; for example, this is the first constant concentration cell, second constant concentration cell, and so forth.

Table 40. Types of information that may be contained in the GWT Model budget file. All terms represent solute flows in dimensions of mass per time.

Flow Type (TEXT)	Method Code (IMETH)	Description
FLOW-JA-FACE	1	intercell solute flow due to advection and dispersion; array of size(NJA)
STORAGE-AQUEOUS	1	solute aqueous storage; array of size (NCELLS)
DECAY-AQUEOUS	1	solute aqueous decay; array of size (NCELLS)
STORAGE-SORBED	1	solute sorbed storage; array of size (NCELLS)
DECAY-SORBED	1	solute sorbed decay; array of size (NCELLS)
SOURCE-SINK MIX	6	mass flow from SSM sources and sinks
CNC	6	mass flow for constant-concentration cells
SRC	6	mass flow for specified mass source cells
LKT	6	mass flow between lake and aquifer
SFT	6	mass flow between stream and aquifer
MWT	6	mass flow between multi-aquifer well and aquifer
UZT	6	mass flow between unsaturated zone cell and aquifer
IST	6	mass flow between mobile and immobile domain
FLOW-JA-FACE	6	flow to or from a cell in another GWT Model (note that this is not implemented yet for the GWT Model); TXT1ID1 is the name of the GWT Model described by this budget file, TXT2ID1 is the name of the GWF-GWF Exchange, TXT1ID2 is the name of the connected GWT Model, TXT2ID2 is the name of the GWT-GWT Exchange, and ID2 is the cell or node number of the cell in the connected model

GWT Model LKT, MWT, SFT, and UZT Packages

For each stress period, time step, and data type that is saved to theLKT, MWT, SFT, and UZT Packages binary output files as `IMETH=6` budget file type. For all advanced transport 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 LKT, MWT, SFT, and UZT Package binary files are mass flows with entries similar to those listed in Tables 36 to 39 for the advanced flow packages.

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) will always be "double" to indicate that floating-point values are written in 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., 1990, Preconditioned Conjugate-Gradient 2 (PCG2), a computer program for solving ground-water flow equations: U.S. Geological Survey Water-Resources Investigations Report 90–4048, 25 p., accessed June 27, 2017, at https://pubs.usgs.gov/wri/wrir_90-4048.

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., Thorne Jr, D.T., Dausman, A.M., Sukop, M.C., and Guo, Weixing, 2008, SEAWAT Version 4—A computer program for simulation of multi-species solute and heat transport: U.S. Geological Survey Techniques and Methods, book 6, chap. A22, 39 p., accessed June 27, 2017, at https://pubs.er.usgs.gov/publication/tm6A22.

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.

Langevin, C.D., Panday, Sorab, and Provost, A.M., 2020, Hydraulic-head formulation for density-dependent flow and transport: Groundwater, v. 58, no. 3, p. 349–362.

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.

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 groundwater 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, 2010, MT3DMS v5.3, Supplemental User's Guide: Technical Report Prepared for the U.S. Army Corps of Engineers, 51 p.

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	BUY	OPTIONS	yes
GWF	BUY	DIMENSIONS	yes
GWF	BUY	PACKAGEDATA	yes
GWF	STO	OPTIONS	yes
GWF	STO	GRIDDATA	no
GWF	STO	PERIOD	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	CSUB	OPTIONS	yes
GWF	CSUB	DIMENSIONS	yes
GWF	CSUB	GRIDDATA	no
GWF	CSUB	PACKAGEDATA	yes
GWF	CSUB	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
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

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	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	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
GWF	OC	OPTIONS	yes
GWF	OC	PERIOD	yes
GWT	ADV	OPTIONS	yes
GWT	DSP	OPTIONS	yes
GWT	DSP	GRIDDATA	no
GWT	CNC	OPTIONS	yes
GWT	CNC	DIMENSIONS	yes
GWT	CNC	PERIOD	yes
GWT	DIS	OPTIONS	yes
GWT	DIS	DIMENSIONS	yes
GWT	DIS	GRIDDATA	no
GWT	DISV	OPTIONS	yes
GWT	DISV	DIMENSIONS	yes
GWT	DISV	GRIDDATA	no
GWT	DISV	VERTICES	yes
GWT	DISV	CELL2D	yes
GWT	DISU	OPTIONS	yes
GWT	DISU	DIMENSIONS	yes
GWT	DISU	GRIDDATA	no
GWT	DISU	CONNECTIONDATA	yes
GWT	DISU	VERTICES	yes
GWT	DISU	CELL2D	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
GWT	IC	GRIDDATA	no
GWT	NAM	OPTIONS	yes
GWT	NAM	PACKAGES	yes
GWT	OC	OPTIONS	yes
GWT	OC	PERIOD	yes
GWT	SSM	OPTIONS	yes
GWT	SSM	SOURCES	yes
GWT	SRC	OPTIONS	yes
GWT	SRC	DIMENSIONS	yes
GWT	SRC	PERIOD	yes
GWT	MST	OPTIONS	yes
GWT	MST	GRIDDATA	no
GWT	IST	OPTIONS	yes
GWT	IST	GRIDDATA	no
GWT	SFT	OPTIONS	yes
GWT	SFT	PACKAGEDATA	yes
GWT	SFT	PERIOD	yes
GWT	LKT	OPTIONS	yes
GWT	LKT	PACKAGEDATA	yes
GWT	LKT	PERIOD	yes
GWT	MWT	OPTIONS	yes
GWT	MWT	PACKAGEDATA	yes
GWT	MWT	PERIOD	yes
GWT	UZT	OPTIONS	yes
GWT	UZT	PACKAGEDATA	yes
GWT	UZT	PERIOD	yes
GWT	FMI	OPTIONS	yes
GWT	FMI	PACKAGEDATA	yes
GWT	MVT	OPTIONS	yes
UTL	LAK	DIMENSIONS	yes
UTL	LAK	TABLE	yes
UTL	OBS	OPTIONS	yes
UTL	OBS	CONTINUOUS	yes
UTL	TS	ATTRIBUTES	yes
UTL	TS	TIMESERIES	yes
UTL	TAS	ATTRIBUTES	yes
UTL	TAS	TIME	yes

Publishing support provided by the U.S. Geological Survey
MODFLOW 6 Development Team

For information concerning this publication, please contact:

Office of Groundwater
U.S. Geological Survey
Mail Stop 411
12201 Sunrise Valley Drive
Reston, VA 20192
(703) 648–5001
https://water.usgs.gov/ogw/

https://doi.org/10.5066/F76Q1VQV