

II.8-OFS-RRS OPERATIONAL FORECAST SYSTEM RIVER, RESERVOIR AND SNOW  
(RRS) PREPROCESSOR FUNCTION

Bookmarks

Purpose . . . . .	<a href="#">[Bookmark]</a>
Processing of Instantaneous Data . . . . .	<a href="#">[Bookmark]</a>
Missing Data Values Allowed . . . . .	<a href="#">[Bookmark]</a>
Missing Data Values Not Allowed . . . . .	<a href="#">[Bookmark]</a>
Processing of Mean Data . . . . .	<a href="#">[Bookmark]</a>
Removing Overlapping Periods . . . . .	<a href="#">[Bookmark]</a>
Processing of Mean Data with No Overlaps . . . . .	<a href="#">[Bookmark]</a>
Processing Logic . . . . .	<a href="#">[Bookmark]</a>
<a href="#">[Bottom]</a>	

Contents

	<u>Page</u>
Purpose . . . . .	<u>1</u>
Processing of Instantaneous Data . . . . .	<u>1</u>
Missing Data Values Allowed . . . . .	<u>2</u>
Missing Data Values Not Allowed . . . . .	<u>2</u>
Processing of Mean Data . . . . .	<u>3</u>
Removing Overlapping Periods . . . . .	<u>3</u>
Processing of Mean Data with No Overlaps . . . . .	<u>5</u>
Processing Logic . . . . .	<u>5</u>

[\[Next\]](#) [\[Previous\]](#) [\[Bookmarks\]](#) [\[Top\]](#)

Purpose

The Operational Forecast System (OFS) River, Reservoir and Snow Preprocessor Function (RRS) converts variably spaced river, reservoir and snow observations into time series form.

Mean and instantaneous data are processed differently and data types that do not allow missing values are processed differently from those that do.

Data are processed in the units in which the data are stored on the Preprocessor Data Base. It needed the Processed Data Base write routines convert the values to the units in which the time series are stored in the Processed Data Base.

For mean outflows from a reservoir the RRS Function contains the capability to distribute reported values to simulate a 'power wave'. The controls for his option, including the shape of the wave, are specified using the Preprocessor Initialization Program (PPINIT - see Section VI.3.3 [\[Hyperlink\]](#)).

[\[Next\]](#) [\[Previous\]](#) [\[Bookmarks\]](#) [\[Top\]](#)

Processing of Instantaneous Data

### **Missing Data Values Allowed**

The following rules are used to convert instantaneous observations into time series when missing values are allowed. If there is no observation at the given time step then the number of observations within plus or minus one-half time interval of the time step determines how the observation is processed.

- o number of observations is zero if all observations specify missing values: the value is set to missing
- o number of observations is equal to one: the value is set to the value of that observation
- o number of observations is greater than one but all are on one side of the time step: the value is set to the value of the closest observation
- o number of observations is greater than one and there are observations on either side of the time step: use either linear interpolation between the closest value on each side

### **Missing Data Values Not Allowed**

When missing data are not allowed it is necessary to both interpolate between observations and extrapolate beyond the last observation. The options as specified during station definition by program PPINIT for interpolation and extrapolation are:

- o interpolation
  - linearly interpolate between observations
  - retain previous observation until a new observation is reached
- o extrapolation
  - apply a recession factor to the last observation: the recession factor is defined on a daily basis and has a range of zero to one; a recession of one retains the last observation for all subsequent time steps

The following steps are used to convert instantaneous observations into time series when missing values are not allowed. If there is no observation at the time step then the estimating procedure depends on whether there are non-missing observations before and/or after the time step being estimated:

- o observation exists after time being estimated but not before: if there is no observation in the Preprocessor Data Base before the time being estimated then interpolation is performed between the time series value for the time step just prior to the start of the run (read from Processed Data Base) and the first observation that exists after the time being estimated; if no data exists in

the Processed Data Base for the previous time step then an error message is printed and the time series is not written

- o observations exist both before and after time being estimated: use interpolation
- o observation exists before time being estimated but not after: use extrapolation
- o no observations before or after: use value for latest time step in the Processed Data Base and extrapolate remaining values

[\[Next\]](#) [\[Previous\]](#) [\[Bookmarks\]](#) [\[Top\]](#)

### Processing of Mean Data

Mean data observations contain the time period associated with the observation in addition to the date/time and data values. Mean data require more processing than instantaneous data because the data represent a period of time and not just an instant.

Instantaneous data only have one observation per data/time value while mean data may have more than one observation per data/time because of different observation periods. This can result in observation periods that overlap. For example the following mean data observations have overlapping observation periods:

<u>Hour</u>	<u>Data Value</u>	<u>Observation Period</u>
06Z	6 CFS	6 hours
18Z	2 CFS	6 hours
18Z	4 CFS	18 hours

The first step in processing mean data is to sort through the observations and remove any overlaps. After this step the values can be computed for each time interval for which observations exist for the entire time interval. For data types that do not allow missing values it is necessary to compute values for missing periods that fall between observations by interpolation and to extrapolate values for all periods beyond the last observation.

[\[Next\]](#) [\[Previous\]](#) [\[Bookmarks\]](#) [\[Top\]](#)

### **Removing Overlapping Periods**

Overlapping periods of mean discharge and fraction snowfall data must be removed before the observed data can be converted to a time series. In order to remove the overlaps the shorter observation periods are processed first. Data for longer observation periods which overlap shorter periods are distributed by the shorter period values. A pointer array contains the number of the observations in order from the shortest to the longest observation period and determines the order in which observations are processed. Observations with the same observation period are ordered with the observations with later observation times coming before earlier ones. A work array containing hourly values for the processing period is initially set to missing.

The observations are then processed in the order defined in the pointer array and the missing values replaced by the observed ones. Once an observed value is written into the work array it is not replaced. If a portion of an observation period exists in the work array the observation value is distributed using values in the work array from shorter observation periods. The volumes or fraction snowfall of the longer periods are maintained whenever possible as the shorter periods are used to distribute them. The distribution process allows discharge values to be negative or fraction snowfall values to be outside the range of zero to one. When the work array no longer contains any missing values or all observations have been processed then all overlapping periods have been removed.

The following example illustrates how overlapping periods of discharge data are resolved. At the start the work array contains all missing values:

Observed Data:

<u>Date/Time</u>	<u>Data Value</u>	<u>Observation Period</u>
12Z	6 CFS	6 hours
18Z	4 CFS	18 hours

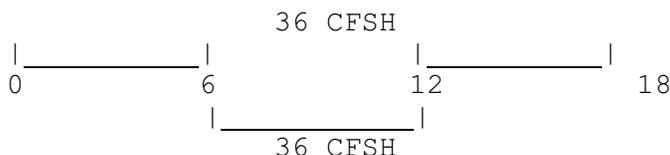
_____ 4 CFS _____			
0Z	6Z	12Z	18Z
_____ 6 CFS _____			

The 12Z observation is processed first because it has the shorter observation period. In the work array the missing hourly values from 6 to 12Z are replaced by 6 CFS. The second observation may not replace the values already in place from 6 to 12Z. The shorter period value is used to distribute the longer period creating new values that are added to the work array. First the mean values are converted into volumes:

$$6 \text{ CFS} * 6 \text{ hours} = 36 \text{ CFSH}$$

$$4 \text{ CFS} * 18 \text{ hours} = 72 \text{ CFSH}$$

In the shorter period (6 to 12Z) 36 CFSH occurred leaving 36 CFSH to be distributed in the remaining portion of the longer period:



The 36 CFSH is then evenly distributed over the 12-hour period (0-6Z and 12-18Z) yielding 3 CFS. The hourly values from 0 to 6Z and from 12 to 18Z are replaced with 3 CFS and the overlapping period no longer exists:



If the shorter period has a greater volume than the longer period then the rate for the shorter period is assumed to be valid and the remaining portions of the longer period are set to zero. If this occurs then a warning message is printed. Mean fraction snowfall data is processed using the same technique with the generated values ranging from zero to one.

It should be noted that mean data observations are first processed into the hourly value work array even if there are no overlapping periods.

[\[Next\]](#) [\[Previous\]](#) [\[Bookmarks\]](#) [\[Top\]](#)

### **Processing of Mean Data with No Overlaps**

After the observations have been transferred to the hourly work array the values for each time interval in the time series are computed based on whether missing values are allowed:

- o missing values allowed: Time series values are computed by computing the mean value for each time interval from the hourly values in the work array. If any of the hourly values in the work array are missing then the value for the time interval is set to missing.
- o missing values not allowed: If no missing values are allowed then the missing hourly values in the work array must first be estimated before computing the time series values. The estimation procedure uses the same interpolation and extrapolation options that are used for instantaneous data and depends on whether there are non-missing values in the work array before and/or after the hour estimated.
  - o non-missing value exists after but not before hour being estimated: The hour is estimated by interpolating between the value for the time step just prior to the start of the run (read from Processed Data Base) and the value for the next non-missing hour.
  - o non-missing values exist both before and after hour being estimated: use interpolation
  - o non-missing value exists before hour being estimated but not after: use extrapolation
  - o no non-missing hours exist in the work array: use value for the latest time step in the Processed Data Base and extrapolate

[\[Next\]](#) [\[Previous\]](#) [\[Bookmarks\]](#) [\[Top\]](#)

### Processing Logic

The processing logic for RRS data is as follows:

- Step 1: Call subroutine RPPREC to read the ORRS parameters from the Preprocessor Parametric Data Base. If the ORRS parameters are not found then call RPPREC to read the RRS parameters from the Processor Parametric Data Base for a station.
- Step 2: Call subroutine RPDRRS to read observations for the current station and data type from the Preprocessor Data Base. If the data type does not allow missing data then read all observations in the Preprocessor Data Base through the end of the requested period; otherwise just read observations for the requested period.
- Step 3: Check for a RRSMSNG MOD and set observations to missing if:
- the MOD is for the current station and data type
  - missing data are allowed for the current station and data type
  - the time period for which the observations are to be set to missing is within the run period
- Repeat for all RRSMSNG MODs for this station and type. If the entire run period is contained in the period to be set to missing then go to Step 6.
- Step 4: If the observed data for the current station and data type are to be displayed then output the data.
- Step 5: Convert the observations to time series using the techniques described above.
- Step 6: Call subroutine WPRD to write the current time series to the Processed Data Base.
- Step 7: If the time series data for the current station and data type are to be displayed then output the data.
- Step 8: Repeat Steps 2-7 until all data types for the current station are processed.
- Step 9: Repeat Steps 1-8 until all stations are processed.

[\[Previous\]](#) [\[Bookmarks\]](#) [\[Top\]](#)