SUMMARY OF PRMS

Current Version 4.0.3 – June 19, 2017

Please see the updated tables on the PRMS software distribution page for summaries of changes to input and output variables, including identification of new and removed parameters and variables at:
http://wwwbrr.cr.usgs.gov/projects/SW_MoWS/PRMS.html

ABSTRACT

The need to assess the effects of variability in climate, biota, geology, and human activities on water availability and flow requires computer models that simulate the hydrologic cycle at a watershed scale. The Precipitation-Runoff Modeling System (PRMS) is a deterministic, distributed-parameter; physical process based modeling system developed to evaluate the response of various combinations of climate and land use on streamflow and general watershed hydrology. The primary objectives are: (1) simulate hydrologic processes including evaporation, transpiration, runoff, infiltration, and interflow as determined by the energy and water budgets of the plant canopy, snowpack, and soil zone on the basis of distributed climate information (temperature, precipitation, and solar radiation); (2) simulate hydrologic water budgets at the watershed scale for temporal scales ranging from days to centuries; (3) integrate PRMS with other models used for natural-resource management or with models from other scientific disciplines; and (4) provide a modular design that allows for selection of alternative hydrologic-process algorithms from the standard PRMS module library.

HISTORY

>   PRMS Version 4.0.3 (6/19/2017)
>   PRMS Version 4.0.2 (5/01/2016)
>   PRMS Version 4.0.1 (3/11/2015)
>   PRMS Version 4.0.0 (2/01/2015)
>   PRMS Version 3.0.5 (4/24/2012)
>   PRMS Version 3.0.4 (1/15/2012)
>   PRMS Version 3.0.3 (11/29/2012)
>   PRMS Version 3.0.2 (9/22/2012)
>   PRMS Version 3.0.1 (2/6/2012)
>   PRMS Version 3.0.0 (11/15/2011)
>   MMS Versions 1.0.0 through 1.2.1 (1991 through 2002)
>   PRMS Version 2.1 (1/17/1996)
>   PRMS Pre-version 2.1 (1983 - 1996)

SYSTEM REQUIREMENTS

PRMS is written in the Fortran 90 and C programming languages. The code has been used on personal computers running various forms of the Microsoft Windows operating system and Linux based computers.

PRMS Version 4.0.3 – March 23, 2017 - This release corrected major bugs in the `potet_pm` and `potet_pt` modules and minor bugs in the `soilzone` and `gwflow` modules. Screen and Model Output File output updated for readability and output of additional information. See the updated tables for updates to parameters, variables, modules, and dimensions at:
ftp://brrftp.cr.usgs.gov/pub/mows/software/prms/4.0.3/PRMS_tableUpdates_4.0.3.pdf

FUNCTIONALITY

The current version of PRMS includes the following modules and Utility Routines (listed in computation order):

`basin`	Basin Module
`climateflow`	Climate and flow parameters and variables Module
`cascade`	Cascading-Flow Module
`obs`	Observed-Data Module
`prms_time`	Time Variable Computation Utility Routine
`soltab`	Potential Solar-Radiation Module
`temp_1sta`	One-Station Air-Temperature-Distribution Module
`temp_laps`	Lapse-Station Air-Temperature-Distribution Module
`temp_dist2`	Inverse-Distance Air-Temperature-Distribution Module
`precip_1sta`	One-Station Precipitation-Distribution Module
`precip_laps`	Lapse-Station Precipitation-Distribution Module
`precip_dist2`	Inverse-Distance Precipitation-Distribution Module
`xyz_dist`	Multiple Linear Regression Precipitation and Temperature-Distribution Module
`ide_dist`	Inverse distance and elevation Precipitation and Temperature-Distribution Module
`climate_hru`	Pre-computed and Distributed Climate Module
`ddsolrad`	Degree-Day Solar-Radiation Distribution Module
`ccsolrad`	Cloud-Cover Solar-Radiation Distribution Module
`potet_jh`	Jensen-Haise Potential-Evapotranspiration Module
`potet_hamon`	Hamon Potential-Evapotranspiration Module
`potet_pan`	Pan-Evaporation Potential-Evapotranspiration Module
`potet_hs`	Hargreaves-Samani Potential-Evapotranspiration Module
`potet_pt`	Priestley–Taylor Potential-Evapotranspiration Module
`potet_pm`	Penman–Monteith Potential-Evapotranspiration Module
`transp_frost`	Frost-Based Active Transpiration Period Module
`frost_date`	Frost Date Preprocess Utility Routine
`transp_tindex`	Temperature Index Based Active Transpiration Period Module
`intcp`	Precipitation-Interception Module
`snowcomp`	Snow Module
`srunoff_smidx`	Non-linear Source Area Surface-Runoff and Infiltration Module
`srunoff_carea`	Linear Source Area Surface-Runoff and Infiltration Module
`soilzone`	Soil-Zone Module

gwflow	Groundwater Reservoir Module
subbasin	Subbasin Module
routing	Stream Network Computations Routing Utility Routine
strmflow	Streamflow Module
muskingum	Muskingum Streamflow Routing Module
strmflow_in_out	Streamflow Routing Module with inflow equals outflow for each segment
strmflow_lake	Streamflow Routing Module with on-channel lakes
water_balance	Water Balance Debug Utility Routine
nhru_summary	Write User-Selected HRU-based Variables to CSV File Module
prms_summary	Write Watershed Summary Basin-averaged Variables to CSV File Module
basin_sum	Watershed Flow-Summary Module
map_results	Map Based Output Module
write_climate_hru	Generate Climate-by-HRU Files Preprocess Utility Routine

BUG FIXES – by module:

potet_pt
- The computation of the net long wave radiation was corrected.

potet_pm
- The computation of the net long wave and net radiation coefficients used to compute potential ET were corrected.

muskingum
- If K_coef was set < 1.0, it was possible to get a divide by 0 under rare conditions, this was corrected.

soilzone
- Changed declared dimensions from **nhru** to **one** for variables *basin_cpr_stor_frac*, *basin_gvr_stor_frac*, *basin_pfr_stor_frac*, *basin_soil_lower_stor_frac*, *basin_soil_rechr_stor_frac*, and *basin_sz_stor_frac*. This does not affect any computations, but, could cause unpredictable results for output of these variables in statvar and animation files and runtime graphs.
- Computation of *basin_pfr_stor_frac* was computed incorrectly using *pref_flow_stor* instead of *pfr_stor_frac*. This does not affect any other computations.

gwflow
- The determination on whether to add storage on the basis of the value of **gwstor_minarea** is done after adding recharge instead of before.

basin
- If an HRU had open and closed surface-depression storage portions, computations were incorrect, this was corrected.

SMALL CHANGES THAT DO NOT AFFECT ANY COMPUTATIONS

call_modules
- Updated output to screen and to the model output file for improved readability and added output of the Parameter File name.

cascade
- Removed checks for specification of farfield flow as farfield computations are not supported.

subbasin
- Changed units in variable descriptions for *subinc_wb* and *subinc_deltastor* from cfs to inches.

map_results
- Messages for incomplete or excess accounting for mapping between HRU map and target map includes more information.

basin

- Summary statistics for areal portions and date and times reformatted.

NEW FUNCTIONALITY

`nhru_summary`
- Added output yearly total and mean yearly time series on the basis of control parameter **nhruout_freq** (1 = daily, 2 = monthly, 3 = both, 4 = mean monthly, 5 = mean yearly, 6 = yearly total); daily files have the suffix .csv; monthly files have the suffix _monthly.csv; mean monthly have the suffix _meanmonthly.csv; and mean yearly and yearly total have the suffix _yearly.csv.

NEW VARIABLES

- Added variable *basin_swrad*, which is set equal to *basin_potsw*. This change impacts modules `ccsolrad`, `ddsolrad`, `climate_hru`, and `climateflow`.
- Added `subbasin` variables: *subinc_rain*, *subinc_snow*, *subinc_stor*, *subinc_recharge*, *subinc_szstor_frac*, and *subinc_capstor_frac*.
- Added `basin_sum` variables: *basin_swrad_yr*, *basin_swrad_tot*, and *basin_swrad_mo*, which are added to summaries in the **model_output_file** on the basis of selected print options.

PRMS Version 4.0.2 – July 29, 2016 - This release added new parameters and variables, fixed bugs, added more checks for valid input values, general code clean up, mostly to reduce mixed floating-point computations by changing some variables to double precision, some to single precision, and using FORTRAN intrinsic functions to convert variables prior to mixed-precision computations. Additional information about input and output files and simulation period are printed to the screen. See the updated tables for updates to parameters, variables, modules, and dimensions at: ftp://brrftp.cr.usgs.gov/pub/mows/software/prms/4.0.2/PRMS_tableUpdates_4.0.2.pdf

BUG FIXES – by module:

`potet_pt`
- The code incorrectly used the value of the average HRU air temperature in degrees Fahrenheit when it was supposed to be in degrees Celsius in one equation. The equation is correct in the PRMS-IV documentation report.

`potet_pm`
- The equation is different from the PRMS-IV documentation report.

`potet_hs`
- The kt coefficient was first being computed each day using an equation based on average temperature as described in http://www.zohrabsamani.com/research_material/files/Hargreaves-samani.pdf. Then this value was multiplied by the hs_krs parameter. Now, only the hs_krs parameter is used in the Hargreaves-Samani equation, which matches the PRMS-IV documentation. The kt coefficient equation can be used to estimate the hs_krs parameter: coef_kt = 0.00185*(temp_dif**2) − 0.0433*temp_dif + 0.4023, where temp_dif is the difference between the HRU maximum and minimum temperature in units of degrees Celsius.

`precip_laps`
- Arrays for parameters **rain_adj_lapse** and **snow_adj_lapse** were not allocated causing application to fail.

`routing`

- Variables *seginc_potet, seginc_gwflow, seginc_ssflow, seginc_sroff, seginc_swrad, seg_gwflow, seg_ssflow, and seg_sroff* were computed if the cascading flow option was active, which was an error if parameter **hru_segment** was not specified. These variables are not computed if the cascading flow option is active.
- The value of *seg_lateral_inflow* was not set correctly when the cascading flow option was active; it is now set to *strm_seg_in*.

subbasin
- Subbasin storage did not include surface-depression or lake storage.

climate_hru
- Variable *basin_obs_ppt* was set to 0.0 if precipitation values are not input in a CBH file; thus, an error occurred if parameter **precip_module** was not specified equal to climate_hru.

map_results
- Yearly output was not computed when parameter **mapvars_freq** = 5.
- Corrected IF blocks that determine whether or not parameters **gvr_cell_pct** and **gvr_hru_id** were needed; **gvr_cell_pct** needed when **nhrucell** not equal **ngwcell** and **gvr_hru_id** needed when **nhru** not equal **nhrucell**.

gwflow
   Variable *basin_lake_seepage* was computed incorrectly by using incorrect values of **hru_area**.

basin_sum
- The value of the yearly observed streamflow or monthly value of computed basin streamflow and monthly observed streamflow was not printed for **print_type** = 2.
- Detailed output did not have the right line length, thus missing a value.
- Yearly detailed output now includes evaporation for lakes and interception storage.

ide_dist
- Computations of inverse distance and elevation allowed to be less than 0 for precipitation computations so that they are consistent with computations for temperature computations.

srunoff_smidx and srunoff_carea
- Variable *contrib_fraction* was declared as dimension **one** and data type double when it should be dimension **nhru** and data type single. This error could have caused memory problems if the variable was output in a Statistics Variable (statvar) or Animation File, or using the PRMS GUI runtime plots, but would not affect any computations.

soilzone
- Computation of variable *soil_lower_ratio* was set incorrectly to *soil_moist*/**soil_moist_max** instead of *soil_lower/soil_lower_stor_max* and values of *soil_lower_ratio* for lake and inactive HRUs were not initialized to 0. No other computations are affected.
- In the compute_soilmoist routine, the check for infiltration water in excess of **soil_moist_max** minus *soil_to_gw* (local variable *excs*) was > *Infil* should have been a check for *excs>Infil*perv_frac* as *excs* is computed for the whole HRU area and *infil* for only the pervious area on an HRU; this condition was not likely to have occurred.
- Make sure any flow from gravity to preferential-flow reservoirs is set to 0 or current value (previous versions could have used value from previous HRU if current HRU in loop does not have preferential flow reservoir).

basin
- Corrected code to ensure a divide by zero is not permitted for the condition that there are closed depressions but not open depressions; that is, if any values for parameter **dprst_frac_open** are specified = 0.
- If the first value of parameter **dprst_area** was specified < 0 or not included in the Parameter File, the values of **dprst_frac_hru** (default value = 0) were ignored if specified. Thus, if **dprst_frac_hru** was not included in the Parameter File no surface-depression storage would be simulated; this check should

have been reversed. Use of parameter **dprst_area** is now deprecated, so users should now use **dprst_frac_hru** and specify values >= 0. The default value of **dprst_frac_hru** is now -1.0.

`snowcomp`

- Added check to be sure *snowcov_area* is not equal to zero when *pkwater_equiv* is greater than 0. This would be very rare and only possible when the second value of a snow-depletion curve was specified equal to zero.
- Checks were added to be sure *pkwater_equiv* is set to 0.0 if it is computed as a very small negative value, which would be very rare.
- Added check for when *pkwater_equiv* < 1.0E-10 and *snowcov_area* > 0; if true, set *snowcov_area* to 0. This is possible when the energy is enough to melt the snowpack on the previous time step. This bug affected computations for impervious and surface-depression storage evaporation for days without snow and snow depletion curves that specified snow-covered area exists when the snow-water equivalent is equal 0.
- In calin function, *snowcov_area* was set to 0 when it should not be because it is the snow cover after adding precipitation and before any melt or sublimation; this condition affects computations for impervious and surface-depression storage evaporation for days with snow.
- If snowpack exists and the snowpack density (*pk_den*) <= 0, be sure *pk_den* and the snowpack depth (*pk_depth*) have a value based on parameter **den_max** in run function and calin function; this condition would be rare.
- In routine calin, added check for the value of *freeh2o − pwcap > pkwater_equiv*; if so, use the value of *pkwater_equiv*. This condition would be rare.
- In routine calin, the local variable *apk_ice* did not have a value if *snowcov_area* <= 0; if true, it is set to 0. This condition would be rare.

`water_balance` module:

- Full array for *hru_sroffp* and *hru_sroffi* instead of HRU value printed for water-budget issue for **srunoff** with cascades active.
- Variable *soil_to_ssr* was not included in HRU **soilzone** water-budget computation. This variable is needed because the variable *cap_waterin*, actual water into capillary reservoir, replaced *cap_water_maxin*, the maximum potential water into the capillary reservoir.

UPDATES THAT MIGHT PRODUCE SLIGHT CHANGES IN ASSOCIATED COMPUTATIONS – general:

- Some single-precision variables were changed to double-precision variables and vice versa; also, most modules now use FORTRAN intrinsic functions to explicitly designate mixed-precision computations. These changes are intended to limit the possibilities of different results on different computers and compilers, to provide more consistent floating-point comparisons, and to have more consistent round-off issues. These updates could change memory requirements and execution time very slightly. Modules affected: `obs`, `cascade`, `ccsolrad`, `ddsolrad`, `climate_hru`, `temp_dist2`, `precip_dist2`, `ide_dist`, `xyz_dist`, `potet_hamon`, `intcp`, `snowcomp`, `srunoff_smidx`, `srunoff_carea`, `soilzone`, `gwflow`, `routing`, `water_balance`, `nhru_summary`, `map_results`, `prms_summary`, `subbasin`, `climateflow`, and `basin`.
- Small values (>0.0 and < 1.0E-05) of precipitation are used in computations; previously these values were assumed to be below round-off tolerance and set to 0. This affects modules `obs`, `precip_1sta`, `climate_hru`, `ide_dist`, `xyz_dist2`, and `precip_laps`.

- Small values (>0.0 and < 1.0E-06) of computed potential evapotranspiration were considered to be round-off error and set to 0; now those values are used in computations. This change affects modules `potet_hamon`, `potet_hs`, `potet_jh`, `potet_pan`, `potet_pm`, and `potet_pt`.
- `climateflow` module: Small values (>0.0 and < 1.0E-06) of computed mixed precipitation were considered to be round-off error and the event was set to all snow or all rain depending on the precipitation form, now those values are used in computations.
- Computation of saturation vapor pressure for module `potet_pm` now uses an equation by Irmak and others (2012; Journal of Hydrology, v. 420-421, p. 228), to be consistent with module `potet_pt`. This can reduce execution time.
- Small values (>0.0 and < 1.0E-04) specified for **hru_percent_imperv** and **dprst_area** are used in computations; previously these values were assumed to be below round-off tolerance and set to zero.
- Values of canopy interception computed between 0.0 and 1.0E-05 were considered to be round-off error and set to 0.0; these values are now left in the canopy. Module affected: `intcp`.
- Double precision values < 1.0D-15 are treated as zero instead of < 1.0D-10.
- Module `cascade`: changed check for excess GWR cascade fraction so that any values > 1 set to 1 instead of only > 1.00001. This change makes the check consistent with the HRU cascade fraction check.
- Module `snowcomp`:
  - instead of using < −1.0E-10 to check for round-off issues in some computations, < 0.0 is now used. Instead of using > 1.0E-06 to check for round-off issues in some places, > 1.0E-09 is now used.
  - values of snowpack water equivalent (*pkwater_equiv*) computed between 0.0 and 1.0E-09 were considered to be round-off error and set to 0.0; these values are now used.
  - Values < 0.0 are set to 0.0 with a warning message printed when control parameter **print_debug** > -1. This condition accounts for negative round-off error due to mixed precision computations and may occur under rare conditions.

NEW FUNCTIONALITY

The values of <u>all</u> parameters specified in the Parameter File are now checked to determine if they fall within the suggested minimum and maximum range. If values are specified outside the range, a warning message is issued. If the user wants to specify values out of the range, set control parameter **parameter_check_flag** = 0 to deactivate these checks. Previously, select parameters were checked within individual modules.
`nhru_summary`
- Added functionality to output monthly and mean monthly time series based on the value of new control parameter **nhruout_freq** (1 = daily, 2 = monthly, 3 = both, 4 = mean monthly), daily files have the suffix .csv; monthly files have the suffix _monthly.csv, and mean monthly have the suffix _meanmonthly.csv.
`gwflow` and `cascade`
- Added option to allow GWRs to be swales with new control parameter **gwr_swale_flag** (0 = not allowed; 1 = groundwater flow routed to groundwater sink; 2 = groundwater flow routed to stream network).
`cascade` module:
- Allow groundwater-reservoir cascades to be equal to HRU cascades with **cascadegw_flag** = 2.
`call_modules`
- Specification of control parameter **print_debug** = -1 reduces amount of screen output.
`ccsolrad`

- Removed restart subroutine so `ccsolrad` can be used in a restart simulation if `ddsolrad` was used for the antecedent simulation. The switch from `ddsolrad` to `ccsolrad` is still allowed.

`snowcomp` module:
- Code added to initialize snow states based on **snowpack_init** and related parameters in the "init" procedure.

`water_balance`
- Code related to GSFLOW removed.

`basin_sum` module:
- Labeled first value of reports as initial storage.
- Added print of water balance values to reports when print_type = 1
- Added print of *basin_intcp_stor* and *basin_lakeevap_yr* to yearly report when print_type = 2

`map_results` module:
- Stop if negative mapping fraction specified.
- Allow values > 0.0 and 1.0E-06 for parameter **gvr_cell_pct** instead of treat them as 0.0.

MODULES REMOVED

`check_nhru_params` – code for module moved to function in file utils_prms.f90 that is called within `call_modules`.

`cloud_cover` – code returned to module `ccsolrad`.

INPUT SPECIFICATION CHANGES:

`obs` and `climate_hru`
> Values in Data File or CBH File are not checked for being specified as not a number (NaN), thus user must ensure such values are not specified.

`climate_hru`
> Precipitation values specified in CBH Files less than 0.0 are flagged as an ERROR.

`precip_dist2`
> Values specified for parameter **psta_mon** less than or equal to 0 are flagged as an ERROR.

OUTPUT FILE REMOVED

The prms.log file is not generated as it is redundant with information written to the model output file. The log was present to produce a similar file as output by GSFLOW and included description and version of PRMS and modules.

PRMS Version 4.0.1 – March 11, 2015 - This release added new functionality, parameters, and variables, fixed bugs, added more checks for valid input values, general code clean up. The primary change is the addition of the flexible dimension option described below that allows flexibility for specification the spatial and temporal distribution of parameter values.

NEW MODULES (code taken mostly from existing modules)

`potet_pm`	Penman-Monteith potential evapotranspiration computation and distribution module, activated using control parameter **et_module**.
`check_nhru_params`	select **nhru**-dimensioned parameter values are checked to see if specified values are within the suggested range.
`prms_time`	module for time related computations, such as current time step, Julian day, and whether in summer or winter.
`routing`	computes variables used by stream flow modules, such as code related to fluxes associated with each segment.
`lake_route`	reads most lake related dimensions, parameters and computes some of variables used with lake computation modules.
`water_balance`	debug module that checks that water balances are maintained for the overall water budget, soil-zone computations, canopy interception computations, snow computations, surface runoff and impervious computations, and groundwater computations. Active when **print_debug** is specified equal to 1.
`nhru_summary`	produces output files of daily computed results for user-selected nhru-dimensioned variables in a CSV format. Results for each selected variable are written to a separate file.

GENERAL

1. Initialize procedure is completed prior to stopping for input parameter range errors. Previously, the simulation would stop after finding the first invalid parameter value. Thus, more invalid values can be caught in a single execution. This is activated using new control parameter parameter_check_flag, which is described below.
2. Updated many parameter and variable descriptions to be consistent with PRMS documentation manual.
3. Initialize procedure is completed prior to stopping for input parameter range errors. Previously, the simulation would stop after finding the first invalid parameter value. Thus, more invalid values can be caught in a single execution. This is activated using new control parameter **parameter_check_flag**, which is described below.
4. Updated many parameter and variable descriptions to be consistent with PRMS users' manual.
5. Use of initial conditions options (activated using control parameters **init_vars_from_file** and **save_vars_to_file**) does not write parameter values to the Initial Condition File. Parameter values specified in the Parameter File are read and used in a restart simulation. Storage initialization parameter values, parameters **soil_moist_init**, **soil_rechr_init**, **snowpack_init**, **ssstor_init**, and **gwstor_init**, are ignored in a restart simulation. Writing to and reading from Initial Conditions Files are removed from modules `potet_hamon`, `transp_frost`, `potet_jh`, `smbal_prms`, `ssflow_prms`, `soltab`, `precip_dist2`, `hru_sum_prms`, `ccsolrad`, `ddsolrad`, `precip_1sta_laps`, `prms_summary`, `subbasin`, `basin`, `cascade`, and `map_results`. Thus, these modules do not have to be active in a restart simulation if active in a spin-up (or antecedent) simulation, except for `basin`, which is always active.
6. New variable (*IGNOREPPT*=1.0E-5) added. It is used to check for values of measured or computed precipitation values that are considered too small to be actual precipitation. If a precipitation value is less than *IGNOREPPT* the value is set to 0.0.

7. New variable (*SMALLPARAM*=1.0E-4) added. It is used to check a few parameter values for being too small to be used and thus set to 0 a warning message is printed if **parameter_check_flag** specified equal to 0. Parameters checked include **pmn_mo**, **psta_mon**, and **lake_coef**. An error message is issued and execution stops if control parameter **parameter_check_flag** is specified equal to 1 for this check.

8. Code related to computations of far field flows (flow leaving the model domain boundary not through the stream network) was removed. Adding an extra stream segment to send flow outside the domain is the recommended replacement.

NEW FUNCTIONALITY ADDED TO MODULES

Flexible dimension option: Previously, parameters had only one option for the number of values (dimension(s)) specified in the Parameter File. Now, many parameters can be specified using the original dimension(s) or using compatible dimensions up to a maximum number of values based on the specified dimension(s). For example, some parameters having dimension of **nmonths** now have a maximum dimension of **nhru**,**nmonths**. Flexible dimensions for a parameter with a maximum dimension of **nhru**,**nmonths** are **one**, **nmonths**, **nsub**, **nsub**,**nmonths**, **nhru**, and **nhru**,**nmonths**. Flexible dimensions for a parameter with a maximum dimension of **nhru** are **one**, **nsub**, and **nhru**. Flexible dimensions for a parameter with a maximum dimension of **nssr** are **one**, **nsub**, and **nssr**. Flexible dimensions for a parameter with a maximum dimension of **ngw** are **one**, **nsub**, and **ngw**. PRMS will read the dimension, number of values, and values from the Parameter File. If the parameter is not specified at the maximum dimension, the parameter values will be automatically expanded to the maximum dimension by the code. Thus, the user has several options to specify the number of parameter values based on the spatial and temporal variability, available data, or for some other purpose. The maximum number of values for most parameters has not changed. Maximum parameter dimensions are identified in PRMS-IV updated tables 1-1 and 1-3 (ftp://brrftp.cr.usgs.gov/pub/mows/software/prms/4.0.1/PRMS_tableUpdates_4.0.1.pdf). The flexible dimension option was added to accommodate simulation of large model domains that required increased spatial and/or temporal distribution of parameter values. Additionally, the number of lines in Parameter Files can be significantly reduced by specifying a single (dimension **one**) or **nsub** values for parameters that have a constant value for all HRUs or subbasins. This capability may change results when dimensions are specified greater than the original dimension(s). If the parameter dimensions are not changed, results should be the same. However, some computations in ddsolrad and ccsolrad are based on variables for each HRU rather than basin-wide variables, so the potential solar radiation (variable *swrad*) can be significantly different than previous versions for large model domains.

Parameters affected with maximum dimension indicated are: **hamon_coef**(**nhru**,12) in module potet_hamon, **jh_coef**(**nhru**,12) in module potet_jh, **radadj_intcp**(**nhru**,12), **radadj_slope**(**nhru**,12), **dday_intcp**(**nhru**,12), **dday_slope**(**nhru**,12), **tmax_index**(**nhru**,12), and **radmax**(**nhru**,12) in module ddsolrad, **radadj_sppt**(**nhru**), **radaj_wppt**(**nhru**), **ppt_rad_adj**(**nhru**,12), **crad_coef**(**nhru**,12), **crad_exp**(**nhru**,12), and **radmax**(**nhru**,12) in module ddsolrad, **epan_coef**(**nhru**,12) in modules potet_pan and intcp, **potet_sublim**(**nhru**) in modules snowcomp and intcp, **frost_temp**(**nhru**) in module frost_date), **tmax_lapse**(**nhru**,12) and **tmin_lapse**(**nhru**,12) in module temp_laps, **tmax_adj**(**nhru**,12) and **tmin_adj**(**nhru**,12) in modules temp_1sta, temp_dist2, ide_dist, xyz_dist2 and temp_laps, **tmax_allrain**(**nhru**,12), **tmax_allsnow**(**nhru**,12), and **adjmix_rain**(**nhru**,12) in modules precip_1sta, precip_dist2, climate_hru, ide_dist, xyz_dist and precip_laps, **adjust_snow**(**nhru**,12) and **adjust_rain**(**nhru**,12) in modules ide_dist and xyz_dist

melt_look(nhru), **melt_force(nhru)**, **tstorm_mo(nhru**,12), **cecn_coef(nhru**,12) , **emis_noppt(nhru**,12), and **freeh2o_cap(nhru**,12) in module `snowcomp`

Parameters **adj_by_hru**, **hru_subbasin**, **rain_sub_adj**, and **snow_sub_adj** that were specified for module `climate_hru` were removed as the flexible dimension option makes these unnecessary. **rain_adj** and **snow_adj** can have the dimensions (**nsub**,**nmonths**), so are used instead of **rain_sub_adj**, and **snow_sub_adj**. Parameter **hru_subbasin** is required if module `subbasin` is active.

An example of parameter with a maximum number of values equal to **nhru** that might have the same value for all HRUs is **carea_max**. In this case, values for each HRU can be specified to the default value as:

```
####
carea_max
1
one
2
0.6
```

`map_results`	added option to output daily mapped results (parameter **mapvars_freq** specified equal to 7). Notes: the units of output variables are equal to the units of the variable for **mapvars_units** specified equal to 0. Units for variables could be, for example, inches, inches/day, degrees Fahrenheit, and degrees Celsius. However, when the **mapvars_units** value is specified greater than 0, the units of all selected **map_results** output variables as specified by control parameter **mapOutVar_names** must have units of either inches or inches/day as the code only converts inches to a metric unit.
`climate_hru`	CBH files can be input as binary files with the identical information when new control parameter **cbh_binary_flag** specified equal to 1. Humidity and wind speed data can be input as CBH files. Potential ET can be adjusted from values input in CBH files using new parameter **potet_cbh_adj**.
`strmflow_lake`	the order of stream segments and lakes was specified by parameters input to the `cascade` module with **wtrbdy** prefix-named parameters. The module now uses the same parameters as other stream network routing modules, that is, **tosegment**, **hru_segment**, and **obsin_segment**. This change requires that each lake includes a stream segment as well as being specified as a lake HRU. A segment within a lake is specified using **segment_type**.
`obs` and `call_modules`	store dimension values in Initial Conditions Files for a spin-up simulation and are checked to be sure they are the equal to the dimensions for a restart simulations.
`prms_summary`	added output of the first value of the variable *streamflow_cfs* array to the CSV file (control parameter **csvON_OFF** specified equal to 1); if the value of dimension **nobs** is specified equal to 0, then the value 0.0 is output for *streamflow_cfs*.

NEW PARAMETERS

Parameters specified in the Control File:

parameter_check_flag	0 means treat some parameter range checks as WARNINGs as done for most parameters in previous versions, if specified as 1 these checks are treated as ERRORs, if specified as 2 the parameters are checked and then the simulation stops even if no ERRORs, default = 1.

cbh_check_flag	0 means do not check values in CBH file; 1 means to check for invalid values, such as NaN, character strings, < lower bound, and > upper bound, plus end of file during a simulation and non-sequential time series, such as not having the correct number of days in each year, default = 1. Specifying **cbh_check_flag** equal to 0 should only be done after the CBH file(s) are verified using **cbh_check_flag** specified equal to 1. Setting **cbh_check_flag** equal to 0 can reduce execution time.
cbh_binary_flag	0 means CBH files are text files as in the past; 1 means the CBH files are in binary format with same order of values as text file version.
nhruOutVar_names	array of variable names to write to CSV files each day.
nhruOutBaseFileName	string to define the prefix for each output file for the nhru_summary module, one for each variable; this can be a full or relative path.
nhruOutON_OFF	flag to activate nhru_summary module (0 means do not use module; 1 activates module, default = 0).
nhruOutVars	number of output variables, i.e., the number of variable names specified using **nhruOutVars_names**.
csv_output_file	pathname of output file written by the prms_summary module.
csvON_OFF	flag to activate the prms_summary module (0 means do not use module; 1 activate module, default = 0).
humidity_cbh_flag	flag to specify to read a CBH file with humidity values (0=no; 1 =yes, default = 0).
humidity_day	file name of the humidity CBH file; this can be a full or relative path.
windspeed_cbh_flag	flag to specify to read a CBH file with wind speed values (0=no; 1 =yes, default = 0).
windspeed_day	file name of the wind speed CBH file; this can be a full or relative path.
print_debug	new option (print_debug specified equal to -1) added to this existing parameter that minimizes warning messages and other messages printed to the screen during a simulation, such as when values of variables *slow_stor* are computed greater than *pref_flow_thrsh*. This can increase execution efficiency.

Parameters specified in the Parameter File(s):

dprst_et_coef(nhru)	surface-depression evaporation adjustment factor, range 0.0 to 1.0, default = 1.0.
sro_to_dprst_imperv(nhru)	impervious surface runoff fraction captured by surface depressions within each HRU. Related parameter **sro_to_dprst(nhru)** is pervious surface runoff capture fraction captured by surface depressions within each HRU, range 0.0 to 1.0, default = 0.2.
segment_flow_init(nsegment)	initial flow rate in each segment, in cfs, range 0.0 to 1.0E07, default = 0.0.
segment_type(nsegment)	Segment type—0=segment; 1=diversion; 2=lake; 3=replace inflow, default = 0.
hs_krs(nhru,nmonths)	calibration coefficient for module potet_hs; replaces removed parameter **potet_coef_hru_mo(nhru,nmonths)**, range 0.005 to 0.06, default = 0.0135.
pt_alpha(nhru,nmonths)	calibration coefficient for module potet_pt; replaces removed parameter **potet_coef_hru_mo(nhru,nmonths)**, range 1.0 to 2.0, default = 1.26.

potet_cbh_adj(nhru,nmonths) calibration coefficient for values specified in a **potet_day** CBH file, range 0.5 to 1.5, default = 1.0.

snowpack_init(nhru) initial snowpack-water equivalent, in inches, range 0.0 to 500.0, default = 0.0.

NEW COMPUTED VARIABLES (available depending on which computation options are active):

canopy_covden(**nhru**)	current value of cover density fraction for each HRU based on whether HRU is in transpiration mode.
hru_intcpevap(**nhru**)	canopy evaporation for each HRU, in inches.
hru_storage(**nhru**)	sum of all storage for all water-holding reservoir for each HRU, in inches.
lake_seepage_gwr(**nhru**)	the seepage from lakes to each GWR in units of inches.
gw_seep_lakein(**nhru**)	dimensioned changed from **nhru** to **nlake**, in inches.
dprst_stor_hru(**nhru**)	average surface depression storage for each HRU, in inches.
basin_contrib_fraction	basin-area weighted average contributing area fraction of pervious area.
contrib_fraction(**nhru**)	contributing area fraction of the pervious area of each HRU.
basin_lake_stor	basin-area weighted average precipitation into all lakes, in inches.
basin_lakeevap	basin-area weighted average evaporation from all lakes, in inches.
unused_potet(**nhru**)	unsatisfied potential evapotranspiration after all ET computations, in inches.
basin_segment_storage	basin-area weighted average total storage in each segment, in inches.
segment_delta_flow(**nsegment**)	cumulative difference of flow into minus outflow for each stream segment, in cfs.
seginc_potet(**nsegment**)	potential ET associated with each segment, in inches.
lake_inflow(**nlakes**)	total inflow into each lake, in cfs.
lake_outflow(**nlakes**)	total evaporation and seepage from each lake, in cfs.
lake_lateral_inflow(**nlakes**)	total lateral flow into each lake, in cfs.
hru_streamflow_out(**nhru**)	total flow to stream network for each HRU, in cfs.
hru_hortn_cascflow(**nhru**)	name change for variable *hru_hortonian_cascadeflow*, in inches.
snowdepth(**nsnow**)	name change of variable *snow*, in inches.
lake_outcfs(**nsegment**)	name change of variable *lake_outq*, in cfs.

BUG FIXES – by module:

`snowcomp`	Computation of snowpack density and snowpack depth are computed based on a finite difference approximation, which produced slightly incorrect results for days when new snow falls when a snow pack does not exist. This bug could produce significant error when the new snow fall is large compared to the existing snow pack. There was the possibility that the snowpack water equivalent was computed as a value < 0 when the amount of free water was less than the computed amount of free water that the snowpack could hold. Though this condition is likely very rare, it could have resulted in very slight differences in results in the value of variables *pkwater_equiv*, *pk_depth*, and *snowmelt*.
`ide_dist` and `xyz_dist`	measured precipitation values were modified from those in the Data File, thus output values in runtime graphs or files could be different than those in the Data File; a local array is used instead.

`basin_sum`	water balance computations did not include water in stream segments when using Muskingum flow routing or surface depression storage, if any. Water balance computations are only computed when control parameter print_debug = 1. Used variable *basin_stflow_in* instead of *basin_stflow_out* to set the basin outflow variable *basin_cfs*, this bug only occurred when the `muskingum` module was active.
`basin`	The contributing area to each stream segment was computed incorrectly for some cases. Error check added for specification of any tosegment value equal to that segment, if this were done the code would go into an infinite loop. Checks added for valid values of tosegment; error if > nsegment, <0, or equal to the segment id. Setting of local variable *Hru_elev_feet* was not set when parameter **elev_units** was specified as `1`; this only affected computations when module `potet_pt` is active. Print of fraction of impervious, pervious, and depression storage were labeled as area instead of as fractions of the active basin area.
`gwflow`	Values of parameter **gwflow_coef** specified less than `1.0` are allowed and if specified > `1.0` a warning is issued. Any water added due to specified values of parameter **gwstor_min** was added twice. Error message for the condition that an HRU is specified as a lake and the associated value of parameter **lake_hru_id** is specified equal to `0`, printed the incorrect GWR Identification number. If groundwater discharge to a lake was computed greater than the available groundwater storage, the value was not limited by the available storage; now the discharge is set to the available groundwater storage and a warning message is issued. Computation of lake seepage was removed from the initialize procedure so the initial lake elevation and storage is based only on input parameters instead of after lake elevation and groundwater discharge computations as computed in the run procedure. Warning checks of parameters used incorrect array index. If any HRUs were specified as inactive then the last number of inactive HRUs values were not checked. Seepage from lakes used the incorrect array index, thus, was set incorrectly. *Basin_lake_seep* was divided by the total active area of the model domain twice, thus the value was incorrect.
`soilzone`	Error check added for interflow computation of the equation SQRT(**coef_lin****2.0+4.0***coef_sq***ssres_in) = 0.0, if true a divide by 0 would have occurred, this would be a very rare condition. Corrected setting of variable *soil_zone_max* to account for parameter **soil_moist_max** only being applicable to the pervious area of each HRU, this does not affect other soilzone computations, just the values of variables *soil_zone_max* and *soil_moist_frac*, which are computed results that are not used in other computations. Water balance check when control parameter **print_debug** = 1 was incorrect when swale HRUs are present.
`ccsolrad`, `ddsolrad`, `ccsolrad_prms`, and `ddsolrad_prms` assumed it was always winter for Southern Hemisphere applications.	

`write_climate_hru`   if a transpiration CBH file was specified and the value for control parameter **transp_module** was specified as climate_hru_mo, the CBH file was not correctly produced.

`frost_date`   write of parameter names was incomplete as the output string was not allotted enough characters. Fix was made in functions write_integer_param and write_real_param that are found in the file utils_prms.f90. If a spring or fall frost was not found in a year a divide by zero error was possible. The values for output parameters **spring_frost** and **fall_frost** were output as solar days instead of calendar day, i.e., 10 days are now added to the computed solar day. Function julian_day, found in the file utils_prms.f90, incorrectly computed solar day.

`transp_frost`   Function julian_day, found in the file utils_prms.f90, incorrectly computed solar day

`lake_route` and `strmflow_lake` if an HRU is a lake, the value for the local variable *gwr_type* for that HRU was incorrectly set to 2; it is now set to 1.

`srunoff`   The amount of water used to compute Hortonian runoff from cascading surface runoff was computed as a value on pervious areas only instead of the amount of water from the entire contributing HRUs; thus could produce significant error when impervious fraction of HRUs is large. The values of variables *hru_sroffi* and *basin_sroffi* were not set correctly when parameter **imperv_stor_max** was specified equal to 0.0 for impervious portions of HRUs.

CHANGES THAT MIGHT PRODUCE SLIGHT CHANGES IN ASSOCIATED COMPUTATIONS – general:

- Most double precision variables changed to single precision to provide more consistent floating-point comparisons and have more consistent round-off issues. Change reduces memory requirements up to 10% and slight decrease in execution time.

- Changed variables *seginc_ssflow, seginc_sroff, seginc_gwflow, seginc_swrad, seg_outflow, seg_inflow, lake_area, dprst_sroff_hru, dprst_seep_hru,* and *hortonian_lakes* changed from single to double precision as they are sums for sets of HRUs. Only *seg_outflow* and *seg_inflow* are used in computations in multiple modules, while; the others only are computed for output purposes.

- Changed check for very small values from 5.0E-6 to 1.0E-6 to better represent round-off error.

CHANGES THAT MIGHT PRODUCE SLIGHT CHANGES IN ASSOCIATED COMPUTATIONS – by module:

`cascade`   tolerance used to check for the fraction leaving an HRU or GWR adding up to exactly 1.0 changed from 1.00001 to 1.001. If this check finds an issue the cascade links are adjusted.

`potet_pan, potet_jh, potet_hamon, potet_hs, potet_pt,` and `potet_pm`
        now sets potential evapotranspiration to 0.0 for values computed less than 1.0E-6 instead of only when a computed value was negative. Thus, very small computed potential evapotranspiration values are ignored.

`potet_pt`	constant used to compute saturation vapor pressure changed from 17.269 to 17.26939. Equation to compute the latent heat of vaporization were computed in units of Calories/gram and then converted to kilojoules/kilogram. Now they are computed in units of kilojoules/kilogram.
`ccsolrad`	now sets computed canopy density to 0.0 for values < 1.0E-6 instead of only if computed value was negative. Thus, very small computed canopy density values are ignored.
`intcp`	values of parameters **covden_win** or **covden_sum** specified < 1.0E-4 and some computations that result in values very < 1.0E-5 are not reset to 0.0 as previously done.
`snowcomp`	set some and compares other computations to being less than 1.0E-6 instead of less than 0.0, which can allow for very small values of some variables. Canopy density on each HRU for each time step is used in computations instead of only values of parameter **covden_win**. If parameters **den_init** and/or **den_max** values are specified less than 1.0E-06, they are set to 1.0E-06 instead of 0.1 and 0.6, respectively.
`srunoff`	value of variable *hru_hortn_cascflow* was not reset to 0.0 for each time step; computed value is not used in other computations and is only available when cascading flow is active. Removed check for computed values of *imperv_stor* less than 0.0 and then reset to 0.0 to allow for round-off error bias to be reduced.
`gwflow`	computation of surface-depression storage fractions for round-off issues would sometimes produce very small closed depression storage fractions, which is now fixed.
`soilzone`	Added checks for computed infiltration less than 0.0 and computed interflow coefficient equal to 0.0, these would not likely occur.
`ddsolrad`	if computed radiation adjustment factor based on parameters **dday_slope** and **dday_intcp** is greater than the value of parameter **radmax**, it is set to **radmax** rather than as computed.
`frost_date`	the sum of local variables *fallFrostSum* and *springFrostSum* computed as floating point values instead of integer prior to rounding to nearest integer. These values were set to the truncated integer values, they are now rounded, that is, set to the nearest integer. Maximum value of the fall frost changed from 366 to 365 and minimum value of spring frost changed from 0 to 1. Computation of the *basin_fall* and *basin_spring* uses rounding of the integer value instead of truncation.
`transp_tindex`	all computations use degrees Fahrenheit instead of checking temp_units and converting computations to Fahrenheit, this should not produce a noticeable change in results, but, does increase execution efficiency. When the value of parameter **transp_end** equals the current month and the current day is the first day of the month transpiration is turned off (variable *transp_on* is set to 0) and the related local variables, *transp_check* and *tmax_sum*, are set to 0. After this check, if the current month equals the value of parameter **transp_beg**, *transp_check* and *tmax_sum* are checked to determine if transpiration needs to be turned on if the value of *tmax_sum* is greater than the value of parameter **transp_tmax**. If true variable *transp_on* is set to 1, *transp_check* is set to 1 and, tmax_sum is set to 0. Previously, the checks were reversed, which could keep

	transpiration on for a few days in a month, depending on the value of *tmax_sum* when the value of **transp_beg** and **transp_end** were specified equal to each other.
muskingum	allow specification of parameter **K_coef** to be less than 1.0, if found a warning message is issued instead of an error message when control parameter **parameter_check_flag** is specified greater than 0. If computed Muskingum flow coefficient ($c2$), based on parameters **K_coef** and **x_coef**, for a segment is < 0.0, this means the travel time through segment is small, thus outflow is primarily equal to inflow for the segment. If this condition is true, the computed Muskingum coefficient $c1$ is decreased by the computed value of coefficient $c2$ and $c2$ is set to 0.0. If the computed value of Muskingum coefficient $c0$ is < 0.0 then outflow from a segment would be greater than a day, the maximum travel time, which is mainly dependent on yesterday's flows. If this condition is true, the value of $c1$ is decreased by $c0$ and $c0$ is set to 0.0. Previously, those Muskingum equation coefficients were not adjusted based on these conditions.
temp_1sta and temp_laps	initial value used to replace a missing value changed from 50.0 **tmax_allrain**(**start_month**), this only affects the first time step.
basin	reordered code to determine variables *hru_perv*, *hru_imperv*, *dprst_frac_hru*, *dprst_area_max*, *dprst_frac_open*, *dprst_frac_clos*, *dprst_area_clos_max*, *dprst_area_open_max* to maintain restriction that the fraction of the HRU that is pervious is at least 0.0001. This change might produce slight changes in results for rare combinations of input parameters.
map_results	**gvr_cell_pct** values specified less than 1.0E-6 are set to 0.0, previously the lower limit was 1.0E-10.
obs	measured precipitation values specified less than 0.0001 are set to 0.0.

INPUT SPECIFICATION CHANGES – by module:

ccsolrad and ddsolrad	check added to be sure at least one value of parameter **hru_solsta** is specified greater than 0 when dimension **nsol** is specified greater than 0. If all values of **hru_solsta** were specified equal to 0 an array would be referenced beyond its memory limit.
basin_sum	the default value for parameter **print_freq** changed from 1 to 3.
basin	error check added to be sure dimensions **nhru** = **nssr** = **ngw** when using the surface-depression storage computation option. Error check added for values of parameter **op_flow_thres** specified greater than 1.0. Added check for values of parameter **hru_percent_imperv** specified less than 0.0 or greater than 0.999. Previously, if either of these conditions were true, the values were set to 0.0 and 0.999, respectively; now an error message is printed and execution stops. Added check for HRUs not connected to a stream segment as specified using parameter **hru_segment**. This check is applicable only when stream-segment routing and/or cascading flow routing are inactive; a warning message is issued to the screen and written in the Model Output File identifying any HRUs not connected. Added check for values of parameter **dprst_area** specified greater than the corresponding value of **hru_area** for any HRU; an error message is printed and execution stops. Added check for isolated stream segments, that

is a segment that does not receive inflow or flow to another segment as specified by parameter **tosegment**); error unless control parameter **parameter_check_flag** is specified equal to 0. Added checks to be sure valid values of **elev_units** and **cov_type** are specified; if an invalid value an error message is printed and execution stops. Values of parameters **soil_moist_max** and **soil_rechr_max** are set to 0.0001 if less than 0.0001 instead of set to 0.001 if less than 0.001. Values of parameters **hru_percent_imperv**, **dprst_area**, **op_flow_thres** and computed values of **dprst_area/hru_area** are set to 0.0 if less than 0.0001 with warning message issued.

gwflow	checks added for values of parameter **gwflow_coef** specified greater than 1.0 or GWRs specified as being a swale; if these conditions are found a warning message issued.
srunoff_smidx and srunoff_carea	added check for values of parameter **carea_max** specified greater than 1.0; if true an error message is printed and execution stops. Added check for the computation of local variable *carea_max* for srunoff_smidx based on parameters **smidx_coef**, **smidx_exp**, and **soil_moist_max** results in a value greater than 2.0; warning message is printed if control parameter **parameter_check_flag** specified equal to 1. Added checks for values of parameters **dprst_flow_coef**, **dprst_seep_rate_open** and **dprst_seep_rate_clos** specified greater than 1.0; if true an error message is printed and execution stops.
soilzone	don't reset parameters **ssr2gw_rate**, **soil_moist_init**, **soil_rechr_init**, **ssstor_init**, **sat_threshold**, **slowcoef_lin**, **slowcoef_sq**, **fastcoef_lin**, and **fastcoef_sq** if they are specified as invalid, instead if true an error message is printed and execution stops. Specification of **soil_rechr_max**>**soil_moist_max**, **soil_moist_init**>**soil_moist_max**, **soil_rechr_init**>**soil_moist_max**, *ssres_stor*>**sat_threshold**, **soil_rechr_init**>**soil_rechr_max**, **soil_moist_max**<0.0001, and **soil_rechr_max**<0.0001 are treated as warnings and reset to valid values instead of errors when control parameter **parameter_check_flag** specified equal to 0, previously these always were treated as errors. Added checks for **slowcoef_sq**, **slowcoef_lin**, **fastcoef_sq**, and **fastcoef_lin** specified equal to 0 as this could cause a divide by zero; if true an error message is printed and execution stops unless **parameter_check_flag** is specified equal to 0, for which a warning message is printed and any values less than 0.0001 are set to 0.0001. Check added for specifying valid values of **soil_type**, which can be 1, 2, or 3; if any invalid values are specified an error message is printed and execution stops
climateflow	make sure parameters **basin_solsta** and **basin_tsta** are always assigned a valid value, which is needed as they are written to the restart file. Added check for computed values of *tmaxf* greater than *tminf*; if true a warning message is printed and execution continues.
potet_pan	added check to be sure pan evaporation time series is included in Data File, if values of parameter **hru_pansta** are specified greater than dimension **nevap** or equal to 0 an error message is printed and execution stops.
intcp	if an values of parameter **epan_coef** are specified less than 0 an error message is printed and execution stops. This check is made only when pan evaporation data are

included in the Data File. If any values of parameters **covden_win** or **covden_sum** are specified less than 0 or greater than 1 an error message is printed and execution stops.

`xyz_dist` and `ide_dist` added checks to be sure data for at least 2 temperature and precipitation measurement stations are specified to be used in computations.

`xyz_dist` added check for specifying values of parameters **x_div**, **z_div**, **y_div**, **ppt_div**, **tmax_div**, and **tmin_div** equal to 0.0, as 0.0 would cause a divide by 0.0. If true an error message is printed and execution stops.

`potet_jh` added check for values of parameter **jh_coef_hru** specified greater than 150 or less than -50 and values of **jh_coef** specified greater than 10 or less than -1; if true a warning message is printed when control parameter **parameter_check_flag** is specified equal to 0; else an error message is printed and execution stops.

`map_results` added check to be sure the type of selected output variables is either real or double. Added check for selected output variables dimensioned by a value not equal to dimension **nhru**. Added checks for specification of **gvr_cell_id** values greater than dimension **ngwcell** and **gvr_hru_id** values specified greater than dimension **nhru**. If any of these checks are true an error message is printed and execution stops. If the value of parameter **mapvars_freq** is specified equal to 0 then the `map_results` module is not active even if control parameter **mapOutON_OFF** is specified equal to 1. If dimensions **nhru** and **nhrucell** are specified with equal values, then parameters **gvr_hru_id** and **gvr_cell_pct** are not required and ignored if specified in the Parameter File(s).

`temp_1sta` and `temp_laps` if too many missing values (greater than parameter **max_missing**) were found an error message is printed and the execution stops instead of continuing.

`call_modules` added checks for deprecated module names, if found a warning message is printed, but allow to maintain downward compatibility for older applications the module name is set to the current name.

OUTPUT CHANGES

- Print of list of valid modules with longer and more consistent descriptions.
- When module `climate_hru` is active for more than one type of CBH file, the module description is printed for each climate type. For example, if **climate_hru** is used to input temperature and precipitation values, a line is printed with the `climate_hru` version identification for the active Temperature Distribution and Precipitation Distribution modules. Previously, the identification was printed once no matter how many climate types were input using `climate_hru`.
- HRU identification numbers are printed allowing for seven digits instead of six throughout the code.
- Module: `temp_1sta_laps` added print of module version identification as it was missing.
- File srunoff.f90 print of module version identification changed to indicate whether using smidx or carea method.
- Module: `basin_sum` – when control parameter **print_debug** value is specified equal to 4, some debug values were summed instead of printed individually. Write variable *basin_potsw* instead of *orad* in summary tables. Printing of water balance values changed from F9.4 to F9.3.
- New function print_date added in file utils_prms.f90, which is used to print the current time step in consistent formats for warning and error messages: modules affected: `potet_pan`, `ddsolrad`, `precip_dist2`, `ccsolrad`, `temp_dist2`, `obs`, `ide_dist`, `snowcomp`, `gwflow`, `climateflow`, and `soilzone`.

- New functions check_param_value, check_param_limits, checkint_param_limits, checkdim_param_limits, check_param_zero, and check_restart_dimen added in file utils_prms.f90 so that duplicate code is reduced for checking many parameters for valid input values and to make warning and error messages more consistent.
- Module: `soilzone` – message for values of variables *slow_stor > pref_flow_thrsh* can be turned off as well as many warning messages for other modules if control parameter print_debug is specified equal to -1; this message was printing for very small differences.
- Module: call_modules – allow the same file name to be used control parameters save_vars_to_file and init_vars_from_file, need to be careful as this means the values stored when the file was first generated are overwritten. The start and end clock date and time and the elapsed time at end of simulation are output to the screen in Model Output file.
- Module: prms_summary – changed output format of variable values from F12.6 to PE14.6.

PRMS Version 3.0.5 (4/24/2012) - This release mainly focused on general code clean up and bug fixes.

General

1. MMF declmodule replaced with Fortran print_module
2. MMF write and open of model_output_file replaced with Fortran routines (opstr replaced with write_outfile). Affected modules: hru_sum_prms.f90, basin_sum.f90, basin.f90, call_modules.f90
3. MMF functions isleap and julian replaced with Fortran functions leap_day and julian_day
4. MMF function getstep replaced with an internal counter the current time stop (variable Timestep) is set to 0 in basin and incremented in obs.
5. Parameter checks in initialize procedure only performed for active HRUs
6. Restart capability added to all except: potet_hs.f90, potet_pt.f90, potet_hamon_prms.f, temp_2sta_prms.f, ccsolrad_prms.f, ddsolrad_prms.f, hru_sum_prms.f90, frost_date.f90, write_climate_hru.f90
7. precip_1sta.f90 and precip_laps.f90 combined into single file: precip_1sta_laps.f90
8. temp_1sta.f90 and temp_laps.f90 combined into single module: temp_1sta_laps.f90
9. srunoff_smidx.f and srunoff_carea.f combined into single module: srunoff.f90
10. transp_frost.f90: parameters spring_frost and fall_frost only checked for active HRUs to set initial values of basin_transp_on and transp_on
11. write_climate_hru.f90: changed output format from E10.2 to E12.4.
12. ddsolrad.f90: determination if the simulation time is in the summer based on equinox for the Northern and Southern Hemisphere is determined in the obs module instead of this module so that this code is in on location.
13. temp_dist2.f, precip_dist2.f, strmflow_lake.f, soilzone.f, climateflow.f, call_modules.f converted to F90: temp_dist2.f90, precip_dist2.f90, strmflow_lake.f90, soilzone.f90, climateflow.f90, call_modules.f90
14. obs.f90: make sure runoff_units has a value when nobs=0; set summer_flag, increment Timestep
15. utils.f90: added check to be sure when reading orad that the variable name is swrad, which is the variable name used by write_climate_hru. This could be a problem for people who generated their own swrad CBH file and used a different variable name.
16. basin.f90: changed NEARZERO from 1.0E-7 to 5.0E-6; DNEARZERO from 1.0D-12 to 1.0D-10 to better represent round-off error. Removed parameter basin_area. Segment routing order computation moved to here and removed from streamflow routing modules to reduce duplicate code.
17. cascade.f90: code related to water-body cascades removed. Instead the tosegment is used and a stream segment must be associated with each lake if present.

18. New module: strmflow_in_out, which routes flow through the stream network without assuming a routing travel time; thus, outflow from a stream segment equals the sum of the inform from any upstream segments and any lateral flows from associated HRUs.
19. potet_jh.f90 is a merge of files potet_jh.f90 and potet_jh_hru.f90.
20. srunoff.f90: merge of srunoff_smidx.f and srunof_carea.f. New parameter dprst_et_coef(nhru) was added to adjust evaporation potential from surface depressions. Evaporation from open depressions is computed prior to evaporation from closed. Both are computed at the potential rate, limited by unsatisfied ET. Unsatisfied ET is reduced after computing open evaporation to then limit closed evaporation. This was being done in a complicated way, which is now streamlined.
21. strmflow_lake.f90: routing strategy using cascade water body code replaced with tosegment order, which requires a stream segment to be associated with each lake. Code to computes fluxes by segment, such as seginc_gwflow, seginc_ssflow, seginc_sroff, seginc_swrad, seg_lateral_inflow, hru_outflow. Module was reorganized and has had little testing

BUG FIXES

1. potet_pan.f90: check dimension nevap>0 in declare procedure instead of initialize to avoid memory error. CRITICAL BUG FIX: module would not run as return code set to 1 instead of 0. If nevap<1, simulation stops with error message. Valid values for hru_pansta checked for active HRUs only.
2. transp_index.f90: changed maximum value from 12 to 13 so module works with paramtool fix option. Set transp_end_12 for each HRU so that there is a value so as to write the variable to the restart file.
3. basin_sum.f90: print_freq = 6 did not produce monthly and yearly tables, now does. Variables basin_runoff_ratio and basin_runoff_ratio_mo were in units of inches/cfs, changed to inches/inches (decimal fraction).
4. snowcomp.f90: allow cov_type to be > 2 to designate trees, was limited to cov_type=3
5. gwflow.f90: minimum elevlake_init changed from 0.0 to -10000.0 feet. BUG FIX: with strmflow_lake active with weirs or gates, seepage_gwr was in inches when it needed to be in acre-inches.
6. muskingum.f90: default for variable TS was set to 0.0 instead of 1.0, which for K_coef values < 1.0 would cause x_max to be set to 0.0 (not sure if this was a bug, Michael identified this as a problem). Checks were added for K_coef > 24.0. If parameter_check_flag = 1, this is an error, else a warning message is issued and K_coef is set to 24. Check added for C2<NEARZERO, if so, C1 set to C1 + C2 and C2 set to 0.0 (short travel time). Check added for C0<NEARZERO, if so, C1 set to C1 + C0 and C0 set to 0.0 (long travel time). BUG FIX: the routing loop was modified to properly include streamflow_cfs(obsin_segment(iorder)) in the current inflow of a segment and to set the seg_upstream_inflow to 0.0 for each internal time step. This module needs more verification. BUG FIX: obsin_segment was not working when K_coef was not equal to 1.0, the value was divided by the timestep instead of using the input value.
7. soilzone.f90: all preferential flow computations only performed when pref_flow_den > 0.0 for an HRU, previously some were always computed when at least one HRU had a preferential flow reservoir. BUG FIX: the Dunnian flow from the gravity reservoir was not set correctly if a preferential flow reservoir was not present, this was only an issue when computing the soil zone water balance when print_debug = 1. This did not affect the results of the soil zone computations.

PRMS Version 3.0.4 (1/15/2012) - This release mainly focused on bug fixes and changes to warning and error messages.

BUG FIXES

1. Module muskingum: Value of NaN was produced for streamflow if value of K_coef < 1.
2. Module musroute: seginc_swrad for first segment when no HRU contributes to that segment is set to second segment instead of leaving it set to 0.0.

3. Module xlimate_hru: CBH files are checked to be sure first time step is valid and if end-of-file or missing day found during a simulation.
4. Modules srunoff_carea and srunoff_smidx: amount of pervious and impervious surface runoff from open and closed depressions computed incorrectly resulting in too small of surface depression storage surface runoff.
5. basin_sum: water balance and basin_stflow_tot computed and outflow from basin printed using basin_stflow_in instead of basin_stflow_out; added variable basin_runoff_ratio = basin_ppt/basin_cfs.
6. Input error checking changed to get through reading Parameter File (mostly) before stopping on error condition instead of stopping at first input error

----

PRMS Version 3.0.3 (11/29/2012) - This version was prepared but never released because it failed the test cases.

PRMS Version 3.0.2 (9/22/2012) - This release mainly focused on bug fixes with a few minor changes to some input/output files, and parameters.

BUG FIXES

1. Module soilzone: A) limit flows produced in each reservoir to maximum capacities instead of antecedent storage + all inflow to a reservoir; this fix may result in substantial changes in results and is consistent with the PRMS conceptualization of the soil zone.
2. Module soilzone: B) set available potet for the average depth over the pervious portion instead of the whole HRU, this bug may have generated increased ET used by the capillary reservoir when impervious surfaces and surface depression storage are present in an HRU.
3. Module soilzone: C) basin_recharge was computed incorrectly, it was only set to the recharge value of the last HRU. D) if using strmflow_lake and the parameter lake_hru_id was not specified an invalid memory access could take place which would cause evaporation from lakes to be computed incorrectly.
4. Module strmflow_lake: flow out of lakes based on gate openings were computed using values specific to a single lake model, this was changed to be applicable to any model.
5. Module cascade: when greater than 2 cascades from a source to a destination, one or more of these cascades could have been ignored and fraction of cascade computed incorrectly. This would be a rare occurrence and was true for HRU and GWR cascade specifications.
6. Module intcp – A) if pan evaporation data are in Data File (nevap>0) and the ET module is not potet_pan then they are ignored rather than used as potet; B) allow winter cover density to be greater than summer density; previously if winter density was specified > summer, winter density was set to summer.
7. Module map_results: the number of output rows was computed incorrectly, if the dimension ngwcell was not equal to ngwcell/ncol and if nhru does not equal nhru/ncol.
8. Module obs – negative input values of observed runoff were set to -11; they are not changed from the values specified in the Data File, this could have caused problems with automated calibration programs.

OUTPUT AND INPUT ISSUES

1. Some variables changed from single precision to double precision, this might cause a very slight change in results that may have been due to round-off error - in modules: precip_dist2, temp_dist2, xyz_dist, strmflow_lake, basin_sum, and map_results.
2. A list of available and active modules is output to the user's screen.
3. Module climate_hru – new functionality: can input whether an HRU is in a transpiration day in a CBH file (0=no or 1=yes), previously the transpiration period was fixed for the entire simulation time period.

PARAMETER NAME CHANGES

1. temperature adjustment parameters were changed from **tmax_adj** and **tmin_adj** to **tmax_cbh_adj** and tmin_cbh_adj to avoid possible duplicate use of the parameter values. Similarly, precipitation adjustment parameters were changed from **rain_adj** and **snow_adj** to **rain_cbh_adj** and **snow_cbh_adj**. Check the messages about unused tmax_adj, tmin_adj, rain_adj, and/or snow_adj parameters and tmax_cbh_adj, tmin_cbh_adj, rain_cbh_adj and/or snow_cbh_adj are required but not specified. The value of solrad_tmax and solrad_tmin is now always set to basin_tmax and basin_tmin instead of using tmax(basin_tsta) and tmin(basin_tsta) when ntemp>0; this can produce differences in results.
2. Data File: A) missing temperature value indicator changed from less than -89 to less than -99 for modules: temp_laps, temp_1sta, temp_dist2, and xyz_dist.
3. Data File: B) added in variables humidity and wind_speed and dimensions   nwind and nhumid
4. Data File and Climate-by-HRU (CBH) files: Values are checked to see if any are specified as NaN (not a number), which results in an error message and stop of the execution. If any values in a CBH files are missing, the number of missing values and data type are output to the user's screen and the execution stops.
5. Module strmflow_lake: Parameter and variable names using "sfres" or "res" as part of their name, changed to use "lake" instead. The dimension nsfres changed to nlake. If the dimension nsfres is specified, a list of all parameter names that need to change is output to the user's screen and execution halts.
6. Module temp_dist2 - if no valid temperature values found, include in a warning message to indicate that the last valid temperature values is used for that time step.
7. Module hru_sum_prms has been deprecated and is only active if one or more of the following deprecated module are active: potet_hamon_prms, ddsolrad_prms, ccsolrad_prms, soltab_prms. If module hru_sum_prms is not active, parameters pmo and moyrsum are not needed.
8. Variable basin_stflow is replaced with basin_stflow_in, the total amount of lateral flow into the stream network. Variable basin_stflow_out was added, which is the total amount streamflow out of the model domain.
9. Module map_results –Added check to be sure the mapping specification is complete (gvr_cell_id(i)>0, gvr_hru_id(i)>0, and gvr_cell_pct>1.0E-10) from the HRU map to the target map; a warning message is output to the user's screen if this condition is detected. This check produces a warning message, as in some cases it might be desirable to map only a portion of an HRU to a target map spatial unit, such as, at a model boundary where the HRU map and target map have different spatial extents.

PRMS Version 3.0.1 (2/6/2012) - This release mainly focused on improving the messages that come to the command line window. Many users of PRMS were ignoring the error and warning messages because that they could not be understood. Also, users were uncertain about which modules were active in the runs they were making. The new messages will help with both of these cases.

1. There were some changes made to increase the precision of the snow and soil moisture module algorithms. Some of the FORTRAN variables were changed from single precision to double precision floating points. The consequence of this is some changes in the results in the simulation of snowpack storage and melt, and soil-moisture storage and recharge. These changes are evident in (sometimes) the 4th and (usually) the 5th significant digit.
2. No changes were made in this version which impact the format or content of the input and output files in any way.

PRMS Version 3.0.0 (11/15/2011) - This version of PRMS has been designed as a stand-alone program that can be executed on a Linux or Microsoft Windows platform. In some ways, this version may appear to return to the concepts and design of the earliest versions of PRMS. Much of the support functionality provided by MMS has been stripped away in favor of a "batch execution" mode for maximum application flexibility and

computational efficiency. This approach also supports maximum portability between computers running the Windows and Linux operating systems. Focus has been placed on ease of deployment, installation, and reliability over the MMS concepts of "model building." However, the module and function library developed for the MMS version of PRMS have been shown to be useful and have been retained.

_____

MMS Version MMS-windows-07-01-26 (1/26/2007) - The MMS version of PRMS was ported to PCs running the Windows operating system using Cygwin.

_____

MMS Versions 1.0.0 through 1.2.1 (1991 through 2002) –
- These were UNIX only releases. Although computationally efficient, the procedure required to add hydrologic-process algorithms to the original code was less than adequate. As a result, the architecture and modular structure of PRMS were redesigned to allow better integration and hydrologic-process algorithm-development capabilities. This new structure was the USGS Modular Modeling System (MMS), an integrated system of computer software developed for simulating a variety of water, energy, and biogeochemical processes that included PRMS.
- The basic hydrologic-process algorithms in PRMS were maintained in the MMS version; however, the use of MMS enabled the addition of new process algorithms and the enhancement of many of the features and capabilities in the original PRMS. These additions included graphical systems and networked data systems that took advantage of increased computational power.

_____

PRMS Version 2.1 (1/17/1996) - This was the pre-MMS version of PRMS. It was precompiled for computers running the DOS and DG UNIX operating systems.

PRMS Pre-version 2.0 (1983 - 1996) - PRMS originally was developed as a single FORTRAN 77 program composed of algorithms encoded in subroutines, each representing an individual physical process of the hydrologic cycle.

For the processes related to temperature distribution, solar-radiation distribution, evaporation, transpiration, and surface runoff, two or more different algorithms were encoded, each representing a different conceptual approach. A specific algorithm was selected at run time by setting values in the input file. This modular-design concept enabled the creation and application of a model that was most appropriate for a given study, and supported the long-term goal to expand the available process-simulation capabilities of PRMS.
This version of PRMS uses "punch card" formats for input files, line-printer-generated output plots, use of the USGS's National Water Data Storage and Retrieval system, and the job-control language specifications necessary to execute PRMS on the Amdahl and Prime computer systems.