# Appendix 3. Modflow Upgrades and Updates

This appendix covers the set of upgrades and revisions to the MODFLOW-2005 base code incorporating new features that expand on the existing packages (fig. 3.1). It begins by discussing some global changes to the MODFLOW base code and how the "Options" section of input to certain packages has been changed to a block-style input (appendix 1). It then discusses expansions to the Discretization (DIS) and Basic (BAS) packages, which now have calendar dates included as part of the simulations to support additional, related options. The next section covers the new Budget Group feature, which allows some of the packages to divide their budgets into multiple groups for printing to the Listing File or separate Budget File (BAS package option **BUDGETDB**). The appendix continues with a discussion of the new Well (WEL) package and continued availability of the original package, which is now accessed with the keyword **WEL1** in the Name file. Subsequent sections present modifications to the General Head Boundary (GHB) package that includes variable conductance; improvements to the Newton-Raphson numerical solver (NWT), Stream Flow Routing (SFR), Parameter Value (PVAL), Multiplier Array (MULT), and Zone Array (ZONE) packages; the LIST and Name files; and a summary of the new Warning Package (WARN), which holds all warnings from packages. This appendix concludes with the new time-step based output available for the Observation (OBS) process and a new stress-period based input of convergence criteria for the NWT and Precondition Conjugate Gradient (PCG) solvers.

| Package | Full Name and Description |
|---|---|
| Name | Name file that describes all packages in use for a simulation and location of their input files. |
| LIST | Listing file that contain a transcript of the simulation input and output. |
| WARN | Warning file that contain a transcript of the simulation errors and warnings. |
| BAS | Basic package that contains global options and initial starting heads. |
| DIS | Discretization package that specifies the model's spatial and temporal discretization. |
| PCG | Precondition Conjugate Gradient numerical solver for the groundwater equations. |
| PCGN | Precondition Conjugate Gradient Nonlinear numerical solver for the groundwater equations. |
| NWT | Newton-Raphson numerical solver for the groundwater equations. |
| SFR | Stream Flow Routing package that simulates 1-D river and stream flow. |
| GHB | General Head Boundary package that defines a head dependent boundary condition. |
| WEL | Well package that extracts or injects groundwater. |
| PVAL | Parameter Value package that overrides each package parameter values. |
| MULT | Multiplier Array package that specifies a set of arrays that can be used with parameters. |
| ZONE | Zone Array package that specifies a set of arrays that can be used with parameters. |
| OBS | Observation process that reports observations for various packages.<br>(Packages part of the process: HOB, DROB, DRTOB, GBOB, CHOB, RVOB.) |

**Figure 3.1.** MF-OWHM list of modified of MODFLOW-2005 base-packages. For MF-OWHM2, these specific packages are either completely rewritten or had substantial code updates to add new features. [**Abbreviation**: 1-D, one-dimensional.]

## Free Format Input Files Are Now Default

The input file structure used by MODFLOW-2005, by default, had a fixed-format input. This input structure relied on the number of columns of text to define the input. For example, if an input file required two integer variables to be read in a fixed-format, the first integer must reside within the first 10 columns, and the second integer must reside within the second 10 columns. If the integers were outside the correct columns, then MODFLOW would "crash." Figure 3.2 is an example of input formatted for a fixed-format read of the numbers `28` and `123`.

If the user wanted to use free format, then it was required to specify the **`OPTION`** keyword **`FREE`** in BAS. Free format indicates that the input data may be placed anywhere on a line and multiple values are separated by either one or more spaces or tabs or by one comma. This release of MF-OWHM2 presumes free format by default, as if the BAS option **`FREE`** is always specified. Under rare circumstances, the fixed-format input is not compatible with **`FREE`**. Figure 3.3 shows two examples of acceptable fixed formats that fail to be read correctly using a free-format option. When read as fixed 10-column fields of formatted input, the first loads the numbers `25` and `35` (fig. 3.3*A*), and the second loads the numbers `1551313035` and `9203456138` (fig. 3.3*B*).

To maintain compatibility with fixed-format input files that fail to load correctly under the default free format, the BAS package options include the new keyword **`NOFREE`**. This option enables the legacy fixed-format input structure for the base MODFLOW packages. If the option **`FREE`** is specified for a BAS package, then it is ignored because it is now the default option for all simulations.

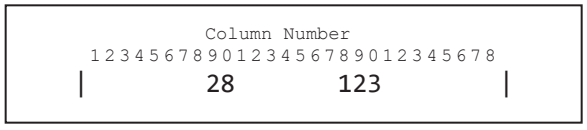


**Figure 3.2.** Fixed-format input example that supplies 2 integer inputs that each occupy 10 columns of a row of text input. The software reads an integer from columns 1 to 10 and another integer from columns 11 to 20.

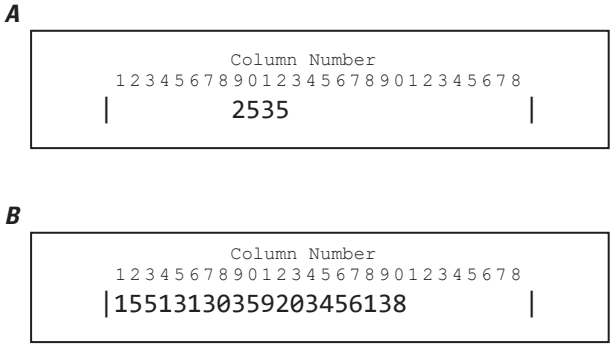


**Figure 3.3.** Examples of acceptable fixed formats that fail to be read correctly using a free-format option. If coded for reading fixed-format input, the software reads an integer from columns 1 to 10 and another integer from columns 11 to 20. *A*, Example that results in a fixed-format read to load 25 and 35, whereas a free-format read loads 2,535 and raises an error for failing to load a second number. *B*, Example that results in a fixed-format read to load 1551313035 and 9203456138, whereas a free-format read loads 15,513,130,359,203,456,138 and raises an error for failing to load a second number.

## Double Precision Number Simulation

MODFLOW-2005 stores floating point numbers as a mix of single and double precision. A single precision number contains seven significant digits—for example `1.234567`—before rounding and can have scientific notation to `1E±38`. Double precision has 15 significant digits—for example `1.23456789012345`—and can have scientific notation to `1E±308`. The disadvantage of double precision numbers is that the random-access memory (RAM) required to store the number (8 byte) is twice that of single precision numbers (4 byte). Given that current computer systems are not memory limited, as was the case in 2005, all real numbers in MF-OWHM2 are now stored using double precision. This improves the accuracy of the simulation and may reduce the simulation time as a result of faster solver convergence.

To maintain backward compatibility with post-processing programs, such as ZoneBudget, the MODFLOW-2005 standard output, binary files—those written with *ULASAV*, *UBDSVA*, *UBDSVB*, *UBUDSV*, *UBDSV1*, *UBDSV2*, *UBDSV3*, and *UBDSV4*—write the double precision numbers as single precision. The most common output file that uses these write utilities is the MODFLOW Cell-By-Cell (CBC) file. The BAS package offers a new option, **`DOUBLE_PRECISION_CBC`**, that instructs MF-OWHM2 to maintain the double precision numbers when writing to the CBC. It should be noted that another advantage of using single precision variables to write to the CBC is that the resulting binary output file uses approximately half the hard drive space as a double precision binary file.

## Additional Convergence Metric

The groundwater-flow finite difference equations can be compactly written as Ah = RHS, where A is a matrix, h is a vector containing every model cell's head, and RHS represents the right-hand side of the equations (non-head-dependent components). A solver iteration, called an "outer iteration," requires each package—using the current set of heads in h—to modify the RHS and the A matrix, and then solves for a new h vector. Please see Harbaugh (2005) for a full definition of how the A matrix is assembled and h is solved for.

MODFLOW solvers typically have two convergence criteria. The first is a head tolerance—called `HCLOSE`—that represents the change between the previous solved head vector and the newly solved head vector, h, that is aggregated to a single number using the $L_2$ or $L_\infty$ norm. The second criterion is a flow residual tolerance—called `RCLOSE`—that represents the discrepancy between the matrix-vector multiplication of A and the updated h vector to the RHS. Specifically, it solves for Ah – RHS = r. The vector, r, has model units of $L^3/T$ and represents the flow residual error. This second solver's convergence criterion is met when the $L_2$ or $L_\infty$ norm of $r$ is less than `RCLOSE`; the type of norm depends on the solver used. If the equation is solved "perfectly," then the flow residual vector is zero (that is, $r = 0$).

MF-OWHM2 includes a third criterion, called the Relative Volume Error, before convergence is allowed. The Relative Volume Error is the flow residual error vector, r, normalized by the corresponding model cell volume vector, v, which mathematically is r/v and has model units of 1/T. MF-OWHM2 requires that the $L_\infty$ norm of the Relative Volume Error be less than `0.025` before convergence is allowed. Specifically, it is required that $\left\| r/v \right\|_\infty \leq 0.025$ must be true before convergence is allowed. To override the default value, BAS has the option **`MAX_RELATIVE_VOLUME_ERROR`**, which is followed by a new maximum allowed relative-volume error.

## Package Options Moved to Block-Style Input

The Basic (BAS), Unsaturated-Zone Flow (UZF), Streamflow Routing (SFR), General-Head Boundary (GHB), and Well (WEL) packages were modified to have their keyword **OPTIONS** loaded using a block style input. The `Options` line of the BAS package, located on Data Set 1 (see the "New Basic Package Options" section), is replaced with **BEGIN OPTIONS** and then has one option per line; the block is terminated by the keyword **END**. The original BAS input `Options` line—that is, the original input structure—is still supported to maintain backward compatibility, but the `Options` line does not support the input of any of the new keywords discussed in this appendix. If the **OPTIONS** block is used, then it must be located at the beginning of the BAS input file along with any other *Block Style* input. If the package supports additional block inputs, then the order of the block inputs does not matter—for example, the **BUDGET_GROUPS** and **OPTIONS** block may be specified in any order. Additionally, if the package uses the optional keyword **PARAMETER**, then it may appear before or after the block input.

For backward compatibility with MODFLOW-2005, the `Options` line is still checked for the options that it supports, but all MF-OWHM2 specific options must now be in the new block style. For models that still use the `Options` line, an innocuous warning is raised during runtime indicating that the input should be changed to block style. If only the original options are used, then the program continues after the warning, but if one of the new options of MF-OWHM2 is used, then it raises an error and stops the program.

In the release of MODFLOW-NWT version 1.1 (Niswonger, and others, 2011), the `Options` and keyword input sections for the SFR and UZF packages were changed to support a *Block Style* input but are backwardly compatible and still support specifying all options on a single line. To maintain backward compatibility in this version of MODFLOW-NWT, the SFR and UZF options only support *Block Style* input. MF-OWHM2 does check for the original `Options` input location, and if it finds the keywords, an error is raised indicating that the keyword must be moved to an **OPTIONS** block. Figure 3.4 shows an example BAS **OPTIONS** block (using MF-OWHM2 options) that must be placed at the start of the input file.

```
BEGIN OPTIONS
      START_DATE  DATE
      NOFREE
      CBC_EVERY_TIMESTEP
      NO_FAILED_CONVERGENCE_STOP
      PERCENTERROR PDIFFPRT
END
```

**Figure 3.4.** Example of **OPTIONS** block (using MF-OWHM2 options) that must be placed at the start of the BAS package file. [For using during a simulation, DATE is replaced by the starting calendar date of the model; PDIFFPRT is replaced by the mass error percentage that results in raising a percent error warning.]

## Calendar Dates

A new Fortran module containing code composed of a calendar datetime object is included in MF-OWHM2. The calendar datetime object allows the use of calendar dates as part of the simulation input and for MF-OWHM2 to keep track of the starting and ending date of each time step. The calendar dates work with all the MF-OWHM2 time units (DIS variable `ITMUNI`), but it is not recommended to use a time unit of years (`ITMUNI = 5`) because of the ambiguity between leap years and non-leap years. The time unit of undefined (`ITMUNI = 0`) is treated as if a time unit of days (`ITMUNI = 4`) had been specified.

If calendar dates are used by any of the MF-OWHM2 features, then a starting calendar date must be specified with the BAS package option **`START_DATE`**, which is followed by the starting calendar date (`DATE`). The **`START_DATE`** keyword replaces the use of the DIS package keyword **`STARTTIME`**, which was used in MF-OWHM to define a starting date using decimal years. **`STARTTIME`** is still supported but is not recommended because it does not account for leap years nor does it allow calendar dates for MF-OWHM2 input. Figure 3.5*A* defines the symbology used, and figure 3.5*B* lists the accepted formats for `DATE`. It is recommended to use either "American" style date format, mm/dd/yyyy, or to use the International Organization for Standardization (ISO) standard, yyyy-mm-dd, for specifying calendar dates.

If the **`START_DATE`**'s "`DATE`" is specified as a decimal year, DYear (fig. 3.5*B*), then it is automatically converted to a calendar date. MF-OWHM2 converts a decimal year to a calendar date by first splitting the integer part from the fractional part (such as `1979.307` to `1979` and `0.307`). The integer part is the calendar year that is either a "non-leap" or "leap" year—the latter of which includes February 29th. The fractional part is then converted to a Julian day of the year (DoY) by multiplying by `365` or `366` days—depending on if the year is a "non-leap" or "leap" year, respectively. The DoY is then converted to the appropriate month and day. This procedure is different when using the DIS package "**`STARTTIME`** DYear" option, which did not distinguish between "non-leap" and "leap" years and assumed `365.2425` days in all years. Figure 3.5*C* presents the day of the year for "non-leap" and "leap" years with the corresponding decimal fraction of a year for select dates. The fraction of the year is determined by subtracting one from the day of the year and dividing it by the total number of days in the year. This results in January 1st always having a fraction of the year equaling 0.0, but December 31st is never 1.0—instead, it is approximately 0.99726 of a year. The reason the fraction is set up this way is to ensure that the date always starts at the start of the day, which is 00:00:00. An example conversion of `1979.307` yields the calendar date `4/23/1979` with a 24-hour clock time of 1:19:12 (hour:minute:second, respectively).

If a starting calendar date is specified, then MF-OWHM2 keeps track of each stress-period date and adds the calendar date to the volumetric budget in the Listing File and relevant package output options. Specifically, when calendar dates are specified in the HOB package, the HOB output file includes both the decimal year and the calendar date with each observation. When importing this date format into spread sheet programs (for example, Microsoft Excel®), the delimiter "T" may need to be removed for the date-time value to be recognized correctly. This can be done by doing a search and replace for the letter "T" with a blank space. (Alternatively, the Excel® "Convert Text to Columns Wizard" allows you to specify any character as a delimiter. By specifying "T" as a delimiter, Excel® reads the date and time as separate input data values.)

***A***

| Symbol | Representation |
|---|---|
| mmm | Three letter month or full name - Jan or January |
| mm | Two-digit month number - 01 or 1, with valid values being 1 to 12 |
| dd | Two-digit day of month number - 01 or 1, with valid values being 1 to 31 |
| yyyy | Four-digit year number - must be four digits - 1979, with valid values being 0000 to 9999 |
| | |
| **T** | Separator to indicate that a time of day is included after calendar date |
| | |
| hh | Two-digit hour, 24-hour format - 01 or 1, with range from 00 to 23 |
| mm | Two-digit minute of hour - 01 or 1, with range from 00 to 59 |
| ss | Two-digit second of minute - 01 or 1, with range from 00 to 59 |

***B***

| Date Format | Comment |
|---|---|
| mm/dd/yyyy | American style date |
| mmm/dd/yyyy | American style date in which month is represented by 3 letters |
| mm/yyyy | Automatically sets day to 1 |
| yyyy-mm-dd | ISO Standard |
| mm/dd/yyyy**T**hh:mm:ss | T separates calendar date from time |
| yyyy-mm-dd**T**hh:mm:ss | T separates calendar date from time |
| | |
| DYear | Decimal year - 1979.307, implies 4/23/1979 |

**Figure 3.5.** Input for Calendar Dates module in MF-OWHM2: *A*, calendar date symbology; *B*, calendar date formats accepted; and *C*, day of the year and fraction of a year for select dates during a non-leap or leap year. [The fraction of the year is determined by subtracting one from the day of the year and dividing it by the total number of days within the year.]

*C*

| Date (mmm-dd) | Day of Year | | Fraction of Year | |
|---|---|---|---|---|
| | Non-Leap | Leap | Non-Leap | Leap |
| Jan-01 | 1 | 1 | 0.000000 | 0.000000 |
| Jan-15 | 15 | 15 | 0.038356 | 0.038251 |
| Jan-31 | 31 | 31 | 0.082192 | 0.081967 |
| Feb-01 | 32 | 32 | 0.084932 | 0.084699 |
| Feb-28 | 59 | 59 | 0.158904 | 0.158470 |
| Feb-29 | — | 60 | — | 0.161202 |
| Mar-01 | 60 | 61 | 0.161644 | 0.163934 |
| Mar-15 | 74 | 75 | 0.200000 | 0.202186 |
| Mar-31 | 90 | 91 | 0.243836 | 0.245902 |
| Apr-01 | 91 | 92 | 0.246575 | 0.248634 |
| Apr-15 | 105 | 106 | 0.284932 | 0.286885 |
| Apr-23 | 113 | 114 | 0.306849 | 0.308743 |
| Apr-30 | 120 | 121 | 0.326027 | 0.327869 |
| May-01 | 121 | 122 | 0.328767 | 0.330601 |
| May-15 | 135 | 136 | 0.367123 | 0.368852 |
| May-31 | 151 | 152 | 0.410959 | 0.412568 |
| Jun-01 | 152 | 153 | 0.413699 | 0.415301 |
| Jun-15 | 166 | 167 | 0.452055 | 0.453552 |
| Jun-30 | 181 | 182 | 0.493151 | 0.494536 |
| Jul-01 | 182 | 183 | 0.495890 | 0.497268 |
| Jul-15 | 196 | 197 | 0.534247 | 0.535519 |
| Jul-31 | 212 | 213 | 0.578082 | 0.579235 |
| Aug-01 | 213 | 214 | 0.580822 | 0.581967 |
| Aug-15 | 227 | 228 | 0.619178 | 0.620219 |
| Aug-31 | 243 | 244 | 0.663014 | 0.663934 |
| Sep-01 | 244 | 245 | 0.665753 | 0.666667 |
| Sep-15 | 258 | 259 | 0.704110 | 0.704918 |
| Sep-30 | 273 | 274 | 0.745205 | 0.745902 |
| Oct-01 | 274 | 275 | 0.747945 | 0.748634 |
| Oct-15 | 288 | 289 | 0.786301 | 0.786885 |
| Oct-31 | 304 | 305 | 0.830137 | 0.830601 |
| Nov-01 | 305 | 306 | 0.832877 | 0.833333 |
| Nov-15 | 319 | 320 | 0.871233 | 0.871585 |
| Nov-30 | 334 | 335 | 0.912329 | 0.912568 |
| Dec-01 | 335 | 336 | 0.915068 | 0.915301 |
| Dec-15 | 349 | 350 | 0.953425 | 0.953552 |
| Dec-31 | 365 | 366 | 0.997260 | 0.997268 |

**Figure 3.5.** —Continued

## Discretization Package (DIS) Improvements

Several modifications were made to the DIS package to make it more representative of real-world time keeping. Specifically, it now has the ability to keep track of leap years, supports calendar dates, and can specify time-step lengths directly (rather than as a multiplier).

### Variable Time-Step Length

The DIS package was modified to allow the user to specify the exact time-step length. The time-step lengths are loaded on the same line as the stress-period information (`PERLEN NSTP TSMULT SS/TR`). This feature is initiated when the time-step count (`NSTP`) is specified as a negative number and the multiplier is set to `1`. Then, the absolute value of the time-step count represents the number of time-step lengths read to the right of the stress-period type (`SS/TR`), and the sum is the stress-period length (overwrites `PERLEN`). This allows the user to customize time-step lengths to match observation times or to create an acceleration factor that uses a more compact set of time steps (for example, `1`, `2`, `7`, `10`, `80` to accelerate to a total of `100` days). The compact numbering can be used to prevent simulation times with decimal parts by requesting time-step lengths to be whole numbers. This is particularly advantageous when the stress periods mimic calendar months, and the month can be broken into different counts of days, such as `31` days, and four time steps could have lengths of `7`, `8`, `8`, and `8` days.

Figure 3.6 is an example of DIS input that uses the variable time-step lengths, `TS_LEN`, for the odd-numbered stress periods. It only shows the stress period specification part of a larger DIS package that specified a time unit of days (`ITMUNI = 4`).

### Specifying Land or Ground-Surface Elevation

The DIS was modified to optionally load a land-surface elevation (LSE, sometimes called ground-surface elevation, GSE) array that represents the model-grid's surface elevations (a digital elevation model, or DEM). This allows a global surface-elevation grid to be accessed by other packages. To have the DIS load the surface elevation in model units, [length, L], the keyword **`SURFACE`** must be placed on the line after reading `DELC` and before `Top` (see the "Full DIS Input Instructions" section). If the keyword is not present, then global surface elevations are ignored and `Top` (Dataset 5b) is loaded instead. If loaded, the LSE array is read with the *Universal Loader* (`ULOAD`) described in appendix 1. Figure 3.7 shows a part of the DIS input for loading an LSE array. As of this report's publication, if global surface elevations are specified in the DIS package, the only package that uses them is the Farm Process (FMP). If it is specified in the DIS, then the ground-surface elevation array is not required to be specified as part of the FMP input; it automatically uses the DIS package array.

```
#  Example Basic DIS Package Input that illustrates using a Variable Time Step Length
#     Input assumes that BAS package includes the option "START_DATE  4/23/1979"
#
  2   3   4   3   4   2  # NLAY  NROW  NCOL  NPER  ITMUNI  LENUNI
  0   0          #  LAYCBD(NLAY)
  CONSTANT 100.  #  DELR(NCOL)
  CONSTANT 100.  #  DELC(NROWS)
  CONSTANT 500.  #  TOP(NCOL, NROW)
  CONSTANT 450.  #  LAY 1 BOTM(NCOL, NROW)
  CONSTANT 300.  #  LAY 2 BOTM(NCOL, NROW)
#  PERLEN  NSTP  TSMULT  ss/tr  [TS_LEN]
  8.000 -2  1     ss  3. 5.        # SP1  4/23/1979  to  5/1/1979
  31.00  2  1     tr               # SP2  5/1/1979  to  6/1/1979
  30.00 -4  1     tr  7. 7. 8. 8.  # SP3  6/1/1979  to  7/1/1979
  31.00  2  1     tr               # SP4  7/1/1979  to  8/1/1979
  31.00 -2  1     tr 14. 16.       # SP5  8/1/1979  to  9/1/1979
```

**Figure 3.6.** Example of DIS input that uses the variable time-step lengths, TS_LEN, for the odd-numbered stress periods. [The DIS input assumes that the BAS package includes "START_DATE 4/23/1979" in the option block.]

```
4.    DELC(NROW) - U1DREL

5a.   [SURFACE   ULOAD(NCOL,NROW)]

5b.   Top(NCOL,NROW) - U2DREL
```

**Figure 3.7.** Part of DIS input and read utilities for loading a land-surface elevation (LSE) array that represents the model-grid's surface elevations. The numbers 4., 5a., and 5b. represent the MODFLOW-2005 DIS package input item numbers.

## LAYCBD Keyword to Disable for All Layers

The DIS package requires loading a flag that indicates if a layer contains a quasi-three-dimensional (3D) confining bed beneath it. The input variable that defines this is called `LAYCBD`, and the user is required to specify its value as either `0` or `1` for each layer (`NLAY` layers) to indicate if it overlies the confining bed. Because the quasi-3D confining bed is a legacy feature of MODFLOW, and its use is limited—and not recommended—a new keyword is available that automatically sets `LAYCBD` to `0` for all layers. If the keyword **`NO_LAYCBD`** is placed where the DIS package expects to find `NLAY` integer flags for `LAYCBD`, then its value is automatically set to zero for all layers. Figure 3.8 is an example of the DIS input that shows the appropriate place to use the **`NO_LAYCBD`** keyword. Note that input items enclosed in brackets, [ ], are optional.

## ITMUNI (TIME) and LENUNI (LENGTH) Support Keywords

The DIS package uses two integer variables to define the simulation-model time and space units. The first variable is the flag `ITMUNI`, which defines the time unit, and the next is `LENUNI`, which defines the length unit. MODFLOW requires all input to be consistent with the specified values of these two variables. Most input packages use the letter T to indicate the time unit defined by `ITMUNI` and the letter L to indicate the length unit defined by `LENUNI`. The DIS package defines these two variables with an integer that represents the time and space units. The input has been extended to also check for use of a keyword instead of the integers. Figure 3.9 provides two lists of supported keywords that may be used in the place of the integer flags to indicate time and distance units of measure.

## Full DIS Input Instructions

The previous sections have described modifications to the DIS package. Each section presented the specific location of the input for the user to add that specific feature that changed. The full input is described here, with the modifications included, to provide a user reference for all of DIS input in one location. Comments are allowed on any line of input when they are preceded by a "`#`" character, except between lines of data loaded with *U1DREL* and *U2DREL* read utilities. Figure 3.10 provides examples of DIS input data items by line. Note that input items enclosed in brackets, [ ], are optional.

```
1. NLAY NROW NCOL NPER ITMUNI LENUNI [XFIRSTCORD YFIRSTCORD GRIDROTATION [COORD_OPTIONS]

2. LAYCBD(NLAY) or NO_LAYCBD

3. DELR(NCOL) - U1DREL
```

**Figure 3.8.** Example of DIS input that shows the appropriate place to use the **NO_LAYCBD** keyword. The numbers, 1., 2., and 3. represent the MODFLOW-2005 DIS package input item numbers.

***A***

| ITMUNI | Keyword Equivalent | |
|---|---|---|
| | Singular | Plural |
| 1 | **SECOND** | **SECONDS** |
| 2 | **MINUTE** | **MINUTES** |
| 3 | **HOUR** | **HOURS** |
| 4 | **DAY** | **DAYS** |
| 5 | **YEAR** | **YEARS** |

***B***

| LENUNI | Keyword Equivalent | |
|---|---|---|
| | Singular | Plural |
| 1 | **FOOT** | **FEET** |
| 2 | **METER** | **METERS** |
| 3 | **CENTIMETER** | **CENTIMETERS** |

**Figure 3.9.** Keywords in the DIS package that specify *A*, model time unit [T] ITMUNI; and *B*, model length unit [L] LENUNI. The keyword, either the singular or plural version, can be used in the place of the original DIS package ITMUNI and LENUNI integer flags. [DIS, Discretization package; [T], time unit used for all input and output during a simulation; [L], length unit used for all input and output during a simulation.]

***A***

1. NLAY NROW NCOL NPER ITMUNI LENUNI [XFIRSTCORD YFIRSTCORD GRIDROTATION [**COORD_OPTIONS**]]
2. LAYCBD(NLAY)
   The keyword **NO_LAYCBD**, may be used on line 2 instead of the integer values to set LAYCBD to zero for all layers
3. DELR(NCOL) - *U1DREL*
4. DELC(NROW) - *U1DREL*

5a. [**SURFACE** ULOAD(NCOL,NROW)]

5b. Top(NCOL,NROW) - *U2DREL*

6. BOTM(NCOL,NROW) - *U2DREL*
   Item 6 is repeated for each model layer and Quasi-3D confining bed in the grid.
   These bottom-of-layer variables are read in sequence going down from the top of the system.
   Thus, the number of BOTM arrays must be NLAY plus the number of Quasi-3D confining beds.
7. PERLEN NSTP TSMULT SS/TR [TS_LEN]
   Item 7 is repeated for each stress period (NPER).

**Figure 3.10.** Discretization (DIS) package input structure and explanation: *A*, DIS input data items by line; and *B*, explanation of the input variable names.

*B*

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.

NPER is the number of stress periods in the simulation.

ITMUNI indicates the time unit of model data, which must be consistent for all data values that involve time. All inputs that use the term "T" will use this as the unit for time. This can be either an integer flag or keyword. The keyword is the same as the name of the time unit. Using "Undefined" units is not recomended.

| Flag | Unit and Keyword |
|---|---|
| 0 | Undefined – Not a supported keyword |
| 1 | **SECOND** |
| 2 | **MINUTE** |
| 3 | **HOUR** |
| 4 | **DAY** |
| 5 | **YEAR** |

LENUNI indicates the length unit of model data, which must be consistent for all data values that involve length. All inputs that use the term "L" (or "$L^m$") will use this as the base unit for length [L], area [$L^2$], and volume [$L^3$]. This can be either an integer flag or keyword. The keyword is
the same as the name of the distance unit. Using "Undefined" units is not recomended.

| Flag | Unit and Keyword |
|---|---|
| 0 | Undefined – Not a supported keyword |
| 1 | **FEET** |
| 2 | **METER** |
| 3 | **CENTIMETER** |

XFIRSTCORD is the X Cartesian coordinate of the model cell center at Row 1, Column 1

YFIRSTCORD is the Y Cartesian coordinate of the model cell center at Row 1, Column 1

GRIDROTATION is the Polar angle of the model grid

**COORD_OPTIONS** the following are accepted keyword options for the coordinate system:

**LLCOORDINATE** is an optional keyword that indicates that XFIRSTCORD and YFIRSTCORD refer to the cell center of row NROW and column 1 (that is, the lower left corner).

**CORNERCOORD** is an optional keyword that indicates that XFIRSTCORD and YFIRSTCORD refer to the cell's outermost corner instead of the cell center.

**PRINTCOORD** is an optional keyword that prints the coordinate arrays to the list file.

LAYCBD is either a single keyword, **NO_LAYCBD**, or a list of NLAY integer flags that indicate if the layer overlies a Quasi-3D confining bed.
0 indicates no confining bed, and a non-zero value indicates a confining bed.
LAYCBD for the bottom layer must be 0.
The keyword **NO_LAYCBD** automatically sets LAYCBD to zero for all layers.

**Figure 3.10.** —Continued

***B** (continued)*

| | |
|---|---|
| DELR | is the cell width along rows. Input one value for each of the NCOL columns. This is a multi-value one-dimensional variable with one value for each model column. |
| DELC | is the cell width along columns. Input one value for each of the NROW rows. This is a multi-value one-dimensional variable with one value for each model row. |
| **SURFACE** | is a keyword that indicates the land surface elevation (LSE) is loaded with ULOAD and reads a NROW by NCOL two-dimensional array. The LSE must be greater than or equal to the elevations specified in Top. |
| Top | is the top elevation of layer 1. If layer 1 represents a water-table aquifer, then it may be reasonable to set Top equal to the land-surface elevation. That is, it may be the same array read by **SURFACE**. |
| BOTM | is the bottom elevation of a model layer or a Quasi-3D confining bed. |
| PERLEN | is the length of a stress period. |
| NSTP | is the number of time steps in a stress period. If NSTP < 0, then the time step length is specified and the absolute value, \|NSTP\|, represents the number of time step lengths that are defined as TS_LEN. |
| TSMULT | is the multiplier for the length of successive time steps. If NSTP < 0, then TSMULT is read but not used. If TSMULT is equal to 1, then the time step evenly divides PERLEN by NSTP. If TSMULT is negative, then its value is automatically set to 1. If TSMULT is not equal to 1, then the length of a time step is calculated by multiplying the length of the previous time step, that is, $\Delta t_i$ = TSMULT $\times \Delta t_{i-1}$. The length of the first time step, $\Delta t_1$, is calculated as $\Delta t_1 = \text{PERLEN}\left(\frac{\text{TSMULT}-1}{\text{TSMULT}^{\text{NSTP}}-1}\right)$. |
| TS_LEN | is only read if NSTP < 0 and represents a user specified time step length. if NSTP < 0, then \|NSTP\| time step lengths are read and PERLEN is set to their sum. |
| SS/TR | is a character variable that indicates if the stress period is transient or steady state. The only allowed options are "SS" and "TR" but these are case insensitive |

**Figure 3.10.** —Continued

## New Basic Package (BAS) Options

The basic package was modified to include a set of new options in the basic package **OPTIONS** block that alter the approach to model runs in MF-OWHM2. Some of the new options are **START_DATE**, **FASTFORWARD**, **INPUT_CHECK**, **BUDGETDB**, **NOCBC**, **NOCBCPACK**, **CBC_EVERY_TIMESTEP**, **PRINT_CONVERGENCE**, **PRINT_FLOW_RESIDUAL**, **PRINT_RELATIVE_VOLUME_ERROR**, **NO_DIM_CHECK**, **DEALLOCATE_MULT**, **TIME_INFO**, and **NO_FAILED_CONVERGENCE_STOP**. The primary new options are listed with short descriptions in figure 3.11 and are discussed in the sections that follow.

*A*

| Keyword | Description |
|---|---|
| **CBC_EVERY_TIMESTEP** | Indicates that the CBC is written to for every time step by all packages that have a non-zero cell-by-cell file specified. |
| **CBC_LAST_TIMESTEP** | Indicates that the CBC is written to at the last time step for each stress period by all packages that have a non-zero cell-by-cell file specified. |
| **CHTOCH** | Flow between adjacent constant-head cells should be calculated. |
| **DEALLOCATE_MULT** | Deallocate MULT package arrays once they are no longer required. |
| **DOUBLE_PRECISION_CBC** | Write CBC using double precision numbers instead of single precision. |
| **FREE** | Free format input for MODFLOW packages (option enabled by default). |
| **INPUT_CHECK** | Run simulation without the solver to check input and output files. |
| **NO_DIM_CHECK** | Skip length, width, thickness, and volume checks in the BAS and DIS. This improves speed for model grids that are known to be error free. |
| **NO_FAILED_CONVERGENCE_STOP** | Simulation continues, even if convergence fails. Same as the BAS option "**STOPERROR** 1E30" |
| **NOCBC** | Disables writing to the cell-by-cell file for all packages. |
| **NOCBCPACK** | Disables writing to the cell-by-cell file for all packages except for the main flow package. |
| **NOFREE** | Fixed-format input for supported MODFLOW packages. This is necessary for legacy MODFLOW models that rely on fixed-format input, which in no longer the default option. |
| **PAUSE** | Requires the user to press enter at the end the simulation. |
| **XSECTION** | Indicates the model is a 1-row cross section for which STRT and IBOUND should each be read as single two-dimensional variables with dimensions of NCOL and NLAY. |

**Figure 3.11.** Complete list and descriptions of supported basic (BAS) package options: *A*, Options that do not have any arguments; *B*, Options that require arguments; and *C*, Options that are associated with writing output information. [The keyword for the desired option must be specified in the BAS OPTIONS block, one per line, and if there is an additional, required, argument, then it is specified in the argument column. Options and arguments must be specified on the same line in the BAS OPTIONS block.]

*B*

| Keyword | Argument | Description |
|---|---|---|
| **DAMPEN_START** | ITER DMP | Apply a dampening factor (DMP) to the first ITER outer-solver iterations of the simulation.<br>Solver convergence is disabled during the first ITER outer-solver iterations.<br>This is useful if the initial conditions are unstable and result in a floating point overflow during the first time step of the simulation.<br>ITER is specified as an **INT** and DMP as a **FLOAT**. Recommended values of DMP are 0.1–0.5. |
| **FASTFORWARD** | SPSTART [SPSTOP] | Run simulation from stress period SPSTART to SPSTOP, by reading each stress period input until SPSTART.<br>The initial conditions defined in the BAS package are used as the initial condition at SPSTART.<br>If SPSTOP is not specified, then it is automatically set to the end of the simulation.<br>SPSTART and SPSTOP are specified either as an integer, to indicate the stress period number, or as a date that is contained by one of the stress periods. If set to a date, then the stress period that contains the date is the starting or ending stress period. |
| **HEAD_DISTANCE_ABOVE_LSE_LIMIT** | ABOVE_LSE_LIM | Imposes an upper limit to the head solution generated from the solver. The limit is a distance above the user specified land surface elevation (LSE). If LSE is not specified, then the TOP of the first layer is used instead. If the solver calculates a head value greater than the distance above the LSE, then the head is set to the elevation that distance represents.<br>ABOVE_LSE_LIM, specified as a **FLOAT**, is the max distance above LSE or TOP that the solver calculated head solution can be.<br>This is useful for models that have floating point overflow errors. |

**Figure 3.11.** —Continued

***B** (continued)*

| Keyword | Argument | Description |
|---|---|---|
| **MAX_RELATIVE_VOLUME_ERROR** | MAX_RVOL_ERROR | MAX_RVOL_ERROR is the maximum allowed relative volume error specified as a **FLOAT**.<br>If not specified, then the default is 0.025<br>Set to large value to disable additional convergence check. |
| **MAXBUDGET** | MXBUD | Specify the maximum number of budget groups allowed. If not specified, then MXBUD is set to 100.<br>Increase MXBUD to a greater value number when an error is raised because there is not enough memory to hold all budget groups.<br>To save on memory you may reduce MXBUD to the number of packages in use plus 10. |
| **MAXPARAM** | MXPAR MXCLST MXINST | Specify maximum number of parameters (MXPAR), parameter clusters (MXCLST), and parameter instances (MXINST).<br>Default is MXPAR=2000, MXCLST=2000000, MXINST=50000 |
| **MIN_SOLVER_ITERATION** | MIN_SOLV_ITER | Require that each time step solve at least MIN_SOLV_ITER outer-solver iterations.<br>This is helpful if solver converges too soon, resulting in mass balance errors. |
| **PERCENTERROR** | PDIFFPRT | Specifies the minimum rate percent error at the end of each time step before a warning is raised and the volumetric budget is printed.<br>If PDIFFPRT is set to 0 then a warning is raised every time step. If not specified, then defaults to 5 percent. |

**Figure 3.11.** —Continued

***B** (continued)*

| Keyword | Argument | Description |
|---|---|---|
| **SHIFT_STRT** | ULOAD | Shift the initial conditions (STRT) for each layer.<br>Input uses the *Universal Loader* (ULOAD) with *List Style* input to read NLAY **FLOAT** numbers that are added to STRT.<br>For a three-layer model (NLAY=3), the following example<br>Adds 9 to the layer 1's STRT array,<br>Adds –7 to the layer 2's STRT array,<br>Adds 0 to the layer 3's STRT array.<br>`SHIFT_STRT INTERNAL`<br>`1 9.0`<br>`2 -7.0`<br>`3 0.0` |
| **START_DATE** | SIM_START_DATE | Specifies the starting date of the simulation.<br>This keyword is required if any input uses calendar dates.<br>This date is a reference for all calendar date input and is used to determine each time step's starting and ending date and time.<br>SIM_START_DATE is specified using any of the MF-OWHM2 accepted date formats that include.<br>Examples that specify April 23rd, 1979, as the starting date:<br>`START_DATE 4/23/1979`<br>`START_DATE 1979-4-23`<br>Examples that include the 24hr clock time for 11:15 PM:<br>`START_DATE 4/23/1979T23:15`<br>`START_DATE 1979-4-23T23:15`<br>Examples that set the seconds to 30:<br>`START_DATE 4/23/1979T23:15:30`<br>`START_DATE 1979-4-23T23:15:30` |
| **STOPERROR** | STOPER | STOPER is a percent error that is compared to the budget percent discrepancy if the solver convergence criteria are not met. If the percent error is less than STOPER, then the simulation does not stop.<br>If not specified, then STOPER is set to 0.0 percent.<br>Note that the option **NO_FAILED_CONVERGENCE_STOP** is equivalent to "**STOPERROR** 1E30" |

**Figure 3.11.** —Continued

*C*

| Keyword | Argument | Description |
|---|---|---|
| **BUDGETDB** | *Generic_Output_OptKey* | Write every time step's groundwater-flow budget in a spreadsheet (database) friendly format.<br>Output contains same information as in the LIST file volumetric rate budget. |
| **CUMULATIVE_RESIDUAL_ERROR_ARRAY** | *Generic_Output_OptKey* | Write at the end of the simulation the cumulative residual error for every model cell. |
| **HEAD_DISTANCE_ABOVE_LSE_PRINT** | ABOVE_LSE_PRNT<br>*Generic_Output_OptKey* | Write to a file any model cell that has a head elevation greater than ABOVE_LSE_PRNT plus the user-specified land surface elevation (LSE). If LSE is not specified, then the TOP of the first layer is used instead.<br>ABOVE_LSE_PRNT is specified as a **FLOAT**. |
| **ITERATION_INFO** | *Generic_Output_OptKey* | Write every time step's the final solver information. |
| **PRINT_CONVERGENCE** | NTERM OUTER_START<br>*Generic_Output_OptKey* | After the OUTER_START solver iterations, write the NTERM cells that had the worst solver change in head to a file. Irrelevant of OUTER_START, the last solver iteration is always printed.<br>If OUTER_START = 0, then only print the last solver iteration.<br>If OUTER_START < 0, then print the last MXITER − OUTER_START iterations. |
| **PRINT_FLOW_RESIDUAL** | NTERM OUTER_START<br>*Generic_Output_OptKey* | After the OUTER_START solver iterations, write the NTERM cells that had the worst flow residual to a file. Irrelevant of OUTER_START, the last solver iteration is always printed.<br>If OUTER_START = 0, then only print the last solver iteration.<br>If OUTER_START < 0, then print the last MXITER − OUTER_START iterations. |

**Figure 3.11.** —Continued

***C** (continued)*

| Keyword | Argument | Description |
|---|---|---|
| **PRINT_RELATIVE_VOLUME_ERROR** | NTERM OUTER_START *Generic_Output_OptKey* | After the OUTER_START solver iterations, write the NTERM cells that had the worst relative volume error to a file. Irrelevant of OUTER_START, the last solver iteration is always printed.<br>If OUTER_START = 0, then only print the last solver iteration.<br>If OUTER_START < 0, then print the last MXITER − OUTER_START iterations. |
| **RESIDUAL_ERROR_ARRAY** | *Generic_Output_OptKey* | Write at the end of each time step the residual error for every model cell. |
| **RESIDUAL_ERROR_ARRAY_THRESHOLD** | PRNT_RES_LIM | If specified, then **RESIDUAL_ERROR_ARRAY** only writes residual errors for time steps that have a rate percent error greater than PRNT_RES_LIM rate percent error. |
| **TIME_INFO** | *Generic_Output_OptKey* | Write each model step's time step length, its starting calendar date, starting decimal year and simulation time to a file. This is useful for post-processing tools. |

**Figure 3.11.** —Continued

## FASTFORWARD—Simulation Time-Frame Adjustments (BAS)

The keyword **FASTFORWARD** is available with the BAS **OPTIONS** block to specify a simulation time window that is a subset of the current stress-period set up. An example of a simulation time-frame adjustment is for a model that is developed with 10 stress periods, but only the simulation of stress-periods 5 through 7 are of interest. The **FASTFORWARD** option would cycle through the input files until reaching stress-period 5, then run the simulation until stress-period 7 completes. This allows analysis of shortened time frames without having to rebuild all the input datasets. Another advantage of this feature is to allow calibration of different simulation-time windows without having multiple copies of the model.

To initiate a **FASTFORWARD**, the keyword must be located in the BAS **OPTIONS** block followed by the starting and ending stress period. If a **START_DATE** is specified as a BAS option ("Calendar Dates" section), then a starting and ending calendar date may be specified for **FASTFORWARD**, which is then translated to the appropriate starting and ending stress periods.

If the **FASTFORWARD** feature is enabled, the simulation rolls forward until the starting stress period is found. The initial conditions specified in the simulation (for example, the BAS STRT variable) are propagated forward and used as the initial conditions at the requested starting stress period. This requires the initial heads to be set to the initial condition for the selected **FASTFORWARD** starting stress period. Figure 3.12 shows the **OPTIONS** block of BAS with the **FASTFORWARD** option. When SPSTART and SPSTOP are set as calendar dates, MF-OWHM2 searches for the first stress period that contains that date. Figure 3.13 presents the search algorithm used by MF-OWHM2 to discern the starting and ending stress periods when they are specified as calendar dates.

Figure 3.14 provides an example of using the **FASTFORWARD** feature for a simulation model. The example presumes that input to the BAS package **START_DATE** option specifies the starting date of 1/1/2000 and there are 12 monthly stress periods (that is, each stress-period length is equal to the length of the respective month; for example, January has 31 days or 744 hours, or 44,640 minutes, or 2,678,400 seconds).

*A*

```
BEGIN  OPTIONS
          #
          FASTFORWARD SPSTART  [SPSTOP]
          #
END
```

*B*

| | |
|---|---|
| **FASTFORWARD** | is a BAS option to specify a simulation-time frame. |
| SPSTART | is the simulation starting point that is specified either as a calendar date or a stress-period number. |
| SPSTOP | optional; is the simulation stopping point that is specified either as a calendar date or stress-period number. If not specified, then SPSTOP is the total number of stress periods (NPER). |

**Figure 3.12.** BAS package **OPTIONS** block with **FASTFORWARD** keyword: *A*, example of structure; and *B*, input description.

*A*

```
IF        DATE < STARTDATE:
                              SP = 1
ELSE IF   DATE > SPDATE(NPER):
                              SP = NPER
ELSE Search for SPDATE(I) <= DATE < SPDATE(I+1):
                              SP=I
```

*B*

| | |
|---|---|
| DATE | is SPSTART or SPSTOP in the format of a calendar date. |
| STARTDATE | Starting date of simulation defined by BAS option **START_DATE**. |
| SPDATE(I) | Date at the start of the $I^{th}$ stress period. |
| NPER | Total number of stress periods (as specified in the DIS package). |
| SP | Selected stress period number. |

**Figure 3.13.** Algorithm for finding first and last stress periods used by the **FASTFORWARD** feature: *A*, the search algorithm used when the **FASTFORWARD** simulation period (values of SPSTART and SPSTOP) is specified using calendar dates; and *B*, explanation of variables used in the search algorithm.

```
#  START_DATE = 1/1/2000
#  NPER = 12 stress periods; each stress period represents 1 month.
#  Stress period length is equal to the number of days in the month it represents
#  Start on stress period 3 and end the simulation at NPER (which is stress period 12)
FASTFORWARD   3

# Start on stress period 3 and end the simulation at NPER (which is stress period 12)
# This is because stress period 3 is from 3/1/2000 through 3/31/2000
FASTFORWARD   3/1/2000

# Start on stress period 3 and end on stress period 5
FASTFORWARD   3    5

# Start on stress period 3 and end on stress period 5
FASTFORWARD   3/1/2000    5/1/2000
```

**Figure 3.14.** Examples of using the **FASTFORWARD** feature for a simulation model. [Examples assume that the simulation starting date is 1/1/2000 and there are 12 monthly stress periods.]

## Input_Check—Cycling Through All Input Files, BAS Option

Similar to the **FASTFORWARD** option, the BAS package now includes the keyword **INPUT_CHECK**. It results in a simulation cycling through all input files. Essentially, this provides a fast method to check if there are any input errors by allowing a simulation run to cycle through the input files without solving the groundwater flow or any process equations. This is especially useful for long simulations that might otherwise take a long time to reach an input section that has an error.

## BUDGETDB—Budget Information Written to Separate Database Friendly File

Previously, the budget information was written to the LIST file (or WBGT file) when requested by the Output Control (OC) package. To obtain budget information independent of the OC package, the BAS option, **BUDGETDB**, has been added. When this option is invoked, the budget information is written to a *Generic_Output* file (appendix 1). The *Generic_Output* file can be either formatted text or binary format.

The **BUDGETDB** file formatted as text includes the following fields on the header for the output from each time step: DATE_START, PER, STP, DELT, SIMTIME, STORAGE_IN, STORAGE_OUT, CHD_IN, CHD_OUT, and so on. The header continues the listing of each package's name with _IN and _OUT appended to its abbreviation. The DATE_START is the calendar date at the start of the time step in the form of yyyy-mm-ddThh:mm:ss (for example, 1979-4-23T21:21:00). The T is used as a separator between the calendar date and 24-hour clock time. The PER and STP are the stress-period and time-step numbers, respectively. The budget output in text format facilitates further analysis with external software such as a database system or spreadsheet.

If the **BUDGETDB** is written to a binary file, then it has a slightly different format. The format is printed in the LIST file and as the first record in the binary file. The first record is a single Fortran Integer, which is a "count" of the number of columns in the database. Then there are "count" text strings of 16 characters length that define the column names. Similar to the text version, the first set of column names are DATE_START, PER, STP, DELT, and SIMTIME. After SIMTIME, the remaining column names depend on which packages are used for the simulation. The binary file does not include the headers for total input and output (IN_OUT) or percent error (PERCENT_ERROR), which are in the text version. After the "count" lines of text strings, the remaining records, one for each time step, contain only the information for the actual budget record defined by the column names. DATE_START is 19 characters long and uses the same format as the text version; PER and STP are integer variables, and the remaining numerical values are Fortran single precision. The numerical values are changed to double precision if the BAS option **DOUBLE_PRECISION_CBC** is included.

## NOCBC and NOCBCPACK—Turn Off Cell-By-Cell Writing (CBC)

Most packages offer a Fortran unit number to write CBC budget information. This file is then used by post processing programs, such as zonebudget, to analyze simulation results. During calibration, the CBC output is often not required. To deactivate CBC output when using versions previous to MF-OWHM2, the input files all had to be modified to set each package's CBC value to zero. This was done to reduce simulation run time and avoid the input and output (I/O) for the excessively large CBC file. There is an increased potential for user-input error when setting all the CBC values to zero and then subsequently resetting them back to the correct unit number. To prevent this, the user can leave the CBC Fortran file unit number in all the packages and add the BAS option keyword **NOCBC**, which automatically suppresses all writing to the CBC file. If the user wishes to only have the flow packages (for example, LPF, UPW, HUF) write to the CBC, then the BAS option keyword **NOCBCPACK** can be used.

## CBC_EVERY_TIMESTEP—Turn On Cell-By-Cell Writing (CBC)

The converse of the BAS option keyword **NOCBC** is the keyword **CBC_EVERY_TIMESTEP**. This keyword forces writing to the CBC at the end of every time step for all packages that have a non-zero Fortran unit specified for their CBC output. The advantage of this is it is not necessary to specify the keyword **SAVE BUDGET** in the OC for every model time step. Alternatively, the BAS keyword **CBC_LAST_TIMESTEP** is available to write to the CBC for the last time step of each stress period.

## Obtaining Solver Information to External File

For nonlinear models, it is advantageous to see how the solver proceeds to convergence to identify model cells that are highly nonlinear and oscillate. For details on the convergence criteria, please review the "Additional Convergence Metric" section, which gives an overview of the different solver convergence metrics. The keywords **PRINT_CONVERGENCE**, **PRINT_FLOW_RESIDUAL**, **PRINT_RELATIVE_VOLUME_ERROR**, **ITERATION_INFO**, **RESIDUAL_ERROR_ARRAY**, and **CUMULATIVE_RESIDUAL_ERROR_ARRAY** allow writing specific convergence information to separate output files. The keywords determine the convergence criteria that are to be written. **PRINT_CONVERGENCE** writes information on how the head changes for each solver iteration. **PRINT_FLOW_RESIDUAL** writes the solver flow residual error (which is compared against RCLOSE). **PRINT_RELATIVE_VOLUME_ERROR** writes the relative-volume error changes for each solver iteration. **ITERATION_INFO** writes the number of outer iterations required by the solver for each time step and the rate error, volume error, and percent error at the end of the time step. **RESIDUAL_ERROR_ARRAY** writes the flow residual error for each model layer at the end of each time step. **CUMULATIVE_RESIDUAL_ERROR_ARRAY** writes the cumulative flow residual error for each model layer at the end of the simulation.

The advantage of these output options is the user can investigate the model cells with large changes for potential input errors or questionable conceptual development for time steps that fail to converge. Common causes of a time step failure to converge are a model cell that is too thin (small vertical thickness), SFR segments with conductance values that are too large, or cells that incur the so-called MODFLOW "Wet-Dry" problem within a MF-OWHM2 simulation. Knowing which cells cause the failed convergence gives the user a point of reference to investigate. Figure 3.15 shows the input format to include the additional convergence output files. Note, all the options are presented for completeness, but one or any combination may be used.

Figure 3.16 lists and defines the header items included in the output files. Figure 3.16*A* defines formatted text headers for **PRINT_CONVERGENCE**. Figure 3.16*B* shows the structure of each binary record if the **PRINT_CONVERGENCE** is accompanied by specifications to write a binary file. Figure 3.16*C* lists and defines the header items included in the output file that results from specifying **PRINT_FLOW_RESIDUAL**. Figure 3.16*D* lists and defines the header items included in the output file that results from specifying **PRINT_RELATIVE_VOLUME_ERROR**.

***A***

```
BEGIN  OPTIONS
   #
   PRINT_CONVERGENCE                NTERM OUTER_START  Generic_Output
   #
   PRINT_FLOW_RESIDUAL              NTERM OUTER_START  Generic_Output
   #
   PRINT_RELATIVE_VOLUME_ERROR      NTERM OUTER_START  Generic_Output
   #
   ITERATION_INFO                   Generic_Output
   #
   RESIDUAL_ERROR_ARRAY             Generic_Output
   #
   CUMULATIVE_RESIDUAL_ERROR_ARRAY  Generic_Output
END
```

***B***

| | |
|---|---|
| NTERM | is the number of cells to print for each outer iteration. Each outer iteration then prints NTERM model cells that had the largest change since the previous iteration. |
| OUTER_START | is the solver iteration to begin printing the NTERM cells. If set to zero, then it only prints the final-converged or last-solver iteration. If set to a negative number, then it prints either final-converged iteration or iterations after the maximum solver iterations minus the OUTER_START plus one. |
| *Generic_Output* | is the file written to. |

**Figure 3.15.** BAS package options for convergence related output file: *A*, example of **OPTIONS** block with keywords; and *B*, explanation of input variables used by the keywords.

***A***

| | |
|---|---|
| SP | is the stress-period number. |
| TS | is the time-step number. |
| ITER | is the outer iteration number |
| LAY | is the layer of the head value. |
| ROW | is the row of the head value. |
| COL | is the column of the head value. |
| HEAD | is the solver calculated head for the outer iteration, ITER. |
| CHNG_HEAD | is the change in head between the current solver iteration, ITER, and the previous iteration, ITER – 1.<br>That is, CHNG_HEAD = HEAD(ITER) - HEAD(ITER-1) |
| DATE | is the month and year of the time step. |
| CELL_ID | is the model cell's unique identifier (ID). The cell ID is useful for re-sorting the file to look at sequential changes in data for a specific model cell.<br>The ID is determined by the following formula:<br>$CELL\_ID = COL + NCOL * (ROW - 1) + NCOL * NROW * (LAY - 1)$ |

***B***

| | |
|---|---|
| STP | INTEGER |
| TS | INTEGER |
| ITER | INTEGER |
| LAY | INTEGER |
| ROW | INTEGER |
| COL | INTEGER |
| HEAD | DOUBLE |
| CHNG_HEAD | DOUBLE |
| DATE | CHARACTER(8), starting date of time step as "mmm-yyyy" |
| ID | INTEGER |

**Figure 3.16.** Lists and descriptions of header items included in the output files that result from specifying *A*, **PRINT_CONVERGENCE** with formatted text; *B*, **PRINT_CONVERGENCE** with binary formatting; *C*, **PRINT_FLOW_RESIDUAL**; and *D*, **PRINT_RELATIVE_VOLUME_ERROR**.

***C***

| | |
|---|---|
| SP | is the stress-period number. |
| TS | is the time-step number. |
| ITER | is the outer iteration number |
| LAY | is the layer of the head value. |
| ROW | is the row of the head value. |
| COL | is the column of the head value. |
| HEAD | is the solver calculated head for the current solver iteration. |
| FLOW_RESIDUAL | is the flow residual error for the current solver iteration. |
| VOL_RESIDUAL | is the flow residual error times the time-step length to yield the volume residual for the current solver iteration. |
| CELL_VOLUME | is volume of the model cell at LAY, ROW, COL. |
| DATE | is the month and year of the time step. |
| CELL_ID | is the model cell's unique identifier (ID). The cell ID is useful for re-sorting the file to look at sequential changes in data for a specific model cell. The ID is determined by the following formula: $CELL\_ID = COL + NCOL * (ROW - 1) + NCOL * NROW * (LAY - 1)$ |

***D***

| | |
|---|---|
| SP | is the stress-period number. |
| TS | is the time-step number. |
| ITER | is the outer iteration number |
| LAY | is the layer of the head value. |
| ROW | is the row of the head value. |
| COL | is the column of the head value. |
| HEAD | is the solver calculated head for the current solver iteration. |
| REL_VOL_ERR | is the relative volume error. |
| VOL_RESIDUAL | is the flow residual error multiplied by the time-step length to yield the volume residual for the current solver iteration. |
| FLOW_RESIDUAL | is the flow residual error for the current solver iteration. |
| DATE | is the month and year of the time step. |
| CELL_ID | is the model cell's unique identifier (ID). The cell ID is useful for re-sorting the file to look at sequential changes in data for a specific model cell. The ID is determined by the following formula: $CELL\_ID = COL + NCOL * (ROW - 1) + NCOL * NROW * (LAY - 1)$ |

**Figure 3.16.** —Continued

## NO_DIM_CHECK—Bypass Warning for Thin Model Cells

The keyword **NO_DIM_CHECK** causes the BAS package to not check for thin or narrow cells in a model. A thin cell is one that has a thickness five orders of magnitude smaller than the thickest cell of the same model layer. The "dimensions check" further looks at DELR and DELC—width and length of cells—for the same five order of magnitude difference in those cell dimensions in the same column or row. If the **NO_DIM_CHECK** keyword is present in the **OPTIONS** block, this check is bypassed, but MF-OWHM2 still checks for negative thicknesses, widths, and lengths of model cells and stops the simulation if it finds a negative value.

## DEALLOCATE_MULT—Reduce MULT Package Memory

The MULT package allocates a large number of arrays that are only in use when constructing a MODFLOW Parameter (or Instance). Once the MULT arrays have been used for their initial purpose, they are no longer used for the rest of the simulation. To reduce the MF-OWHM2 memory (RAM) foot print, the MULT arrays can be released from memory after they have been used. This is done with the BAS package keyword **DEALLOCATE_MULT**. This option is only useful if the MULT package is in use and defines more than 100 multiplier arrays, which results in substantial memory savings.

## TIME_INFO—External File of All Time Step Times

The legacy MODFLOW-2005 output files only include the stress period and time step and may additionally include the time-step length and total simulated time. MF-OWHM2 simulations that specify a starting date in the DIS package include both a decimal year and calendar date as part of the time step time. For post processing convenience, the model time-step's ending simulation time, date, and time-step length can all be printed to an external file. This file can serve both to check that the time discretization is set up correctly and for post-processors to have a look up table of stress period and time step to a corresponding date and decimal year. If a starting date is not specified, only the simulated time is printed.

Figure 3.17 shows the input format to print the time information and describes the new variables. Figure 3.18 lists and describes the header items included in the output file that results from specifying **TIME_INFO**.

```
BEGIN  OPTIONS
              TIME_INFO   Generic_Output
END
```

**Figure 3.17.** Structure of BAS package input used to print the time information to an external file. [*Generic_Output* specifies the location where the output is written.]

| | |
|---|---|
| STEP | is a sequential number that represents the total number of model time steps solved. |
| SP | is the stress-period number. |
| TS | is the time-step number. |
| DELT | is the time-step length. |
| SIMTIM | is the total simulated time at the end of the time step. |
| DYEAR | is the decimal year at the end of the time step. |
| DATE | is the date at the end of the time step formatted as yyyy-mm-ddThh:mm:ss. |

**Figure 3.18.** BAS package option TIME_INFO output file header description.

## Budget_Groups—Splitting a Package Budget Information into Subgroups

The MODFLOW-2005 "Volumetric Budget" output (and cell-by-cell file) for each package includes the total volumetric flow rate in and out between the package and the groundwater flow equation. Only allowing for a single volumetric budget per package can be limiting if a package serves multiple purposes, such as simulating coastal and inland boundary conditions. Several packages were modified to allow the user to define Budget Groups that allow for multiple volumetric budgets per package. The groups contain a unique name that takes the place of the package name in the List file's Volumetric Budget (and in the CBC file), allowing for a more detailed accounting of how the flows for each of the packages interact with groundwater flow. The Budget Group feature is currently available for the MNW2, RIP, WEL, GHB, DRN, DRT, and RIV packages.

The Budget Group input requires two modifications to the standard input file. The first is to declare a **BUDGET_GROUPS** block using *Block Style* input (appendix 1). The **BUDGET_GROUPS** defines all the Budget Group names (BGROUPs). Each BGROUP defined in the block is then printed in the Volumetric Budget. The second modification is to associate all package features with one of the BGROUPs defined in the **BUDGET_GROUPS** block.

The **BUDGET_GROUPS** block is defined at the beginning of the package input file following the optional **PARAMETER** keyword. If there are other input blocks—such as **OPTIONS** or **LINEFEED**—placed after **PARAMETER**, the block order does not matter. In the **BUDGET_GROUPS** block, all budget group names, BGROUP, must be defined using one name per line. If the block is empty, then MF-OWHM2 ignores the **BUDGET_GROUPS** block and proceeds with the default MODFLOW budget name. Figure 3.19 is a list of the packages that **BUDGET_GROUPS** are available for and the location of the **BUDGET_GROUPS** block in the package.

The general input structure for the **BUDGET_GROUPS** block is shown in figure 3.20. The budget group names, BGROUPs, can be any alpha-numeric phrase, up to 16 characters long. This can be repeated for all the potential budget group names. For ease of handling output files, it is recommended, but not required, to keep the budget groups names shorter than 12 characters long. If there are numerous BGROUPs, then the block may specify a single *Generic_Input* (appendix 1) that points to a file containing a list of all the BGROUPs.

| Package | Located after | Located before |
|---|---|---|
| DRN | **PARAMETER** | MXACTD |
| DRT | # Text – optional comment header | MXADRT |
| GHB | **PARAMETER** | **TABFILE** |
| MNW2 | # Text – optional comment header | **TABFILE** |
| RIP | # Text – optional comment header | MAXRIP |
| RIV | **PARAMETER** | MXACTR |
| WEL | **PARAMETER** | **TABFILE** |

**Figure 3.19.** List of the packages for which **BUDGET_GROUPS** are available and the location of the **BUDGET_GROUPS** block in the package.

```
BEGIN BUDGET_GROUPS
    # List Budget Names   -- Comments are allowed within the block
    BGROUP_1      # First      budget group name
    BGROUP_2      # Second  budget group name
      ⋮
    BGROUP_N      # Repeat as needed
END
```

**Figure 3.20.** General input structure for the **BUDGET_GROUPS** block. The word BGROUP_X, where X is replaced by a number from 1 to N, represents unique budget group name (BGROUP) that is up to 16 characters long. [⋮ is a place holder for the third through the N – 1 budget group names.]

If Budget Groups are in use (that is, the **BUDGET_GROUPS** block exists and contains at least one BGROUP), then every package feature must be associated with a BGROUP. For example, if the General Head Boundary (GHB) package contains the **BUDGET_GROUPS** block with at least one BGROUP, then every GHB cell must be associated with a BGROUP. The location for specifying the package feature BGROUP association depends on the package. Figure 3.21 presents the input location for BGROUP for each of the supported packages. Figure 3.22 is an example GHB package input—with and without budget groups—and associated example LIST file volumetric budget.

***A***

```
2a.    WELLID  NNODE  BGROUP
```

***B***

```
7.     Layer Row Column NPOLY BGROUP
```

***C***

```
4b.    Layer Row Column Qfact BGROUP [xyz] [TABNAM TSFAC [TABEXP]]
6.     Layer Row Column Q     BGROUP [xyz] [TABNAM TSFAC [TABEXP]]
```

***D***

```
4b.  Layer Row Column Bhead CondFact BGROUP [xyz] [TABNAM TSFAC [TABEXP]]
6.   Layer Row Column Bhead Cond     BGROUP [xyz] [TABNAM TSFAC [TABEXP]]
```

***E***

```
3b. Layer Row Column Elevation CondFact BGROUP [xyz]
5.  Layer Row Column Elevation Cond     BGROUP [xyz]
```

***F***

```
3b. Layer Row Column Elevation CondFact [LayR RowR ColR Rfprop] BGROUP [xyz]
5.  Layer Row Column Elevation Cond     [LayR RowR ColR Rfprop] BGROUP [xyz]
```

***G***

```
4b. Layer Row Column Stage CondFact Rbot BGROUP [xyz] [TABNAM TSFAC [TABEXP]]
6.  Layer Row Column Stage Cond     Rbot BGROUP [xyz] [TABNAM TSFAC [TABEXP]]
```

**Figure 3.21.** Packages that support budget groups and the associated input location that identifies the specific input feature with a budget group name (BGROUP). The supported packages are *A*, MNW2 package; *B*, RIP package; *C*, WEL package; *D*, GHB package; *E*, DRN package; *F*, DRT package; *G*, RIV package. [The number at the start of each input line—such as "2a." or "7."—is the MODFLOW-2005 input data item number.]

***A***

```
# General-Head Boundary package (GHB) for model with NLAY, NROW, NCOL = 1, 5, 2, respectively.
#
# MODFLOW-2005 default budget name is "HEAD DEP BOUNDS"
4     0     # MXACTC  IGHBCB
4     0     # ITMP  NP                        -> Stress period 1
 1   1   1    -5.0   100.0                    # LAY  ROW  COL  Bhead  Cond  BGROUP
 1   1   2    -5.0   100.0
 1   5   1   -10.0   100.0
 1   5   2   -10.0   100.0
```

***B***

```
# General-Head Boundary package (GHB) for model with NLAY, NROW, NCOL = 1, 5, 2, respectively.
#
# Declare Budget Groups – If not declared, MODFLOW-2005 default budget name is "HEAD DEP BOUNDS"
BEGIN BUDGET_GROUPS
                    GHB_OCEAN     # BGROUP_1
                    GHB_INLAND    # BGROUP_2
END
4     0     # MXACTC  IGHBCB
4     0     # ITMP  NP                        -> Stress period 1
 1   1   1    -5.0   100.0  GHB_OCEAN        # LAY  ROW  COL  Bhead  Cond  BGROUP
 1   1   2    -5.0   100.0  GHB_OCEAN
 1   5   1   -10.0   100.0  GHB_INLAND
 1   5   2   -10.0   100.0  GHB_INLAND
```

***C***

```
-----------------------------------------------------------------------------
 VOLUMETRIC BUDGET FOR ENTIRE MODEL AT END OF TIME STEP 1 IN STRESS PERIOD 1
-----------------------------------------------------------------------------

    CUMULATIVE VOLUMES      L**3       RATES FOR THIS TIME STEP      L**3/T
    ------------------                 ------------------------

          IN:                                IN:
          ---                                ---
            STORAGE =     57236.2616             STORAGE =      1846.3310
      CONSTANT HEAD =         0.0000       CONSTANT HEAD =         0.0000
    HEAD DEP BOUNDS =     20717.0354     HEAD DEP BOUNDS =       668.2915
           TOTAL IN =     77953.2970            TOTAL IN =      2514.6225

         OUT:                               OUT:
         ----                               ----
            STORAGE =     35599.0869             STORAGE =      1148.3576
      CONSTANT HEAD =         0.0000       CONSTANT HEAD =         0.0000
    HEAD DEP BOUNDS =     42354.2138     HEAD DEP BOUNDS =      1366.2650

          TOTAL OUT =     77953.3007            TOTAL OUT =      2514.6226
```

**Figure 3.22.** Example General Head Boundary (GHB) package input without and with budget groups: *A*, Example GHB input without **BUDGET_GROUPS**; *B*, Example GHB input that defines two budget group names; *C*, Example LIST file Volumetric Budget output when not using budget groups; and *D*, Example LIST file Volumetric Budget output with two GHB budget groups. [Bold lettering is used to indicate important parts of the figure. Note that if GHB does not use budget groups, then the default MODFLOW Volumetric Budget name for GHB is "HEAD DEP BOUNDS"]

***D***

```
------------------------------------------------------------------------------
 VOLUMETRIC BUDGET FOR ENTIRE MODEL AT END OF TIME STEP  1 IN STRESS PERIOD  1
------------------------------------------------------------------------------

    CUMULATIVE VOLUMES      L**3       RATES FOR THIS TIME STEP      L**3/T
    ------------------                 ------------------------

          IN:                                IN:
          ---                                ---
            STORAGE =     57236.2616           STORAGE =      1846.3310
      CONSTANT HEAD =         0.0000     CONSTANT HEAD =         0.0000
          GHB_OCEAN =     20717.0354         GHB_OCEAN =       668.2915
        GHB_INLAND =          0.0000       GHB_INLAND =         0.0000

           TOTAL IN =     77953.2970          TOTAL IN =      2514.6225

          OUT:                               OUT:
          ----                               ----
            STORAGE =     35599.0869           STORAGE =      1148.3576
      CONSTANT HEAD =         0.0000     CONSTANT HEAD =         0.0000
          GHB_OCEAN =          0.0000         GHB_OCEAN =         0.0000
        GHB_INLAND =     42354.2138       GHB_INLAND =      1366.2650

           TOTAL OUT =     77953.3007          TOTAL OUT =      2514.6226
```

**Figure 3.22.** —Continued

## Two WEL Packages

The well package (WEL) source code was rewritten to mimic the input style of the GHB package with regard to TabFiles and includes the new LineFeed input format and additional options. The original WEL package is still retained in MF-OWHM2 to maintain backward compatibility with respect to the old TabFile inputs. The original WEL package is declared in the Name File by the package name "WEL1", and the new WEL package is declared by the package name "WEL". This allows for two sets of WEL packages to be defined; the only difference between the two is how **`TABFILE`** is specified, LineFeed as an input option, and the MF-OWHM2 **`OPTION`** block keywords.

A new option for the WEL package is provided by the keyword **`SMOOTHING`**, which activates MODFLOW-NWT (Niswonger and others, 2011) style smoothing of pumpage when the head approaches the bottom of a model cell for all flow packages (note that this smoothing works with all the solver packages). This helps alleviate the "Wet-Dry" problem where a high pumping rate in a well causes oscillations in the head calculation for a cell. The new input instructions are presented in figure 3.23.

***A***

**FOR EACH SIMULATION**

0. [#Text]
   Item 0 is optional—"#" must be in column 1. Item 0 can be repeated multiple times.

1a. [**PARAMETER** NPWEL MXL]
   This optional item must start with the keyword "PARAMETER".

2a. [**BEGIN BUDGET_GROUPS**] BLOCK STYLE INPUT*
   [BGROUP] REPEAT AS NEEDED, ONE PER LINE
   [**END BUDGET_GROUPS**]

2b. [**BEGIN LINEFEED** [FEEDOPT]]** BLOCK STYLE INPUT*
   [FEED_FILE] specified using *Generic_Input,*
   REPEAT AS NEEDED, ONE PER LINE
   [**END LINEFEED**]

2c. [**BEGIN OPTIONS**] BLOCK STYLE INPUT*
   [OPT] REPEAT AS NEEDED, ONE PER LINE
   [**END OPTIONS**]

   * Note that the order of the block style input does not matter.
   The order of items 2a, 2b, and 2c is recommended, but not required.
   In each block, comments are preceded with a "#" and blank lines are ignored.

   ** LINEFEED input structure is described in appendix 2.

3. [**TABFILE** NTAB FILEIO TIMEOPTION [**SPBASIS**] [**TABEQN**]]***
   [TABNAM TABLOCATION] READ NTAB TIMES IF NTAB>0,
   ONE TABNAM PER LINE

   *** Note that item 3 may also be specified in the OPTIONS block.
   If it is in the options block, then item 3 cannot be specified
   (double specifying TABFILE will raise an error).

4. MXACTW IWELCB

5. [PARNAM PARTYP Parval NLST [INSTANCES NUMINST]]

6a. [INSTNAM]

6b. [Layer Row Column Qfact [BGROUP] [xyz] [TABNAM TSFAC [TAB_EQN]]

***B***

**FOR EACH STRESS PERIOD**

7. ITMP NP

8. Layer Row Column Q [BGROUP] [xyz] [TABNAM TSFAC [TAB_EQN]
   –Repeat Item 8 ITMP times

9. [PNAME [INAME]]
   –Repeat Item 9 NP times

**Figure 3.23.** Structure of WEL package input format: *A*, for each simulation; *B*, for each stress period; *C*, explanation of input variables used by the keywords.

*C*

| | |
|---|---|
| Text | is an optional character variable (699 characters) that is written to LIST file.<br>Text may be specified on multiple lines, but a "#" must begin each line's Text. |
| NPWEL | is the number of WEL parameters defined in items 5 and 6.<br>Note, a WEL parameter must be defined in items 5 and 6 and then made active using items 7 and 8 to have an effect in the simulation.<br>If not specified, then NPWEL is set to 0. |
| MXL | is the maximum number of WEL cells to be defined using parameters. |
| MXACTW | is the maximum number of wells in use during any stress period.<br>MXACTW includes wells defined using parameters and defined without using parameters.<br>Wells defined in LINEFEED are added to MXACTW, so if only LINEFEED wells are used, then the initial value of MXACTW is set to 0.<br>If LINEFEED is in use and IWELCB is defined in the OPTIONS block, then MXACTW is optional. |
| IWELCB | is a flag and a file unit number. It may be specified in the OPTIONS block or in Data Item 4.<br>If IWELCB > 0, it is the unit number to which cell-by-cell flow terms will be written when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control.<br>If IWELCB = 0, cell-by-cell flow terms will not be written.<br>If IWELCB < 0, well recharge for each well will be written to the LIST file if "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control. |
| BGROUP | is the budget group name, up to 16 characters long.<br>If the BUDGET_GROUPS block is present and contains at least one BGROUP, then BGROUP is required in items 6b and 8 and is one of the names specified in the block. |
| FEED_FILE | is a separate file that represents the LineFeed alternate input (appendix 2).<br>If FEED_FILEs are specified, then this entire input structure is optional (except for the LINEFEED block), and the wells in use are determined by the FEED_FILEs. |

**Figure 3.23.** —Continued

***C** (continued)*

OPT is an optional keyword. Multiple options are allowed, but only one per line within the block. The following are the available options:

"**WEL_CBC** IWELCB" specifies the cell-by-cell unit number, IWELCB.
If this OPT is in the **OPTIONS** block, then IWELCB in Data Item 4 is not required.

"**NOPRINT**" specifies that WEL input lists are not to be written to the listing file.

"**SMOOTHING BY_FRACTION** PHIRAMP [Print_Smooth]" or
"**SMOOTHING BY_LENGTH** PHIRAMP [Print_Smooth]"

specifies that WEL pumping rate is smoothed towards zero when the head of the cell it pumps approaches the cell bottom. This option is available for all flow packages and applies to both confined and convertible layers.

PHIRAMP is the threshold fraction or length of the cell thickness that represents the saturated thickness when smoothing is activated. If the layer is declared as "confined", then the saturated thickness is the cell head minus the cell's bottom elevation.

**BY_FRACTION** indicates that PHIRAMP is a fraction between 0.00001 and 1.0, which is multiplied by the cell thickness to get the smoothing saturated thickness.

**BY_LENGTH** indicates that PHIRAMP is the smoothing saturated thickness.

Print_Smooth is an optional *Generic_Output* file that contains a transcript of the smoothing that is applied to wells.

"**AUXILIARY** abc" or "**AUX** abc" defines an auxiliary variable by the keyword abc.
For each auxiliary variable, abc, a corresponding integer, xyz, is read for each WEL cell in items 6b and 8. The order of input values for each auxiliary abc must be the same order in which values for the xyz are loaded.

"**TABPRINT**" specifies that WEL TabFiles write additional information to the LIST file.

**Figure 3.23.** —Continued

***C** (continued)*

TABFILE is a keyword that triggers reading subsequent TabFile information.

NTAB is the number of tabfiles that will be read.

FILEIO is a keyword that determines how TabFiles are handled in terms of memory usage and file input and output. The accepted values are

0 to indicate the entire TabFile is loaded into memory and processed.

1 to indicate only the part of the TabFile that pertains to the current time step is loaded into memory and processed (recommended option).

**TIMEOPTION** is a required keyword that must be specified by one of the two following keywords.

**SIMTIME** specifies that TabFile times use the model simulated time with time units specified by the DIS package input and a starting point of 0. This is the default operation if **TIMEOPTION** is not specified.

**REALTIME** specifies that TabFile times use decimal years that begin with the date specified in either the BAS package option **START_DATE** (recommended) or the DIS package keyword **STARTTIME** (legacy option).

**IGNORE_TIME** specifies that the time values specified be ignored when writing to the TabFile and instead an ordered one-to-one relationship between the time steps and TabFile entries (that is, a row in the TabFile) is assumed.

**SPBASIS** is an optional keyword; it indicates that TabFile times should be parsed based on the stress period rather than the time step. If the **IGNORE_TIME** option is used, then one row of the TabFile is loaded for each stress period.

**TABEQN** is an optional keyword; it indicates that TAB_EQN is an equation that is loaded and that the TabFile value is passed to it. Please see appendix 2 for more details.

TAB_EQN is an equation that is read when the **TABEQN** keyword is specified. This equation must be enclosed in single quotes and can perform any of the operations defined by the "ExpressionParser" (Hanson and others, 2014). The following are reserved names that are replaced with values:

**TAB** is replaced in TAB_EQN with the TabFile value for the current step. For example, '5*TAB + TAB^2' multiplies the TabFile value by 5 and then adds its square to that value.

**SIM** is replaced in TAB_EQN with the simulated time at the end of the time step.

**REL** is replaced in TAB_EQN with the decimal year at the end of the time step.

TABNAM is a unique name (maximum of 20 characters) that identifies the TabFile.

TSFAC is a scale factor multiplied with the final TabFile value. If TAB_EQN is specified, then it is done first and TSFAC is applied to the result.

TABLOCATION is either the location (relative or absolute) of the TabFile or the keyword **EXTERNAL** followed by a unit number that is defined in the Name file. The TabFile is self-counted, stored, and associated with the unique ID TABNAM.

**Figure 3.23.** —Continued

***C** (continued)*

| | |
|---|---|
| PARNAM | is the name of a parameter up to 10 characters long and is not case sensitive. |
| PARTYP | is the type of parameter. For the WEL Package, the only allowed parameter type is **Q**, which defines values of the volumetric recharge rate. |
| Parval | is the parameter value. Values defined in the Parameter Value File supercede Parval. |
| NLST | is the number of wells that are included in a non-time-varying parameter or in each instance of a time-varying parameter. |
| LAYER | is the layer number of the model cell that contains the well. |
| ROW | is the row number of the model cell that contains the well. |
| COLUMN | is the column number of the model cell that contains the well. |
| Qfact | is the factor used to calculate well recharge rate from the parameter value. The recharge rate is the product of Qfact and the parameter value. |
| xyz | represents any auxiliary variables for a well defined in the **OPTIONS** block. |
| ITMP | is a flag and counter. It is optional if LineFeed is used and NPWEL = 0<br>If ITMP < 0, then non-parameter well data is reused<br>If ITMP ≥ 0, then ITMP is the number of non-parameter well data to load in Item 8. |
| NP | is the number of parameters in use in the current stress period, where 0 ≤NP ≤ NPWEL. ITMP must be specified if NP is specified. Specifying NP is optional if NPWEL = 0. |
| Q | is the volumetric net-recharge rate.<br>A positive value indicates recharge, and a negative value indicates discharge (pumping). |
| PNAME | is the parameter name in use for the stress period. NP parameter names are read. |

**Figure 3.23.** —Continued

## General Head Boundary (GHB) Flow Package Linkage and Other Updates

### GHB Options in Block-Style Input

The GHB options originally were to the right of the first input line after the optional **PARAMETER** keyword. This structure caused problems if there were a large number of options or the user made a mistake with one of the options, which resulted in the rest being ignored. To overcome this limitation, the option part of the input was moved to a *Block-Style* input just after the **PARAMETER** keyword (see item 2b of the "GHB Input Structure" section for more details). If it is desired, the **PARAMETER** keyword may be placed in the **OPTIONS** block (either way it is parsed correctly).

The *Block-Style* input starts with the words "**BEGIN OPTIONS**", has one option per line, and is terminated with the words "**END OPTIONS**". In the block, any blank lines are automatically skipped, and all comments must be preceded by a "#" symbol. Figure 3.24 shows an example set of options.

```
BEGIN OPTIONS
   # COMMENT
   NOPRINT                          # SUPRESS GHB LIST WRITING
                                    # SKIPPED LINE
   AUX FLOW_PACK_COND               # AUXILIARY OPTION - Flow Package Linkage
   AUX VARIABLE_CONDUCTANCE         # AUXILIARY OPTION - Variable Conductance
END OPTIONS
```

**Figure 3.24.** Structure of the GHB package **OPTIONS** block with example options.

## GHB Flow Package Linkage

The MODFLOW 2005 General Head Boundary Package (GHB) input required the user to specify a "conductance" that represented the porous medium that the boundary condition flow passed through to the aquifer. To provide a direct linkage to the MF-OWHM2 Flow package, the GHB package is now configured to optionally construct this conductance from the flow package horizontal hydraulic conductivity. The GHB can obtain its conductance from the flow package only when using the Layer Property Flow (LPF) and Upstream Weighting Flow (UPW) packages. To construct the conductance, the boundary condition (the $GHB_{i,j,k}$ cell, fig. 3.25) is assumed to be a model cell adjacent to the model cell it is associated with and has identical aquifer properties. The identical properties are hydraulic conductivity, row width ($\Delta r$), column length ($\Delta c$), and vertical height ($\Delta v$). If the model cell is convertible, then the GHB's vertical height is set equal to the cell's saturated thickness at the start of the time step.

The advantages of this linkage are that the boundary cell mimics the aquifer properties, making the input easier for the user, and that calibration is more constrained because the GHB conductance varies with the model cell conductance. Further, if the GHB cell is in a model layer defined as "convertible," where the conductance varies with the saturated thickness, then the cell's GHB calculated conductance also varies with saturated thickness. When this occurs, the saturated thickness is calculated using both the GHB boundary head (`BHead`) and the model cell's head (`HNEW`). For GHB cells using the vertical hydraulic conductivity, only use the saturated thickness and do not apply any correction factors for water-table conditions.

When the GHB package is linked to the flow package, the GHB input conductance (`Cond`) and the **`PARAMETER`** conductance factor (`CondFact`) are still part of the package input. Instead of being applied directly as the conductance, however, they act as a multiplier that is applied to the calculated conductance. This allows the user to scale the calculated GHB conductance to a larger or smaller value, if required for calibration or to improve the boundary flows.

To link the GHB package to the flow package, an auxiliary keyword called **`FLOW_PACK_COND`** must be specified in the GHB **`OPTIONS`** block. Note that MODFLOW-2005 auxiliary keywords are preceded by the word aux, so "**`AUX FLOW_PACK_COND`**" is the proper input. If the keyword is present, then each GHB cell's auxiliary value indicates which hydraulic conductivity direction to use to build the conductance (row, column, or vertical). Figure 3.26 lists the accepted auxiliary values, which indicate the GHB flow direction and if it varies with saturated thickness. The negative options (`-2` and `-1`, fig. 3.26) indicate that the GHB conductance does not vary with saturated thickness. The positive options `2` and `1` (fig. 3.26) indicate that if the model layer is declared as convertible, then the GHB conductance varies with saturated thickness.

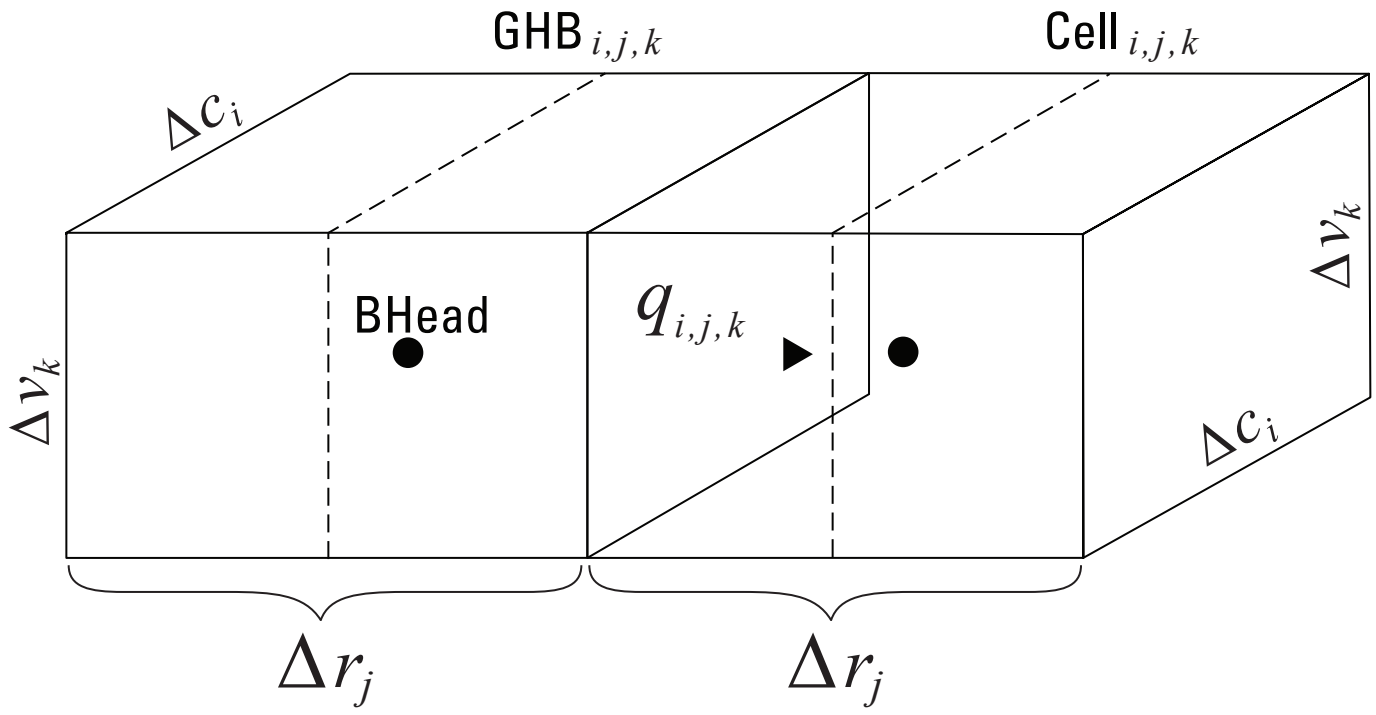


**Figure 3.25.** Example GHB (general head boundary) cell that flows in the row direction into model cell *i, j, k*. The BHead is the user-specified GHB boundary head value, $q_{i,j,k}$ is the flow rate, and $\text{Cell}_{i,j,k}$ is the model cell connected to the boundary condition.

```
-2 := GHB Flow in Column Direction,  conductance never varies with saturated thickness.
-1 := GHB Flow in Row Direction,     conductance never varies with saturated thickness.
 0 := GHB uses conductance specified in input, no flow package linkage.
 1 := GHB Flow in Row Direction,     conductance varies with saturated thickness, if convertible.
 2 := GHB Flow in Column Direction,  conductance varies with saturated thickness, if convertible.
 3 := GHB Flow in Vertical Direction, conductance never varies with saturated thickness.
```

**Figure 3.26.** Auxiliary values used to indicate which hydraulic conductivity direction to use to build the conductance.

## GHB Variable Conductance

One of the limitations of the GHB package had been that the conductance does not vary with saturated thickness. The GHB package is now configured to vary the user-specified conductance linearly with saturated thickness regardless of the layer type definition, convertible or confined (**VARIABLE_CONDUCTANCE**). To invoke this feature, an auxiliary variable is required and declared as "**AUX VARIABLE_CONDUCTANCE**" in the Options section of the input structure. If the keyword is present, then each GHB cell's auxiliary value is set to one to indicate the conductance varies with saturated thickness or to zero to not vary. This auxiliary variable is different from "**AUX FLOW_PACK_COND**" because it will vary the conductance for both confined and convertible model cells.

The saturated thickness is calculated as a fraction (SAT_FRAC) of cell saturation either using the boundary head or the model cell head, whichever is larger. The saturated fraction is then multiplied by the user-specified conductance for calculating the boundary flows. This is similar to how UPW handles saturated thickness between two model cells. Figure 3.27 presents the decision tree for determining SAT_FRAC.

A GHB cell that has enabled **VARIABLE_CONDUCTANCE** always varies the conductance (fig. 3.27). Specifically, if the same GHB cell has the flow package linkage is in use (**FLOW_PACK_COND**), then the conductance always varies with saturated thickness for the auxiliary options -2, -1, 0, 1, and 2 (fig. 3.26). This is different compared to only the **FLOW_PACK_COND** option, which does not vary conductance for -2 and -1 and only varies for convertible layers for 1 and 2.

```
SAT_HEAD = MAX( BHead, HEAD )

IF     ( SAT_HEAD > TOP ) THEN
                        SAT_FRAC = 1
ELSEIF ( SAT_HEAD < BOT ) THEN
                        SAT_FRAC = 0
ELSE
                        SAT_FRAC = (SAT_HEAD – BOT) / (TOP – BOT)
```

where

| | |
|---|---|
| BHead | is user specified boundary head |
| HEAD | is the model cell's head |
| TOP | is the model cell's top elevation |
| BOT | is the model cell's bottom elevation |
| SAT_FRAC | is the resulting saturated fraction |

**Figure 3.27.** General Head Boundary (GHB) package decision algorithm used to calculate a model cell's saturated fraction (SAT_FRAC), which is multiplied with the GHB conductance (Cond) to determine boundary flows.

## GHB Database Friendly Output

An optional database friendly output file is now available for the GHB package. This provides the user with the ability to check each time step's GHB input and easily load the results into data-column based software. This is important with the new "**AUX FLOW_PACK_COND**" and "**AUX VARIABLE_CONDUCTANCE**" because they calculate the conductance on the basis of aquifer properties and these can vary by time step. It also provides direct access to the resulting boundary flows for post-analysis. To activate the database output option, the keyword **DBFILE** followed by a *Generic_Output* file (appendix 1) name must be added to the **OPTIONS** Block (fig. 3.28). The *Generic_Output* file may be either a text file or binary. Figure 3.29 explains the meaning of each header.

If the GHB uses **BUDGET_GROUPS** (section "Budget_Groups—Splitting a Package Budget Information into Subgroups"), then it can print a database-friendly output file that only contains the model cells for a specific budget group. This functions identically to DBFILE option, but only writes a subset of the general head boundary cells. To activate the database output option, the keyword **BUDGET_GROUP_OUTPUT** is added to the **OPTIONS** block and is followed by the a BGROUP defined in the **BUDGET_GROUPS** block, then followed by a *Generic_Output* file name. Figure 3.30 is an example **OPTIONS** block that makes the database file "`./output/GHB_Inland_Flow.txt`", which only contains the GHB cells that are part of the GHB_LAND group.

```
BEGIN  OPTIONS
               DBFILE   Generic_Output
END
```

**Figure 3.28.** Structure of the General Head Boundary (GHB) package **OPTIONS** block input with the **DBFILE** keyword. This option writes GHB flows to a format to print the time information to an external file. [Generic_Output is the location to write the output file to.]

***A***

| | | |
|---|---|---|
| DATE_START | is the calendar date at the start of the time step (yyyy-mm-ddThh:mm:ss) | |
| PER | is the stress period number | |
| STP | is the time step number | |
| DELT | is the time step length | [T] |
| SIMTIME | is the total simulated time at the end of the time step in model units | [T] |
| LAY | is layer of the GHB cell | |
| ROW | is row of the GHB cell | |
| COL | is column of the GHB cell | |
| GHB_CONDUCTANCE | is the conductance of the GHB cell | [$L^2$/T] |
| GHB_HEAD | is the boundary head value of the GHB cell | [L] |
| HEAD | is the groundwater head for the model cell at LAY, ROW, COL | [L] |
| GHB_FLOW_RATE | is the flow rate from the boundary condition. Positive is inflow to groundwater | [$L^3$/T] |

***B***

| | |
|---|---|
| DATE_START | CHARACTER(19), starting date formatted as 'yyyy-mm-ddThh:mm:ss' |
| PER | INTEGER |
| STP | INTEGER |
| DELT | DOUBLE |
| SIMTIME | DOUBLE |
| LAY | INTEGER |
| ROW | INTEGER |
| COL | INTEGER |
| GHB_CONDUCTANCE | DOUBLE |
| GHB_HEAD | DOUBLE |
| HEAD | DOUBLE |
| GHB_FLOW_RATE | DOUBLE |

**Figure 3.29.** General Head Boundary (GHB) package **OPTIONS** block keyword **DBFILE** text file header explanation and binary record structure: *A*, text-header explanation; and *B*, binary-record structure. [[T], unit of time in model units; [L], length in model units; [$L^2$/T], area per time in model units; [$L^3$/T], volumetric rate in model units; CHARACTER(19) indicates a record is 19 characters long (19 bytes); INTEGER is a 4-byte integer record; DOUBLE is a 8-byte floating-point number record.]

```
BEGIN  BUDGET_GROUPS

     # Assumes that GHB input is appropriately tagged with each BGROUP
     GHB_SEA      # BGROUP 1
     GHB_LAND     # BGROUP 2
END BLOCK

BEGIN OPTIONS
     # Print Database file of GHB cells associated with GHB_LAND budget group
     #                          BGROUP     Generic_Output
     BUDGET_GROUP_OUTPUT  GHB_LAND  ./output/GHB_Inland_Flow.txt
END OPTIONS
```

**Figure 3.30.** Example of General Head Boundary (GHB) package **BUDGET_GROUPS** and **OPTIONS** block input to illustrate the use of the **BUDGET_GROUP_OUTPUT** option.

## GHB Input Structure

Figure 3.31 is the input directions for the GHB package. The **BUDGET_GROUPS** block is defined in "Budget_Groups—Splitting a Package Budget Information into Subgroups" section and the **LINEFEED** block is defined in appendix 2. If a **LINEFEED** block is used, then the entire input structure beyond that block is optional—except for the **LINEFEED** block—and the FeedFile's input is applied for each stress period.

*A*

**FOR EACH SIMULATION**

0. [#Text]
   Item 0 is optional—"#" must be in column 1. Item 0 can be repeated multiple times.

1a. [**PARAMETER** NPGHB MXL]
   This optional item must start with the keyword "PARAMETER".

2a. [**BEGIN BUDGET_GROUPS**] BLOCK STYLE INPUT*
   [BGROUP] REPEAT AS NEEDED, ONE PER LINE
   [**END BUDGET_GROUPS**]

2b. [**BEGIN LINEFEED** [FEEDOPT]]** BLOCK STYLE INPUT*
   [FEED_FILE] specified using *Generic_Input,*
   REPEAT AS NEEDED, ONE PER LINE
   [**END LINEFEED**]

2c. [**BEGIN OPTIONS**] BLOCK STYLE INPUT*
   [OPT] REPEAT AS NEEDED, ONE PER LINE
   [**END OPTIONS**]

   * Note that the order of the block style input does not matter.
   The order of items 2a, 2b, and 2c is recommended, but not required.
   In each block, comments are preceded with a "#" and blank lines are ignored.

   ** LINEFEED input structure is described in appendix 2.

3. [**TABFILE** NTAB FILEIO TIMEOPTION [**SPBASIS**] [**TABEQN**]]***
   [TABNAM TABLOCATION] READ NTAB TIMES IF NTAB>0,
   ONE TABNAM PER LINE

   *** Note that item 3 may also be specified in the OPTIONS block.
   If it is in the options block, then item 3 cannot be specified
   (double specifying TABFILE will raise an error).

4. MXACTB IGHBCB

5. [PARNAM PARTYP Parval NLST [INSTANCES NUMINST]]

6a. [INSTNAM]

6b. [Layer Row Column Bhead CondFact [BGROUP] [xyz] [TABNAM TSFAC [TAB_EQN]]

*B*

**FOR EACH STRESS PERIOD**

7. ITMP NP

8. Layer Row Column Bhead Cond [BGROUP] [xyz] [TABNAM TSFAC [TAB_EQN]]
   –Repeat Item 8 ITMP times

9. [PNAME [INAME]]
   –Repeat Item 9 NP times

**Figure 3.31.** Structure of General Head Boundary (GHB) package input format: *A*, for each simulation; *B*, for each stress period; and *C*, explanation of input variables used by the keywords.

*C*

| | |
|---|---|
| Text | is an optional character variable (699 characters) that is written to LIST file.<br>Text may be specified on multiple lines, but each line's Text must begin with a "#". |
| NPGHB | is the number of GHB parameters defined in items 5 and 6.<br>Note, a GHB parameter must be defined in items 5 and 6 and then made active using items 7 and 8 to have an effect in the simulation.<br>If not specified, then NPGHB is set to 0. |
| MXL | is the maximum number of GHB cells to be defined using parameters. |
| MXACTB | is the maximum number of GHB cells in use during any stress period.<br>MXACTB includes GHB cells defined using parameters and defined without using parameters. GHB cells defined in LINEFEED are added to MXACTB, so if only LINEFEED GHB cells are used, then the initial value of MXACTB is set to 0.<br>If LINEFEED is in use and IGHBCB is defined in the OPTIONS block, then MXACTB is optional. |
| IGHBCB | is a flag and a file unit number. It may be specified in the OPTIONS block or in Data Item 4.<br>If IGHBCB > 0, it is the unit number to which cell-by-cell flow terms will be written when "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control.<br>If IGHBCB = 0, cell-by-cell flow terms will not be written.<br>If IGHBCB < 0, boundary flow for each GHB cell will be written to the LIST file if "SAVE BUDGET" or a non-zero value for ICBCFL is specified in Output Control. |
| BGROUP | is the budget group name, up to 16 characters long.<br>If the BUDGET_GROUPS block is present and contains at least one BGROUP, then BGROUP is required in items 6b and 8 and is one of the names specified in the block. |
| FEED_FILE | is a separate file that represents the LineFeed alternate input (appendix 2).<br>If FEED_FILEs are specified, then this entire input structure is optional (except for the LINEFEED block), and the GHB cells in use are set by the FEED_FILEs. |

**Figure 3.31.** —Continued

***C** (continued)*

OPT is an optional keyword. Multiple options are allowed, but only one per line within the block. The following are the available options:

"**GHB_CBC** IGHBCB" specifies the cell-by-cell unit number, IGHBCB.
If this OPT is in the **OPTIONS** block, then IGHBCB in Data Item 4 is not required.

"**DBFILE** *Generic_Output*" writes GHB boundary flow to an external file.

"**BUDGET_GROUP_OUTPUT** BGROUP *Generic_Output*" writes for the specified GHB BGROUP the boundary flow to an external file.

"**NOPRINT**" specifies that GHB input lists are not to be written to the listing file.

"**AUXILIARY** abc" or "**AUX** abc" defines an auxiliary variable by the keyword abc. For each auxiliary variable, abc, a corresponding integer, xyz, is read for each GHB cell in items 6b and 8. The order of input values for each auxiliary abc must be the same order in which values for the xyz are loaded. The following are some of the supported auxiliary options.

"**AUX FLOW_PACK_COND**" specifies that xyz is a set of integers (flag values) that indicates how the GHB conductance is assembled.

"**AUX VARIABLE_CONDUCTANCE**" specifies that xyz is a set of integers (flag values) that indicates if saturated thickness is used to scale the GHB conductance.

"**TABPRINT**" specifies that GHB TabFiles write additional information to the LIST file.

**Figure 3.31.** —Continued

***C** (continued)*

| | |
|---|---|
| TABFILE | is a keyword that triggers reading subsequent TabFile information. |
| NTAB | is the number of TabFiles that will be read. |
| FILEIO | is a keyword that determines how TabFiles are handled in terms of memory usage and file input and output. The accepted values are<br>0 to indicate the entire TabFile is loaded into memory and processed.<br>1 to indicate only the part of the TabFile that pertains to the current time step is loaded into memory and processed (recommended option). |
| **TIMEOPTION** | is a required keyword that must be specified by one of the two following keywords.<br>**SIMTIME** specifies that TabFile times use the model simulated time with time units specified by the DIS package input and a starting point of 0. This is the default operation if **TIMEOPTION** is not specified.<br>**REALTIME** specifies that TabFile times use decimal years that begin with the date specified in either the BAS package option **START_DATE** (recommended) or the DIS package keyword **STARTTIME** (legacy option).<br>**IGNORE_TIME** specifies that the time values specified be ignored when writing to the TabFile and instead an ordered one-to-one relationship between the time steps and TabFile entries (that is, a row in the TabFile) is assumed. |
| **SPBASIS** | is an optional keyword; it indicates that TabFile times should be parsed based on the stress period rather than the time step. If the **IGNORE_TIME** option is used, then one row of the TabFile is loaded for each stress period. |
| **TABEQN** | is an optional keyword; it indicates that TAB_EQN is an equation that is loaded and that the TabFile value is passed to it. Please see appendix 2 for more details. |
| TAB_EQN | is an equation that is read when the **TABEQN** keyword is specified. This equation must be enclosed in single quotes and can perform any of the operations defined by the "ExpressionParser" (Hanson and others, 2014). The following are reserved names that are replaced with values:<br>**TAB** is replaced in TAB_EQN with the TabFile value for the current step. For example, 'TAB/3 + TAB^2' divides the TabFile value by 3 and then adds its square to that value.<br>**SIM** is replaced in TAB_EQN with the simulated time at the end of the time step.<br>**REL** is replaced in TAB_EQN with the decimal year at the end of the time step. |
| TABNAM | is a unique name (maximum of 20 characters) that identifies the TabFile. |
| TSFAC | is a scale factor multiplied with the final TabFile value. If TAB_EQN is specified, then it is done first and TSFAC is applied to the result. |
| TABLOCATION | is either the location (relative or absolute) of the TabFile or the keyword **EXTERNAL** followed by a unit number that is defined in the Name file. The TabFile is self-counted, stored, and associated with the unique ID TABNAM. |

**Figure 3.31.** —Continued

***C** (continued)*

| | |
|---|---|
| PARNAM | is the name of a parameter; up to 10 characters long and is not case sensitive. |
| PARTYP | is the type of parameter. For the GHB Package, the only allowed parameter type is **GHB**, which defines values of the general-head boundary hydraulic conductance. |
| Parval | is the parameter value. Values defined in the Parameter Value File supersede Parval. |
| NLST | is the number of GHB cells that are included in a non-time-varying parameter or in each instance of a time-varying parameter. |
| LAYER | is the model layer of the cell affected by the head-dependent boundary. |
| ROW | is the model row of the cell affected by the head-dependent boundary. |
| COLUMN | is the model column of the cell affected by the head-dependent boundary. |
| Bhead | is the boundary head. This value can be over written by a TabFile. |
| CondFact | is the factor used to calculate hydraulic conductance from the parameter value. The calculated conductance is the product of CondFact and the parameter value. If the "**AUX FLOW_PACK_COND**" option is in use and the corresponding xyz is nonzero, then the conductance is calculated by the flow package and then multiplied by the CondFact and the parameter value. |
| xyz | represents any GHB auxiliary variables defined in the **OPTIONS** block. The values of auxiliary variables must be present in each repetition of items 6b and 8 if they are defined in item 2b. The values must be specified in the order used to define the auxiliary variables in item 2b. |
| ITMP | is a flag and counter. It is optional if LineFeed is used and NPGHB = 0 If ITMP < 0, then non-parameter GHB data is reused If ITMP ≥0, then ITMP is the number of non-parameter GHB cell to load in Item 8. |
| NP | is the number of parameters in use in the current stress period, where 0 ≤ NP ≤ NPGHB. ITMP must be specified if NP is specified. Specifying NP is optional if NPGHB = 0. |
| Cond | is the hydraulic conductance of the interface between the aquifer cell and the boundary. If the "**AUX FLOW_PACK_COND**" option is in use and the corresponding xyz is nonzero, then the conductance is calculated by the flow package and then multiplied by the Cond. |
| PNAME | is the parameter name in use for the stress period. NP parameter names are read. |

**Figure 3.31.** —Continued

Figure 3.32 is an example GHB input that makes use of **BUDGET_GROUPS**, **TABFILES**, and includes the **FLOW_PACK_COND** auxiliary flag. The package input is split into two budget groups, GHB_OCEAN and GHB_LAND. Each group is associated with two GHB model cells to separate out an ocean boundary condition for an inland one in the volumetric budget. BHead for the GHB cells associated with an ocean boundary condition uses the **TABEQN** to translate the sea-level gauge values (−5.0) to a freshwater equivalent (−3.875). This takes the values in the TabFile ("File: GHB_Ocean_Level.txt", fig. 3.32*B*), then passes it through the specified function (for example, **'1.025*TAB + 1.25'**). Conversely, the inland boundary condition does not specify an equation, so it sets BHead to the TabFile value of −5.0 ("File: GHB_Inlnd_Level.txt", fig. 3.32*B*). All the GHB cells have set the xyz auxiliary flag to 2, which indicates that the conductance is calculated using the flow package's hydraulic conductivity in the column direction (fig. 3.26).

*A*

```
# General-Head Boundary package (GHB) Main Input
#        Model has NLAY, NROW, NCOL = 1, 5, 2, respectively. BAS Package START_DATE = 1/1/2000
#
# TabEqn to convert seawater water head (h)to freshwater equivalent head (hf) is
#   hf = 1.025*h - 0.025*Zp       where Zp is the cell center elevation
#                                 For this example, Zp = –50, so "–0.025*Zp" = + 1.25, which makes
#   hf = 1.025*h + 1.25
#
BEGIN BUDGET_GROUPS
                    GHB_OCEAN     # BGROUP_1
                    GHB_INLAND    # BGROUP_2
END
#
BEGIN OPTIONS
                    NOPRINT
                    #         Generic_Output
                    DBFILE  ./output/GHB_Flow.txt
                    #                      BGROUP       Generic_Output
                    BUDGET_GROUP_OUTPUT GHB_INLAND ./output/GHB_Inland_Flow.txt
                    #
                    AUX FLOW_PACK_COND   # indicates auxiliary flag “xyz” is specified
END
# if TABNAM is specified, then Bhead replaced by Tabfile result
#          NTAB  FILEIO  TIMEOPTION   [SPBASIS]  [TABEQN]
TABFILE   2     1      REALTIME             TABEQN
  #
  OCEAN_TAB  ./Tabfiles/GHB_Ocean_Level.txt   # TABNAM   TABLOCATION
  INLND_TAB  ./Tabfiles/GHB_Inlnd_Level.txt
  #
4     0      # MXACTC  IGHBCB
4     0      # ITMP  NP                          -> Stress period 1
# L  R  C  BHead  Cond  BGROUP       xyz  TABNAM      TSFAC   TAB_EQN
  1  1  1  NaN    1.0   GHB_OCEAN     2   OCEAN_TAB   1.0     ‘1.025*TAB + 1.25’
  1  1  2  NaN    1.0   GHB_OCEAN     2   OCEAN_TAB   1.0     ‘1.025*TAB + 1.25’
  1  5  1  NaN    1.0   GHB_INLAND    2   INLND_TAB   1.0
  1  5  2  NaN    1.0   GHB_INLAND    2   INLND_TAB   1.0
```

**Figure 3.32.** Example General Head Boundary (GHB) package input: *A*, GHB package main input file; *B*, TabFiles that are read by the GHB input; and *C*, example output that is produced from the GHB input.

***B***

File: GHB_Ocean_Level.txt

```
# Ocean sea level gauge
# Note the Tabfile has only two entries with the same value, so BHead is always set to -5
#
1/1/2000  -5.0
2/1/2000  -5.0
```

File: GHB_Inlnd_Level.txt

```
# Inland boundary condition level
# Note the Tabfile has only two entries with the same value, so BHead is always set to -10
#
1/1/2000  -10.0
2/1/2000  -10.0
```

***C***

File: GHB_Flow.txt

| DATE_START | PER | STP | DELT | SIMTIME | LAY | ROW | COL | GHB_CONDUCTANCE | GHB_HEAD | HEAD | GHB_FLOW_RATE | GHB_BUD_GROUP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2000-01-01T00:00:00 | 1 | 1 | 31.0 | 31.0 | 1 | 1 | 1 | 959.72155 | -3.875 | -4.180 | 292.49909 | GHB_OCEAN |
| 2000-01-01T00:00:00 | 1 | 1 | 31.0 | 31.0 | 1 | 1 | 2 | 960.35909 | -3.875 | -4.053 | 166.99563 | GHB_OCEAN |
| 2000-01-01T00:00:00 | 1 | 1 | 31.0 | 31.0 | 1 | 5 | 1 | 920.90466 | -10.0 | -5.719 | -3937.45729 | GHB_INLAND |
| 2000-01-01T00:00:00 | 1 | 1 | 31.0 | 31.0 | 1 | 5 | 2 | 921.38166 | -10.0 | -5.619 | -3994.56188 | GHB_INLAND |

File: GHB_Inland_Flow.txt

| DATE_START | PER | STP | DELT | SIMTIME | LAY | ROW | COL | GHB_CONDUCTANCE | GHB_HEAD | HEAD | GHB_FLOW_RATE |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2000-01-01T00:00:00 | 1 | 1 | 31.0 | 31.0 | 1 | 5 | 1 | 920.90466 | -10.0 | -5.719 | -3937.45729 |
| 2000-01-01T00:00:00 | 1 | 1 | 31.0 | 31.0 | 1 | 5 | 2 | 921.38166 | -10.0 | -5.619 | -3994.56188 |

**Figure 3.32.** —Continued

## Streamflow Routing (SFR) Upgrades

### Time-Series Files and Line Feed

The Stream Flow Routing (SFR) Package is linked both to **LINEFEED** and Time Series Files to define the inflow at specified SFR segments. The **LINEFEED** and Time Series Files are described in detail in appendix 2. The SFR **LINEFEED**, described in appendix 2, is mentioned here to emphasize that it is a new feature of MF-OWHM2 that adds flexibility to the SFR input. To use **LINEFEED** with SFR, the **BEGIN LINEFEED** block must be specified at the start of the input file and closed with the keyword **END**. Time Series Files have a similar input structure as **LINEFEED** files because they are loaded as a Time Series File Group (appendix 2) in the block **BEGIN TIME_SERIES_INPUT**, which contains the Time Series File Group that loads the integer ID, which represents the SFR segment to which the Time Series File applies. Figure 3.33 shows the general input structure for loading time-series stream-inflow data from files using the **TIME_SERIES_INPUT** block within the SFR package.

***A***

```
BEGIN TIME_SERIES_INPUT
      #
      ISEG  Option  Generic_Input  # Repeat as needed
      #
END
```

***B***

| | |
|---|---|
| ISEG | is the SFR segment identifier that has its inflow set by the Time Series File (TSF). |
| **OPTION** | is the Time Series File interpretation option. For a complete description, please see Appendix 2. The following are select options presented in order of recommended use: |
| | **TIME_MEAN** uses an elapsed time weighted average of all data points that lie within the time step start and ending dates. |
| | **STEP_FUNCTION** uses the closest data point located before the end of the time step. |
| | **NEXT_VALUE** uses the closest data point located after the end of the time step. |
| | **NEAREST** uses the closest data point located near the end of the time step. |
| | **CONSTANT** VALUE disables the TSF and only uses VALUE for all time. |
| | **INTERPOLATE** using the two data points that are closest to the end of the time step. |
| *Generic_Input* | is location of the time series file. |

**Figure 3.33.** Inputting the time-series stream-inflow data using the **TIME_SERIES_INPUT** block within the SFR package: *A*, SFR TSF input structure; and *B*, explanation.

## Separate Flow Output File

The Stream Flow Routing (SFR) Package now supports an optional database friendly output. This provides the user with the ability to check each time step's SFR output in a data-column based software for post-analysis.

To activate the database output file, the user must add to the SFR input's **OPTIONS** block the keyword **DBFILE** followed by a *Generic_Output* file name (fig. 3.34*A*). Please see the section on *Generic_Output* (appendix 1) for further details and options for *Generic_Output* files. Figure 3.34*B* is an example **OPTIONS** block that writes the database file to "./output/SFR_DB.txt". If a calendar date is provided (BAS package option **START_DATE**), then the date is included in the output; otherwise, "NaN" is placed in the DATE_START column. The output header is a listing of column names corresponding to contents (fig. 3.35).

*A*

```
BEGIN  OPTIONS
               DBFILE   Generic_Output
END
```

*B*

```
BEGIN  OPTIONS
               DBFILE   ./output/SFR_DB.txt
END
```

**Figure 3.34.** Structure of the Streamflow Routing (SFR) package **OPTIONS** block input with the **DBFILE** keyword. This option writes SFR flows to a format to print the time information to an external file: *A*, general input structure; and *B*, example that specifies a file location. [Generic_Output is the location to write the output file to.]

*A*

| | | |
|---|---|---|
| DATE_START | is the calendar date at the start of the time step (yyyy-mm-ddThh:mm:ss). | |
| PER | is the stress period number. | |
| STP | is the time step number. | |
| DELT | is the time step length. | [T] |
| SIMTIME | is the total simulated time at the end of the time step in model units. | [T] |
| SEG | is the stream segment identifier. | |
| RCH | is the stream reach identifier. | |
| FLOW_IN | is the streamflow in to the reach. | [$L^3$/T] |
| FLOW_SEEPAGE | is the seepage across the streambed. A positive value is flow out of stream reach to groundwater. | [$L^3$/T] |
| FLOW_OUT | is the streamflow out of the reach. | [$L^3$/T] |
| RUNOFF | is the overland runoff into stream reach. | [$L^3$/T] |
| PRECIP | is the precipitation that falls on the stream reach. | [$L^3$/T] |
| STREAM_ET | is the evapotranspiration rate out of the stream reach. | [$L^3$/T] |
| HEAD_STREAM | is the stream stage expressed as hydraulic head | [L] |
| HEAD_AQUIFER | is the head of the uppermost active model cell that the stream reach is superimposed on. That is, the first active model cell beneath it. | [L] |
| DEPTH_STREAM | is the stream reach's maximum streambed water depth above channel bed. | [L] |
| LENGTH_STREAM | is stream reach's total length. | [L] |
| HEAD_GRADIENT | is the head gradient from the streambed to the aquifer. | [L] |
| COND_STREAM | is the vertical hydraulic streambed conductance. | [$L^2$/T] |
| ELEV_UP_STREAM | is the streambed elevation at the upper part of the reach. | [L] |
| FLOW_WT | is the recharge from the unsaturated zone beneath stream reach. This is only included when the UZF package is specified in the Name file. | [$L^3$/T] |
| CHNG_UNSAT_STOR | is the change in unsaturated-zone storage beneath stream reach. This is only included when the UZF package is specified in the Name file. | [$L^3$/T] |

**Figure 3.35.** Streamflow routing (SFR) package **OPTIONS** block keyword **DBFILE** text file header explanation and binary record structure: *A*, text-header explanation; and *B*, binary-record structure. [[T], unit of time in model units; [L], length in model units; [$L^2$/T], area per time in model units; [$L^3$/T], volumetric rate in model units; CHARACTER(19), indicates a record is 19 characters long (19 bytes); INTEGER, is a 4-byte integer record; DOUBLE, is a 8-byte floating-point number record.]

***B***

| | |
|---|---|
| DATE_START | CHARACTER(19), starting date formatted as 'yyyy-mm-ddThh:mm:ss' |
| PER | INTEGER |
| STP | INTEGER |
| DELT | DOUBLE |
| SIMTIME | DOUBLE |
| SEG | INTEGER |
| RCH | INTEGER |
| FLOW_IN | DOUBLE |
| FLOW_SEEPAGE | DOUBLE |
| FLOW_OUT | DOUBLE |
| RUNOFF | DOUBLE |
| PRECIP | DOUBLE |
| STREAM_ET | DOUBLE |
| HEAD_STREAM | DOUBLE |
| HEAD_AQUIFER | DOUBLE |
| DEPTH_STREAM | DOUBLE |
| LENGTH_STREAM | DOUBLE |
| HEAD_GRADIENT | DOUBLE |
| COND_STREAM | DOUBLE |
| ELEV_UP_STREAM | DOUBLE |
| FLOW_WT | DOUBLE; always included; set to is 0.0 if UZF package is not specified in the Name file. |
| CHNG_UNSAT_STOR | DOUBLE; always included; set to is 0.0 if UZF package is not specified in the Name file. |

**Figure 3.35.** —Continued

## New SFR Options

The SFR package was modified to include a set of new options available in the **OPTIONS** block. The following subsections introduce briefly each of those options. The new options are **DBFILE**, **PRINT_GW_FLOW_RESIDUAL**, **AUTOMATIC_NEGATIVE_ITMP**, and **NOPRINT**. The option **DBFILE** is discussed in the "Separate Flow Output File," so it is not discussed here.

### **PRINT_GW_FLOW_RESIDUAL**—Solver Information to External File

SFR option **PRINT_GW_FLOW_RESIDUAL** writes the convergence information for model cells that contain an SFR segment and reach to separate output files (fig. 3.36). This option is like the BAS package open **PRINT_FLOW_RESIDUAL** ("Obtaining Solver Information to External File" section), except that it only writes the solver flow residual error for model cells that contain an SFR reach.

***A***

```
BEGIN  OPTIONS
   #
   PRINT_GW_FLOW_RESIDUAL NTERM OUTER_START Generic_Output
   #
END
```

***B***

| | |
|---|---|
| NTERM | is the number of cells, that are underneath SFR reaches, to print for each outer iteration. Each outer iteration then prints NTERM model cells that had the largest change since the previous iteration. NTERM cannot exceed the total number of SFR reaches. |
| OUTER_START | is the solver iteration to begin printing the NTERM cells. If set to zero, then it only prints the final-converged or last-solver iteration. If set to a negative number, then it prints either final-converged iteration or iterations after the maximum solver iterations minus the OUTER_START plus one. |
| *Generic_Output* | is the file written to. |

***C***

| | |
|---|---|
| SP | is the stress-period number. |
| TS | is the time-step number. |
| ITER | is the outer iteration number |
| LAY | is the layer of the head value. |
| ROW | is the row of the head value. |
| COL | is the column of the head value. |
| SEG | is the stream segment identifier. |
| RCH | is the stream reach identifier. |
| HEAD | is the solver calculated head for the current solver iteration. |
| FLOW_RESIDUAL | is the flow residual error for the current solver iteration. |
| VOL_RESIDUAL | is the flow residual error times the time-step length to yield the volume residual for the current solver iteration. |
| CELL_VOLUME | is volume of the model cell at LAY, ROW, COL. |
| DATE | is the month and year of the time step. |
| CELL_ID | is the model cell's unique identifier (ID). The cell ID is useful for re-sorting the file to look at sequential changes in data for a specific model cell. The ID is determined by the following formula: $CELL\_ID = COL + NCOL * (ROW - 1) + NCOL * NROW * (LAY - 1)$ |

**Figure 3.36.** SFR package option **PRINT_GW_FLOW_RESIDUAL**. This option outputs to an external file the convergence criteria for model cells that contain an SFR reach: *A*, expected input in the SFR **OPTIONS** block; *B*, explanation of input variables used; and *C*, descriptions of the header items included in the external file.

### AUTOMATIC_NEGATIVE_ITMP—Only Define SFR Network Once

The standard SFR input requires that the entire stream network (segment and reaches) be specified for every stress period. This is necessary if the if there are changes in the network configuration, specified inflows, or specified diversions. If all the SFR stream inflow and diversions are defined by **LINEFEED**, **TABFILE**, or Time Series Files and the stream network does not change, then it is not necessary to define the network more than once. The SFR option **AUTOMATIC_NEGATIVE_ITMP** indicates that SFR should read the first stress period's input, then for each additional stress period reuse that input (which is equivalent to setting `ITMP = -1`). Specified stream inflow and diversions are then updated through the **LINEFEED**, **TABFILE**, or Time Series Files, but the network remains the same. This has the benefit of reducing the SFR input file size and improves the simulation execution speed.

### NOPRINT Option—Reduce LIST File Writing

The SFR package writes a large amount of information to the **LIST** file. Most of this information just repeats the stream-network input. This information is valuable when initially developing the stream network. Once the network has been established, the amount of information written to the **LIST** file can greatly increase the file size and result in longer simulation run time. A new SFR option, **NOPRINT**, that suppresses most of the SFR output to the **LIST** file is now part of MF-OWHM2. To use **NOPRINT**, it must be placed in the SFR **OPTIONS** block.

## PVAL, MULT, and ZONE Automatic Counting

PVAL, MULT, and ZONE packages may have comments, preceded by a "#" character, placed anywhere in their respective input structures. These packages offer automatic counting of the number of parameters, multiplier arrays, or zone arrays, respectively, by setting the count to –1. If the automatic counting is used, then any part of the input file that does not pertain to the actual input must be preceded by a "#" symbol (that is, commented). Otherwise there is a chance that the auto-count routine might include the comment in the count.

## WARN Package

Traditionally, MODFLOW wrote all errors and warnings to the Listing file. Because of the length of the Listing file, it was difficult to identify important warnings that various packages may have raised. The Warning Package (WARN) is an optional output package that contains a copy of all warnings in one location, regardless of the package of origin. The warnings are still written to the Listing file, but the WARN package provides a convenient common location for all warnings and errors. WARN is declared just like any other package with the package name, WARN, followed by the optional unit number, then by the file name, then the optional post-keywords. If using Local Grid Refinement (LGR), the WARN package may only be specified for the parent grid, and all warnings from the child grids are written to that file as well.

## LIST File Improvements

During a MODFLOW and MF-OWHM2 simulation run, a transcript of all operations is written to the Listing file (LIST). Previously, LIST always had to be enabled and declared at the start of the Name file. To remove this limitation, MF-OWHM2 does not require the LIST file for a simulation to continue and the LIST file may be specified anywhere in the Name file. The LIST file is now opened by *Generic_Output_OptKey* (appendix 2), which supports the post-keywords **SPLIT** and **BUFFER**. These two allow for the LIST file to be broken into multiple, smaller, files and to write the LIST to a buffered portion of RAM to improve speed.

### LIST File Is Optional

For large simulation models, the LIST file can become quite large. The large file size can affect hard-drive performance, slowing down the overall simulation run time. This is particularly important during calibration, when multiple copies of the Listing file can occupy a large amount of hard-drive space. If run time and hard-drive space is an issue, the LIST file is now optional for a MF-OWHM2 simulation. If it is not specified in the Name file, then it is not included in the simulation. LIST suppression was included in the MF-OWHM by using the `LSTLVL` feature, but this feature has been removed from MF-OWHM2 version 2 now that the Listing file is optional.

### Splitting the List File into Smaller Parts

Often the LIST file size can be problematic if it exceeds the allowable size to be opened in a text editor. That is, the file size exceeds the computers available RAM or is greater than 2 gigabytes for a 32-bit editor. To overcome this obstacle, the LIST file can now be split into multiple, smaller files. This allows for the LIST file to be created as a series of files that are approximately the same size (typically small enough to open in a text editor). The LIST file is split with the optional post-keyword **SPLIT** followed by the split size in megabytes (MB). After each time step, the size of the LIST file is checked; if it exceeds the split size, then that LIST file is closed, and output listing continues into a newly created "split file." When the LIST file is split, the newly created file has the same file name with a number appended to the end of it. The header in the original file is also included in each of the split files. Note that when the simulation is restarted, all split files are removed. For example, the following could be specified in the Name file: "LIST 55 ./List.txt **SPLIT** 900" to indicate that the LIST file is split when the file size exceeds 900 MB (note that 55 is the LIST file's unit number). The naming sequence used for each of the split LIST files is List.txt, then List01.txt, List02.txt, and so forth until the simulation ends.

### Buffering the List File

Another performance improvement to the LIST file is the ability to reserve a buffer of RAM where LIST output can be pre-written. RAM is a type of the computer system memory that can read and write data faster than the hard drive. By default, in MF-OWHM2, the LIST file reserves a 32-kilobyte (KB) buffer in RAM; once 32 KB of text has been written to the RAM, it is transferred to the hard drive—to the actual LIST file. Although this increases the speed of a simulation by minimizing the frequency of writing to the hard drive, it updates the LIST file in 32-KB chunks (that is, when new text appears in the file). Another limitation of the buffer appears if power is interrupted to the computer; anything stored in the 32-KB buffered RAM is lost before writing to the LIST file. To change the size of the RAM buffer, the keyword **BUFFER** followed by the buffer size in kilobytes, may be included after the LIST file name in the Name file. Specify "**BUFFER 0**" to disable buffering, which results in the immediate writing of the LIST output to the hard drive. From empirical tests of buffering performance, using the **BUFFER** option for the LIST file in MF-OWHM2 resulted in a 5-percent reduction in simulation runtime by changing the buffer size from 256 to 1024 KB. "LIST 55 ./List.txt **SPLIT** 900" is an example that buffers the LIST file in 1024 KB of RAM.

## Name File Updates

The Name file's read utilities were updated to allow unit numbers to be optional for packages and to support a set of new keywords that improve file operations. The sections that follow will provide details of each improved feature.

### New Keywords: Buffer, Read, Write

All Name file packages, DATA files, and DATA(BINARY) files support a new set of keywords that are specified to the right of the file name. Although they are optional, the effects of these keywords improve the input and output performance of MF-OWHM2, and they may be specified in any order. As with the use of LIST file buffering (see "Buffering the List File"), the keyword **BUFFER** followed by a buffer size in kilobytes is available to all packages, DATA files, and DATA(BINARY) files. By default, all Name file's files are buffered to 32 KB of RAM; to disable this, the keyword "**BUFFER 0**" must be added for each package in the Name file. The post-keyword **READ** is like the old keyword **OLD** and forces the file to be opened for 'read-only' access. This allows the same file to be accessed simultaneously by different programs or packages. For example, if two MF-OWHM2 simulations were running, they could access the same DATA file if it was opened with read only access. For **READ** files, the value of **BUFFER** should not exceed the size of the input file (rounded upward to a power of 2); otherwise, RAM is wasted because only as much as the entire file can be pre-loaded into the buffer. The opposite of the **READ** flag is the keyword **WRITE** (similar to the old keyword **REPLACE**), which indicates that the file is an output file that only allows writing access to it. For **WRITE** files, the buffer size should be set after considering both how often the user wants to see the output in the disk file during runtime and the amount of data that will be written to it. Typically, larger output files, such as the cell-by-cell type, should have a larger buffer than smaller ones, such as the HOB package's output. Empirical tests showed that buffer sizes greater than 1024 KB do not improve write performance. Note that Package files are input files, so only the keyword **READ** applies; if the keyword **WRITE** is used, it is ignored. Figure 3.37 indicates the keywords available to LIST, Packages, DATA, or DATA(BINARY) file-opening operations. Note that the keyword **SPLIT** is described in the "Splitting the List File into Smaller Parts" section. Figure 3.38 is an example Name file that makes use of the post-keywords.

| | Supported Post-Keywords | | | | | |
|---|---|---|---|---|---|---|
| Keyword | BUFFER | READ | WRITE | REPLACE | OLD | SPLIT |
| LIST | X | — | X | X | — | X |
| Packages | X | X | — | — | X | — |
| DATA | X | X | X | X | X | — |
| DATA(BINARY) | X | X | X | X | X | — |

**Figure 3.37.** Post-keywords available in the Name file that can be specified after the file name. [Supported post-keywords are marked with an X. Packages refer to all the MODFLOW and MF-OWHM2 packages. Abbreviation: —, not supported; X, is supported.]

```
LIST     50 ./LISTING_FILE.txt  SPLIT 500  BUFFER 1024
BAS6     51 ./BAS_FILE.txt                 BUFFER 64
DIS      52 ./DIS_FILE.txt      READ       BUFFER 64
LPF      53 ./LPF_FILE.txt      READ
DATA     54 ./CBC_FILE.txt      WRITE      BUFFER 512
DATA     55 ./INPUT_FILE.txt    READ       BUFFER 128
DATA   1979 ./SCOTT_FILE.txt    READ       BUFFER 16
DATA   1982 ./NO_BUFFER.txt     WRITE      BUFFER 0
```

**Figure 3.38.** Example Name file to illustrate the use of the post-keywords SPLIT, BUFFER, READ, and WRITE.

## Associate a Variable Name with Text

The Name file now supports assigning a variable name to set of text. The variable name may then be used as part of any file name in the Name file. A variable is declared by enclosing a word within percentage symbols (%) at the start of any line and is associated with the first space-delimited text to the right of it. If it is necessary to associate the variable with text that contains spaces, then it must be enclosed within matching single or double quotes. The variable name, enclosed in quotes, then may be used as any part of the file directory paths and is replaced by the text with which it is associated. This option allows for easily changing output file locations or pointing to different package groups to run different scenarios.

Figure 3.39 is an example Name file that assigns three variables. The first variable declares a common output location for all the data files, the second declares a common input location, and the third illustrates that an entire package path can be stored.

## Package Version Numbers Optional

In MODFLOW-2005, if a package had been updated to a new version, its name in the Name file included the version that was used. Because MODFLOW-2005 did not support, nor include, the previous version of the package, the version number became a potential source of input error. MF-OWHM2 loads both the package with the version number and without if there is only one unique version of the package, so this does not apply to the two WEL packages described in this appendix nor the MNW packages (MNW1 and MNW2). Figure 3.40 lists the packages that can have their names simplified.

```
#
# Variable name set via % % where package is declared.
# The declared variables are %OUTPUT%, %DATA_IN%, and %DIS_PACK%
#
# %VarName%  VarString
%OUTPUT%    ./Output/
#
%DATA_IN%   ./Input/Files/
#
# Quotes are required if the text assigned to the variable contains spaces
#
%DIS_PACK%   "./Scenario A/DIS.txt"
#
# Variable text is substituted where MF-OWHM2 finds %variable%
#
# MAIN PACKAGES ****************************
#
# Package          Unit    File Path
DIS                50    %DIS_PACK%
#
BAS                51    %DATA_IN%BAS.bas
LPF                52    %DATA_IN%LPF.lpf
GHB                53    %DATA_IN%GHB.chd
#
PCGN               54    ./SolverInput/PCGN.pcgn
#
# Output Files ***************************************************
# Package,         Unit,   File Path               Post-Keywords
LIST               60    %OUTPUT%LIST.lst  BUFFER 1024  SPLIT 900
WARN               61    %OUTPUT%Warn.txt  BUFFER 0
#
DATA(BINARY)   69    %OUTPUT%CBC.cbc       WRITE BUFFER 1024
DATA           96    %OUTPUT%HeadOut.fhd   WRITE
#
# Input Files – Necessary for EXTERNAL Unit references ********************
#
DATA           85    %DATA_IN%IBOUND.txt   READ BUFFER 64
```

**Figure 3.39.** Example Name file to illustrate how to associate a variable name with text. The variable name is initialized by surrounding a keyword with percent signs (%). [BUFFER X indicates the file should be first written to X kilobytes of RAM; SPLIT Y indicates that when the file size exceeds Y megabytes that it should start a new one; WRITE indicates the file is write only; READ indicates the file is read only.]

| Package Name | Alternative Name |
|---|---|
| BCF6 | BCF |
| LMT6 | LMT |
| HFB6 | HFB |
| HUF2 | HUF |
| BFH2 | BFH |
| SWI2 | SWI |

**Figure 3.40.** Packages that support using their name without the version number at the end.

## Name File Unit Numbers Are Optional for Packages

The normal set up for the Name file is to have the package name followed by an integer number that represents its Fortran file-unit number. This caused problems for users because each package had to have a unique unit number or there could be strange effects while running a simulation. The unit numbers are now optional, and when not specified, they are auto-assigned a value. Unit numbers are still required for `DATA` and `DATA(BINARY)` input types because they are referenced by unit number (`UNIT`) in other some package inputs (for example, **`EXTERNAL`** `UNIT`). Figure 3.41 is an example Name file that does not specify a unit number for each of the packages.

```
#
# Example illustrates that Package Unit numbers are optional.
#
# MAIN PACKAGES ****************************
#
# Package          [Unit]   File Path
DIS                       "./Scenario A/DIS.txt"
#
BAS                       ./Input/BAS.bas
LPF                       ./Input/LPF.lpf
GHB                       ./Input/GHB.chd
#
PCGN                      ./SolverInput/PCGN.pcgn
#
# Output Files ****************************************************
# Package,         [Unit],  File Path              Post-Keywords
LIST                      ./Output/LIST.lst BUFFER 1024  SPLIT 900
WARN                      ./Output/Warn.txt  BUFFER 0
#
# Unit numbers are still necessary for DATA and DATA(BINARY) files.
# Package,         Unit,    File Path              Post-Keywords

#
DATA(BINARY)   69   ./Output/CBC.cbc      WRITE BUFFER 1024
DATA           96   ./Output/HeadOut.fhd  WRITE
#
# Input Files – Necessary for EXTERNAL Unit references *********************
#
DATA           85   ./Input/IBOUND.txt  READ BUFFER 64
```

**Figure 3.41.** Example Name file to illustrate that specifying a unit number is optional for packages. [BUFFER X indicates the file should be first written to X kilobytes of RAM; SPLIT Y indicates that when the file size exceeds Y megabytes that it should start a new one; WRITE indicates the file is write only; READ indicates the file is read only.]

## Observation Process (HOB, DROB, GBOB, RVOB) New Features

### Write Observations at End of Each Time Step

The Observation Process packages—Head Observation (HOB), Drain Observation (DROB), General Head Observation (GBOB), and River Observation (RVOB)—were modified to allow printing to a separate file at the end of every time step. Previously, this was only available by setting values to the input variables `IUHOBSV`, `IUDROBSV`, `IUGBOBSV`, and `IURVOBSV`, such that observations were written to a separate file only if the simulation completed successfully. The problem with this is that during model calibration, a calibration software runs multiple simulation models with different parameter sets and if one of the evaluated parameter sets results in a simulation failure, then the calibration software stops because of a failure to load the Observation Process output files.

This issue was resolved by allowing the observation process packages to write to an external file at every time step. If the time step is before an observation's time, then the observation value written is either set to HDRY (HOB) or zero (DROB, GBOB, RVOB). This additional output file is initiated by specifying at the start of the input file—that is, after any comments and before specifying Data Set 1—the keyword **`TIME_STEP_PRINT_ALL`** followed by the location of where to write the observations (specified with *Generic_Output*). Formally this is defined as "**`TIME_STEP_PRINT_ALL`** *Generic_Output*", where *Generic_Output* (appendix 1) specifies the file to which the observation package writes its observations at every time step. If it is desired to use the same unit number as the one specified by `IUHOBSV`, `IUDROBSV`, `IUGBOBSV`, and `IURVOBSV`, then the keywords **`EXTERNAL`** or **`DATAUNIT`** should be used with the same unit number as that specified for the corresponding `IUxxOBSV` variable.

### Observations Include Calendar Dates Implicitly

The HOB was modified so that if the simulate specifies a starting date (Calendar Dates section), then it calculates the calendar date of each observation and prints it as part of the output. This is an automatic feature that is included whenever HOB is in use and BAS contains the keyword **`START_DATE`**. Figure 3.42 shows how the new HOB header when the simulation includes calendar dates.

Here, the two new header columns (fig. 3.42) are `DATE` and `DECIMAL_YEAR`. The DATE column contains the calculated date of the observation in the following format: yyyy-mm-dd, where mm is the month, dd is the day of the month, and yyyy is the Gregorian year (note the 24-hour clock time is not included). The `DECIMAL_YEAR` column contains the decimal year equivalent of the observation date and takes into account leap years (365- or 366-day years). Figure 3.43 is an example of the HOB output for one observation when the BAS contains the keyword **`START_DATE`**.

```
"SIMULATED EQUIVALENT" "OBSERVED VALUE" "OBSERVATION NAME” DATE DECIMAL_YEAR
```

**Figure 3.42.** Head Observation Process (HOB) output file header when the BAS package includes the option **`START_DATE`**. If the BAS option is not included, then DATE and DECIMAL_YEAR are not printed.

```
"SIMULATED EQUIVALENT" "OBSERVED VALUE" "OBSERVATION NAME” DATE      DECIMAL_YEAR
 7.96124649048E+01   4.03400001526E+01   SEB_0038_1        1979-4-23 1979.3082192
```

**Figure 3.43.** Head Observation Process (HOB) output file example when the BAS package includes the option **`START_DATE`**. If the BAS option is not included, then DATE and DECIMAL_YEAR are not printed.

## NWT Solver Upgrades

The Newton (NWT) solver previously only worked when using the Upstream Stream Weighting (UPW) package. The UPW package input is nearly identical to the Layer Property Flow (LPF) package, except for one integer flag, `IPHDRY`, that specifies if dry model cells should be set to the MODFLOW HDRY value or retain the head value returned from the solver. The BAS package was modified to allow the use of the NWT Solver with the LPF flow package and to allow for the UPW package to work with the other solvers—for example, PCG, PCGN, and GMG. When the NWT Solver and LPF are used together, dry cell values are retained (`IPHDRY=0`), upstream weighting is still used, and the following LPF options are not supported: **`STORAGECOEFFICIENT`**, **`CONSTANTCV`**, **`THICKSTRT`**, **`NOCVCORRECTION`**, **`NOVFC`**. In short, only the **`NOPARCHECK`** option is supported. If the other options are present, they are read, but not applied. Conversely, if the UPW package is not used with the NWT solver, it does not apply the upstream weighting, and it functions identically to LPF. This feature is provided for the convenience of not having to rebuild the LPF file to test a simulation using the NWT solver nor to rebuild the UPW when evaluating one of the other MODFLOW solvers.

The NWT solver performs a thin-cell check at the start of the simulation. This check automatically removes model cells—by setting their `IBOUND` value to zero—that have a vertical thickness considered too thin. The problem with this check is it can automatically remove a significant part of the model grid that is intended to be a part of the simulation. For example, a three-layer model that has a very "thin" second layer, representing an aquitard, could have most of the model cells removed from the simulation by changing their `IBOUND` values to zero. This would disable any vertical flow between the first and third layers, changing the aquitard to an aquifuge, without the user realizing it. To prevent this from happening NWT, by default, no longer performs the thin-cell check. If this feature is still desired, then the NWT input in the **`OPTIONS`** part of DATASET 1 now supports the keyword **`THIN_CELL_CHECK`**. The keyword **`THIN_CELL_CHECK`** may be placed either before or after the keyword **`CONTINUE`**, if it is present.

***A***

**FOR FIRST STRESS PERIOD OR ALL STRESS PERIODS**

0. [#Text]
   Item 0 is optional—"#" must be in column 1. Item 0 can be repeated multiple times.
1. MXITER ITER1 NPCOND IHCOFADD
2. HCLOSE RCLOSE RELAX NBPOL IPRPCG MUTPCG DAMPPCG [DAMPPCGT]

***B***

**OPTIONAL FOR EACH SUBSEQUENT STRESS PERIOD**

3. MXITER ITER1 NPCOND IHCOFADD
4. HCLOSE RCLOSE RELAX NBPOL IPRPCG MUTPCG DAMPPCG [DAMPPCGT]

**Figure 3.44.** Preconditioned Conjugate-Gradient (PCG) solver package input format: *A*, for each simulation; *B*, for each stress period (optional); and *C*, explanation of input variables used by the keywords.

*C*

| | |
|---|---|
| Text | is an optional character variable (699 characters) that is written to LIST file. |
| MXITER | is the maximum number of outer iterations. For a linear problem, MXITER should be 1 unless more than 50 inner iterations are required, when MXITER could be as large as 10. For nonlinear problems, a larger number is required.<br>If MXITER in Data Item 3 is set to -1, then the previous stress period's convergence criteria are reused, and the rest of Data Item 3 variables and Data Item 4 are not read. |
| ITER1 | is the number of inner iterations. Typically, a value between 30 and 100 is sufficient. |
| NPCOND | is the flag used to select the matrix conditioning method. Accepted values are<br>1 is for modified incomplete Cholesky (for use on scalar computers);<br>2 is for polynomial (for use on vector computers or to conserve computer memory). |
| IHCOFADD | is a flag that determines what happens to an active cell surrounded by dry cells. Accepted values are<br>0 – Cell converts to dry regardless of the HCOF value. (Former default before option).<br>1 – Cell converts to dry only if the HCOF is 0 (no head-dependent stresses or storage terms). |
| HCLOSE | is head-change criterion for convergence, in model units of length [L].<br>If the maximum absolute value of head change from all nodes during an iteration is less than or equal to HCLOSE and RCLOSE is also satisfied, the iteration stops |
| RCLOSE | is residual criterion for convergence, in model units of cubic length per time [$L^3/T$].<br>If the maximum absolute value of the residual at all nodes during an iteration is less than or equal to RCLOSE and HCLOSE is also satisfied, the iteration stops |
| RELAX | is relaxation parameter when NPCOND = 1. RELAX is not used if NPCOND is not 1. Usually, RELAX = 1.0, but sometimes values of 0.99, 0.98, or 0.97 may improve speed. |
| NBPOL | is read, but not used if NPCOND ≠ 2. If NPCOND = 2, then NBPOL indicates whether the estimate of the upper bound on the maximum eigenvalue is 2.0 (set NBPOL = 2), or whether the estimate is to be calculated (set NBPOL ≠ 2).<br>Convergence is generally insensitive to this parameter. |
| IPRPCG | is the printout interval for PCG. If IPRPCG is equal to 0, it is changed to 999.<br>The maximum head change (positive or negative) and residual change are printed for each iteration of a time step whenever the time step is an even multiple of IPRPCG. This prints out also at the end of each stress period regardless of the value of IPRPCG. |
| MUTPCG | is a flag that controls printing of convergence information from the solver to the LIST file. Accepted values are<br>0 – to print tables of maximum head change and residual each iteration.<br>1 – to print only the total number of iterations.<br>2 – to not print convergence information.<br>3 – to print only when convergence fails level 0 information. |
| DAMPPCG | is the damping factor. Set to 1 to indicate no damping or set to<br>>0 – to specify the damping factor applied to steady-state and transient stress periods.<br><0 – to only apply dampening, as -1×DAMPPCG, to steady-state stress periods. |
| DAMPPCGT | is only read if DAMPPCG<0 and is the damping factor for only transient stress periods. |

**Figure 3.44.** —Continued

## PCG and NWT Solver Loading of Convergence Criteria by Stress Period

The PCG and NWT solver packages were modified to allow specification of the convergence criteria by stress period. This feature allows strict tolerances for stress periods that require high accuracy, whereas stress periods that are not as important for obtaining good mass balances can have relaxed tolerances. This customization of solver tolerances by stress period can improve the speed of the simulation.

Input for the PCG and NWT solvers remains the same, except the solver checks whether there are additional convergence criteria specified on the line after the normal input. If the criteria are present, then they are read in at the beginning of each stress period, except the first period (the first stress period is specified by the normal solver-input structure). If any of the input variables fail to be read in the subsequent stress-period information (for example, a blank line or text is found where a number is expected), then the solver writes a warning to the Listing File, stops reading further stress-period-specific solver information, and uses the previously accepted solver information for the remainder of the simulation. This feature has a similar input structure to that of the HFB2 package described in Hanson and others (2014). Input instructions for a modified PCG solver is shown in figure 3.44: *A* is an example of the standard format for the PCG input files, and *B* is read for each additional stress period for which different convergence criteria are read. Items written in blue text can change with each stress period, whereas items written in red text must match what was specified in original input (for example, you cannot change the PCG preconditioner NPCOND in a single simulation run). Figure 3.45 is NWT input structure that includes the new stress period input.

***A***

**FOR FIRST STRESS PERIOD OR ALL STRESS PERIODS**

0. [#Text]

Item 0 is optional—"#" must be in column 1. Item 0 can be repeated multiple times.

1. HEADTOL FLUXTOL MAXITEROUT THICKFACT LINMETH IPRNWT IBOTAV **OPTIONS**
[DBDTHETA DBDKAPPA DBDGAMMA MOMFACT BACKFLAG [MAXBACKITER BACKTOL BACKREDUCE]]

If LINMETH = 1 and **OPTIONS** contains "**SPECIFIED**" then read Item 2 .

2. [MAXITINNER ILUMETHOD LEVFILL STOPTOL MSDR]

If LINMETH = 2 and **OPTIONS** contains "**SPECIFIED**" then read Item 3 .

3. [IACL NORDER LEVEL NORTH IREDSYS RRCTOLS IDROPTOL EPSRN HCLOSEXMD MXITERXMD]

***B***

**OPTIONAL FOR EACH SUBSEQUENT STRESS PERIOD**

4. HEADTOL FLUXTOL MAXITEROUT THICKFACT LINMETH IPRNWT IBOTAV **OPTIONS**
[DBDTHETA DBDKAPPA DBDGAMMA MOMFACT BACKFLAG [MAXBACKITER BACKTOL BACKREDUCE]]

**Figure 3.45.** Newton (NWT) solver package input format: *A*, for each simulation; *B*, for each stress period (optional); and *C*, explanation of input variables used by the keywords. [Note that each data item should be specified on a single line of text in the input file.]

*C*

| | |
|---|---|
| Text | is an optional character variable (699 characters) that is written to LIST file. |
| HEADTOL | is head-change criterion for convergence, in model units of length [L]. |
| FLUXTOL | the residual criterion for convergence, in model units of cubic length per time [$L^3/T$]. |
| MAXITEROUT | is the maximum number of outer iterations allowed. |
| THICKFACT | is threshold, specified as a fraction of the cell thickness, that switches from varying conductance linearly with respect to saturated thickness to varying it parabolically. For example, a 15-meter-thick cell with a THICKFACT = 0.1 indicates that when the saturated thickness is less than 1.5 meters, the conductance varies parabolically. The THICKFACT fraction must be between $1\times10^{-6}$ and 1. |
| LINEMETH | is the flag used to select the matrix conditioning method. Accepted values are<br>1 – GMRES matrix solver is used.<br>2 – χMD matrix solver is used. χMD cannot be used with Local Grid Refinement (LGR). |
| IPRNWT | is a flag to print to the LIST file convergence output. Set to 0 to not print and >0 to print. |
| IBOTAV | is a flag that indicates whether corrections are made to groundwater head relative to the cell-bottom elevation if the cell is surrounded by dewatered cells. A value of 1 indicates that a correction be made, and a value of 0 indicates no correction is to be made. This influences the convergence speed and is problem dependent. |
| **OPTION** | is a set keywords (required and option) that set up solver parameters.<br>One, and only one, of the following options must be selected<br>A rule of thumb is to first try using **MODERATE** and then test **COMPLEX** or **SIMPLE**.<br>Often **SIMPLE** results in faster running models, but<br>with larger mass-balance errors or convergence problems later in the simulation. |
| **SPECIFIED** | indicates that the items enclosed in brackets, [ ], in data item 1 are specified. Also, depending on LINEMETH, data items 2 and 3 are specified. Note that Data Items 2 and 3 are only read once. |
| **SIMPLE** | sets up the NWT parameters that work well for nearly linear models. The option results in faster model runs than the other options but can yield erroneous results and poor estimates of mass balances for highly nonlinear models. This is best for models that are confined or consist of a single unconfined layer that is thick enough to contain the water table within a single layer. |
| **MODERATE** | sets up the NWT parameters that work well for moderately nonlinear models. This is used for models that include nonlinear stress packages and models that consist of one or more unconfined layers. |
| **COMPLEX** | sets up the NWT parameters that work well for nonlinear models. This is used for models that include nonlinear stress packages and models that consist of one or more unconfined layers representing complex geology and surface-water and groundwater interaction. |
| | The following options are optional, and one or all may be specified. |
| **CONTINUE** | indicates that simulation continues, even when convergence fails. This is identical to BAS option **NO_FAILED_CONVERGENCE_STOP**. |
| **THIN_CELL_CHECK** | indicates that NWT will remove model cells within the same layer that have a thickness less than 1 percent of the largest cells thickness. This was the default NWT behavior before this option was added. Since, it was removed, the option was added to enable to legacy feature. |

**Figure 3.45.** —Continued

***C** (continued)*

The following variables in data item 1 are read only if **OPTION** includes the keyword **SPECIFIED**.

| | |
|---|---|
| DBDTHETA | is a coefficient used to reduce the weight applied to the head change between nonlinear iterations. DBDTHETA is used to control oscillations in head. Values range between 0.0 and 1.0, and larger values increase the weight (decrease under-relaxation) applied to the head change. |
| DBDKAPPA | is a coefficient used to increase the weight applied to the head change between nonlinear iterations. DBDKAPPA is used to control oscillations in head. Values range between 0.0 and 1.0, and larger values increase the weight applied to the head change. |
| DBDGAMMA | is a factor used to weight the head change for iterations n – 1 and n. Values range between 0.0 and 1.0, and greater values apply more weight to the head change calculated during iteration n. |
| MOMFACT | is the momentum coefficient "m" and ranges between 0.0 and 1.0. Greater values apply more weight to the head change for iteration n. |
| BACKFLAG | is a flag used to specify whether residual control is to be used. A value of 1 indicates that residual control is active, and a value of 0 indicates residual control is inactive. |
| MAXBACKITER | is the maximum number of reductions (backtracks) in the head change between nonlinear iterations. A value between 10 and 50 works well. MAXBACKITER is only read if BACKFLAG > 0. |
| BACKTOL | is the proportional decrease in the root-mean-squared error of the groundwater-flow equation used to determine if residual control is required at the end of a nonlinear iteration. BACKTOL is only read if BACKFLAG > 0. |
| BACKREDUCE | is a reduction factor used for residual control that reduces the head change between nonlinear iterations. Values can be between 0.0 and 1.0, and smaller values result in smaller head-change values. BACKREDUCE is only read if BACKFLAG > 0. |

Data Sets 2 and 3 are only read once. Please see Niswonger and others (2011) for the exact meaning.

**Figure 3.45.** —Continued

## References Cited

Hanson, R.T., Boyce, S.E., Schmid, W., Hughes, J.D., Mehl, S.M., Leake, S.A., Maddock, T., III, and Niswonger, R.G., 2014, One-water hydrologic flow model (MODFLOW-OWHM): U.S. Geological Survey Techniques and Methods 6–A51, 120 p., https://doi.org/10.3133/tm6A51.

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 6–A16, variously paginated, https://pubs.usgs.gov/tm/2005/tm6A16/.

Niswonger, R.G., Panday, S., and Ibaraki, M., 2011, MODFLOW-NWT—A Newton formulation for MODFLOW-2005: U.S. Geological Survey Techniques and Methods 6–A37, 44 p., https://pubs.usgs.gov/tm/tm6a37/pdf/tm6a37.pdf.