

V.3.3-RES-J JOINT RESERVOIR REGULATION OPERATION

Bookmarks

Identifier [Bookmark]
Application [Bookmark]
Description [Bookmark]
Developed By [Bookmark]
Input Summary [Bookmark]
Organization [Bookmark]
General Syntax Rules [Bookmark]
Time Series Definition [Bookmark]
Topology Definition [Bookmark]
Parameters Definition [Bookmark]
 Parameters for each RESERVOIR Component..... [Bookmark]
 Parameters for each REACH Component..... [Bookmark]
 Parameters for each NODE Component..... [Bookmark]
Rules Definition [Bookmark]
RES-J Method Syntax [Bookmark]
Tables [Bookmark]
Table 1. RES-J Methods [Bookmark]
Table 2. Time series keyword parameters [Bookmark]
Table 3. RES-J Methods, components and variables [Bookmark]
Method ADJUST [Bookmark]
Method BALANCE [Bookmark]
Method CALCINFLOW [Bookmark]
Method LAGK [Bookmark]
Method LOOKUP3 [Bookmark]
Method MAXDECREASE [Bookmark]
Method MAXINCREASE [Bookmark]
Method MAXSTAGE [Bookmark]
Method PASSFLOW [Bookmark]
Method RAINEVAP [Bookmark]
Method SETELEVATION [Bookmark]
Method SETMAX [Bookmark]
Method SETMIN [Bookmark]
Method SETRELEASE [Bookmark]
Method SETSUM [Bookmark]
Method SETWITHDRAW [Bookmark]
Method SPILLWAY [Bookmark]
Index [Bookmark]
[Bottom]

Contents

	<u>Page</u>
Identifier	2
Application	2
Description	2
Developed By	3
Input Summary	3
Organization	3
General Syntax Rules	4
Time Series Definition	4
Topology Definition	6

Parameter Definition	7
Parameters for each RESERVOIR Component	7
Parameters for each REACH Component	11
Parameters for each NODE Component	12
Rules Definition	14
RES-J Method Syntax	18
Table 1. RES-J Methods	26
Table 2. Time series keyword parameters	28
Table 3. RES-J Methods, components and variables	29
Method ADJUST	1
Method BALANCE	1
Method CALCINFLOW	1
Method LAGK	1
Method LOOKUP3	1
Method MAXDECREASE	1
Method MAXINCREASE	1
Method MAXSTAGE	1
Method PASSFLOW	1
Method RAINEVAP	1
Method SETELEVATION	1
Method SETMAX	1
Method SETMIN	1
Method SETRELEASE	1
Method SETSUM	1
Method SETWITHDRAW	1
Method SPILLWAY	1
Index	1

[Next] [Previous] [Bookmarks] [Top]

Identifier: RES-J

[Next] [Previous] [Bookmarks] [Top]

Application: All programs

[Next] [Previous] [Bookmarks] [Top]

Description: Operation RES-J models either a single reservoir or a system of reservoirs and connecting reaches. It uses topological information to describe the network to be modeled as a interconnected group of component objects. Networks may consist of reservoir, reach and node components. Operation RES-J solves the entire network from upstream to downstream one time step at a time. Local flows to reaches and reservoirs and inflows to the upstream-most components must be generated before the RES-J Operation so that they are available as network inputs.

Methods are used to provide modeling functionality to each of the components. Methods are internal RES-J operations assigned to and acting upon network components, causing them to act in certain ways during simulation. For example, a SETRELEASE Method could be assigned to a Reservoir component to cause the reservoir to spill in a prescribed fashion. Methods are analogous to the Schemes/Utilities that are used in Operation RES-SNGL. They are referred to as Methods because their design is consistent with the object-oriented

methodology and to emphasize the fact that Method functionality may not be the same as comparable Schemes in RES-SNGL. Table 1 [Bookmark] lists the available RES-J Methods, the type of network component to which they are applied and a brief description of their functionality.

A Rule consists of a listing of one or more Methods and a governing conditional expression. When the expression evaluates to true the Methods will be enacted. This allows for variable operation of the system based on the state of the reservoir(s). For example, a rule based on the pool elevation of a downstream reservoir may evaluate to true when its elevation reaches a certain low level. The rule would then execute Methods which may include a SETRELEASE on an upstream reservoir to release larger quantities of water to help maintain a water level on the downstream reservoir.

See Chapter II.4-RES-J [[Hyperlink](#)] for a complete description of the reservoir system simulation.

[Next] [Previous] [Bookmarks] [Top]

Developed By: Riverside Technology, inc.

[Next] [Previous] [Bookmarks] [Top]

Input Summary: The input for Operation RES-J contains every piece of necessary parametric information for a complete model run. Because this input file has the potential to be quite large depending on the complexity of the reservoir system, the input is organized in four sections: time series information; topology definition; parameter and states definition, including Method and component descriptions; and rules definition. This discussion will address each of these sections of the input and will provide examples of each.

[Next] [Previous] [Bookmarks] [Top]

Organization: The input deck's individual sections must be organized in a specific order, with the beginning and end of each section's information identified with a keyword and companion keyword syntax:

```
SECTIONNAME
.
.
.
ENDSECTIONNAME
```

A complete input deck's section organization would be:

```
RES-J identifier
TIMESERIES
.
.
.
ENDTIMESERIES
TOPOLOGY
.
.
```

<u>Keyword</u>	<u>Definition and Definition</u>
----------------	----------------------------------

.	
ENDTOPOLOGY	
PARAMETERS	
.	
.	
ENDPARAMETERS	
RULES	
.	
.	
ENDRULES	
ENDRES-J	

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

General Syntax Rules

Input is in free format with fields separated by at least one blank.

Keywords are used by RES-J to enable rapid processing of parameter information while maintaining user readability. Most keywords (but not all) have companion keywords. The keyword, such as 'TOPOLOGY' and its companion keyword, such as 'ENDTOPOLOGY' surround the information associated with the topology definition. Keywords are identified within this documentation by full-capitalization. Optional keywords are enclosed in brackets '['].

Input is not case sensitive. All words and strings are converted to uppercase.

Comments are allowed throughout the input deck and are specified by the '#' character. Any characters on the same line and after the '#' will be ignored.

Information required to be on one line may be continued over multiple lines by ending each line with ' \' (a blank space and backslash). No line in the input file should extend beyond column 72 because it may not be decoded correctly.

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

Time Series Definition

This identifies input and output time series and prepare for the data transfer to and from the Forecast System. It also facilitates use of the time series by assigning an alias to each time series. These time series must correspond to component state variables.

<u>Keyword</u>	<u>Definition and Definition</u>
----------------	----------------------------------

TIME SERIES	Indicates the beginning of the time series section
TIMESTEP	Simulation time interval in hours (must be an integer fraction of 24)

<u>Keyword</u>	<u>Definition and Definition</u>
INPUT	<p>Indicates an input time series for the operation. Following 'INPUT' and on the same line, define the time series using:</p> <ul style="list-style-type: none"> - a Forecast system time series identifier up to 8 characters in length - data type code - data time interval - an alias used by RES-J to facilitate reference of the time series; must be unique among time series aliases and can be up to 12 characters in length
OUTPUT	<p>Indicates a time series that is an output from the Operation. Following 'OUTPUT' and on the same line, define the time series using:</p> <ul style="list-style-type: none"> - a Forecast system time series identifier (maximum 8 characters) - data type code - data time interval - an alias used by RES-J to facilitate reference of the time series; must be unique among time series aliases and can be up to 12 characters in length <p>All output time series must be of the TIMESTEP interval.</p>
ENDTIMESERIES	Indicates the end of the time series section

The following is an example of a time series definition section:

```

TIME SERIES
  TIMESTEP 06
  INPUT PREC_MAD MAP 06 Mad_Map
  INPUT MAD_INFL SQIN 06 Mad_In
  INPUT GAT_INFL SQIN 06 Gat_In
  OUTPUT MAD_RELS RQOT 06 Mad_Out
  OUTPUT GAT_POOL PELE 06 Gat_Elev
ENDTIMESERIES

```

Time series are referenced by components, Methods, rules and expressions using the time series alias and a time series keyword. The time series keyword specifies the contents of the time series (e.g. INFLOW or OBSERVEDRELEASE). Table 2 [Bookmark] lists the valid time series keywords, a description, an indicator to show whether the time series is input or output and the components/Methods to which the time series are applicable. The time series keywords required to parameterize each component and Method are also included in their respective syntax explanations.

[Next] [Previous] [Bookmarks] [Top]

Topology Definition

This section identifies each component of the system and its position relative to other components.

<u>Keyword</u>	<u>Definition and Format</u>
TOPOLOGY	Indicates the start of input for the topology section
RESERVOIR	Indicates a reservoir component. 'RESERVOIR' must be followed by: <ul style="list-style-type: none">- an identifier (or name) for the reservoir, unique to all component identifiers The identifier may be followed by one or more sets of topologic information used to identify components upstream from the current reservoir being defined: <ul style="list-style-type: none">- topological keyword 'BELOW'- an identifier of the upstream component
REACH	Indicates a reach component. 'REACH' must be followed by: <ul style="list-style-type: none">- an identifier (or name) for the reach, unique to all component identifiers. The identifier may be followed by one or more sets of topologic information used to identify components upstream from the current reach being defined: <ul style="list-style-type: none">- topological keyword 'BELOW'- an identifier of the upstream component
NODE	Indicates a node component. 'NODE' must be followed by: <ul style="list-style-type: none">- an identifier (or name) for the node, unique to all component identifiers. The identifier may be followed by one or more sets of topologic information used to identify components upstream from the current reach being defined: <ul style="list-style-type: none">- topological keyword 'BELOW'- an identifier of the upstream component
ENDTOPOLOGY	Indicates the end of input for the topology section

Important considerations when defining topology are:

1. RESERVOIR, REACH and NODE are the defining members (components) of topology.
2. The definition of the topology must proceed from upstream to downstream. Therefore, no component can be defined 'BELOW' another component until the upstream component has been defined.
3. A component may be linked to any number of upstream components.

4. A component cannot have more than one downstream component.

The following is an example of the topology definition section:

```
TOPOLOGY
  RESERVOIR Madden
  REACH Charges BELOW Madden
  REACH YourReach
  RESERVOIR Gatun BELOW Charges BELOW YourReach
  NODE YourNODE BELOW Gatun
ENDTOPOLOGY
```

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

Parameter Definition

This section characterizes each component in the system and describe each Method used during simulation.

In discussing the parameter definition section, component parameterization is addressed first and foremost. Despite being integrated with component definition in the input file parameters section, Method parameterization will only be briefly treated here. Each Method requires in-depth discussion of unique parameter syntax and functionality and therefore will be discussed later.

The parameters section begins with the keyword 'PARAMETERS' and ends with its companion keyword 'ENDPARAMETERS' with all component and Method parameters embedded between.

<u>Keyword</u>	<u>Definition and Format</u>																		
PARAMETERS	Indicates the start of parameter information																		
[UNITS]	Units of the input. 'UNITS' must be followed by: - 'ENGLISH' or 'METRIC' 'UNITS' must be the first item after 'PARAMETERS'. The default is 'METRIC'. Standard units for input are: <table><thead><tr><th></th><th><u>English</u></th><th><u>Metric</u></th></tr></thead><tbody><tr><td>Elevation</td><td>FT</td><td>M</td></tr><tr><td>Instantaneous discharge</td><td>CFS</td><td>CMS</td></tr><tr><td>Mean discharge</td><td>CFSD</td><td>CMSD</td></tr><tr><td>Storage</td><td>ACFT</td><td>M3</td></tr><tr><td>Rainfall</td><td>INCH</td><td>MM</td></tr></tbody></table>		<u>English</u>	<u>Metric</u>	Elevation	FT	M	Instantaneous discharge	CFS	CMS	Mean discharge	CFSD	CMSD	Storage	ACFT	M3	Rainfall	INCH	MM
	<u>English</u>	<u>Metric</u>																	
Elevation	FT	M																	
Instantaneous discharge	CFS	CMS																	
Mean discharge	CFSD	CMSD																	
Storage	ACFT	M3																	
Rainfall	INCH	MM																	
ENDPARAMETERS	Indicates the end of parameter information.																		

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

Parameters for each RESERVOIR Component

Keyword

Definition and Format

The reservoir is the most basic and necessary component in a RES-J implementation. The following information describes the characteristics, states and limitations of the physical reservoir. Information parameterizing the operation of the reservoir is contained in various Methods to be described later.

Keyword

Definition and Format

RESERVOIR

Indicates the start of reservoir parametric information. 'RESERVOIR' must be followed by a reservoir-type identifier defined in the topology section.

[CONSTANT]

Indicates specification of a constant to be used in the RULES definition section. These constants should not be used more than once in an English unit parameterization. Otherwise the conversion to RES-J internal units (SI units) will be erroneously performed more than once. 'CONSTANT' must be followed by:

- 'component identifier.constant identifier'
- the constant's value

The following example assigns a value of 253.0 to a Madden reservoir constant called Flood_Elevation:

```
CONSTANT Madden.Flood_Elevation 253.0
```

TSINPUT

Indicates the existence of an 'inflow' time series. 'TSINPUT' must be followed by:

- time series keyword 'INFLOW'
- an input time series alias defined in the TIMESERIES definition section 1/

All reservoirs located at the top of the system topology tree (no components further upstream) must have at least one usage of 'TSINPUT'.

Reservoirs elsewhere in the system may use 'TSINPUT' to define a local inflow, if desired, with outflow from an upstream component acting as the primary (and necessary) inflow to the reservoir. Outflow from upstream components MUST NOT be defined using TSINPUT. Multiple uses of TSINPUT are acceptable.

[TSOUTPUT]

Indicates that one of the reservoir state time series will be output from RES-J. 'TSOUTPUT' must be followed by:

- time series keyword 'POOL', 'RELEASE', 'STORAGE' or 'WITHDRAW'. (Augmentation is included in the WITHDRAW variable.)
- an output time series alias defined in the TIMESERIES definition section 1/

Multiple uses of TSOUTPUT are acceptable.

<u>Keyword</u>	<u>Definition and Format</u>
TABLE	<p>Keyword indicating start of the elevation-storage table. 'TABLE' must be followed by:</p> <ul style="list-style-type: none"> - table keyword 'ELEV_STOR' signifying the table's use as an elevation-storage table on the same line. <p>Subsequent lines must contain the table data:</p> <ul style="list-style-type: none"> - One elevation and its corresponding storage value pair, per line. Units are as defined above for elevation and storage.
ENDTABLE	Indicates the end of the elevation-storage table.
INITIALPOOL	Initial pool elevation; value must follow.
[INITIALRELEASE]	Initial instantaneous release; value must follow. Defaults to 0.0.
[INITIALWITHDRAW]	Initial instantaneous withdrawal; value must follow. Defaults to 0.0. (Augmentation is included in the WITHDRAW variable as a negative quantity.)
[INITIALINFLOW]	Total initial instantaneous inflow; value must follow. Defaults to 0.0.
[PREVIOUSPOOL]	Pool elevation one time step prior to carryover save; value must follow. Defaults to INITIALPOOL.
[PREVIOUSRELEASE]	Instantaneous release one time step prior to carryover save; value must follow. Defaults to 0.0.
[PREVIOUSWITHDRAW]	Instantaneous withdrawal one time step prior to carryover save; value must follow. Defaults to 0.0. (Augmentation is included in the WITHDRAW variable as a negative quantity.)
[PREVIOUSINFLOW]	Total instantaneous inflow one time step prior to carryover save; value must follow. Defaults to 0.0.
[MINPOOL]	Minimum allowable pool elevation; value must follow. If 'MINPOOL' is not explicitly defined it will default to the lowest elevation specified in the elevation-storage table
[MINRELEASE]	Minimum allowable instantaneous release; value must follow. If 'MINRELEASE' is not explicitly defined it will default to 0.0.

ENDRESERVOIR Indicates the end of reservoir parametric information.

Carryover transfer:

Pool elevation, release, withdrawal and inflows (initial and previous values) are saved.

During segment redefinition, the carryover is treated as 'general' carryover, that is data are considered independent of defined Methods for a given reservoir. Therefore, transformation of carryover will only occur if the carryover pool elevation lies outside the range of elevation on a modified elevation-storage curve. Otherwise, the carryover will simply be transferred from the old carryover array to the new carryover array. Both initial and previous values are handled identically.

The following rules apply:

```
release(new) = release(old)
withdrawal(new) = withdrawal(old)
inflow(new) = inflow(old)
```

The pool elevation will be changed by the following rule if elevation-storage curve is changed during segment redefinition.

```
pool elevation:
  if pool(old) > max elev. (new), then
    pool(new) = max elev. (new)
  if pool(old) < min pool (new), then
    pool(new) = min elev. (new)
  otherwise,
    pool(new) = pool(old)
```

The following are examples of RESERVOIR component input:

#Beginning of Reservoir definition
RESERVOIR Madden

```
TSINPUT INFLOW MAD_INFLOW
#with alias. Mad_In as providing
```

```
#most upstream. The only
```

#Madden's declaration (as a #reservoir) and state information.

```
#Specification of the time series
```

```
#the INFLOW data for Madden. The
#specification below displays the
#option of specifying an input time
#series using the full identifier.
#An inflow time series is
#required if the component is the
#input time
```

```
#series type that may be specified
#in the RESERVOIR/REACH/NODE
#section is inflow-every other time
#series is considered to be
#method-specific and must be
#declared with its associated
#method.
```

```
TSOUTPUT RELEASE MADRELES RQIE 01
```

```

#Specification of output of RELEASE
#values to the Mad_sim_rel time
#SERIES. Any output from any #COMPONENT
MUST be declared in the

#RESERVOIR/REACH/NODE section.
#Definition of Madden's elevation-
#storage table. Must begin and
#end with TABLE and
#ENDTABLE. If units are English,
#then storage units are ACFT,
#otherwise M  $\bar{3}$ /. This table must
#otherwise M  $\bar{3}$ /. This table must
#be named elev_stor.
TABLE elev_stor
  80 1000000
  100 10000000
  110 10500000
  115 11200000
  125 12100000
  130 12600000
  135 12600000
  140 13000000
ENDTABLE
INITIALPOOL 135 #Required.
INITIALRELEASE 38.371
MINPOOL 100 #Optional - defaults to lowest pool
#from the elev_stor table.
ENDRESERVOIR #Signifies the end of Madden's
#information.
RESERVOIR Gatun
TSOUTPUT POOL Gat_elev #Output of pool elevation.
CONSTANT Gatun.Identifier 56.5
#Use of the CONSTANT is for the
#rules definitions. The Identifier
#can be any text and it must
#be preceded by the
#component name. See the rules
#section for details on how
#CONSTANT values are used.
TABLE elev_stor
  80 1000000
  100 10000000
  110 10500000
  115 11200000
  125 12100000
  130 12400000
  140 13000000
ENDTABLE
INITIALPOOL 132.5
INITIALRELEASE 45
ENDRESERVOIR
#End of reservoir information

```

[Next] [Previous] [Bookmarks] [Top]

Parameters for each REACH Component

The reach is used primarily to route water from one component to another through use of routing Methods parameterized elsewhere. Because reach characteristics are part of the routing Methods, reach parameters consist only of constants and time series labels.

<u>Keyword</u>	<u>Definition</u>
REACH	Indicates the start of reach parametric information. 'REACH' must be followed by a reach-type identifier defined in the topology section.
[CONSTANT]	Indicates specification of a constant to be used in the RULES definition section. These constants should not be used more than once in an English unit parameterization. Otherwise the conversion to RES-J internal units (SI units) will be erroneously performed more than once. 'CONSTANT' must be followed by: <ul style="list-style-type: none"> - 'component identifier.constant identifier' - the constant's value
TSINPUT	Indicates the existence of an inflow time series to be applied at the top of the reach. 'TSINPUT' must be followed by: <ul style="list-style-type: none"> - time series keyword 'INFLOW' - an input time series alias defined in the TIMESERIES definition section <u>1</u>/ <p>All reaches located at the top of the system topology tree (no components further upstream) must have at least one usage of 'TSINPUT'. Reaches elsewhere in the system may use 'TSINPUT' to define a local inflow, if desired, with outflow from an upstream component acting as the primary (and necessary) inflow to the reach. Outflow from upstream components MUST NOT be defined using TSINPUT. Multiple uses of TSINPUT are acceptable.</p>
[TSOUTPUT]	Indicates that one of the reach state time series will be output from RES-J. 'TSOUTPUT' must be followed by: <ul style="list-style-type: none"> - time series keyword 'OUTFLOW' or 'RELEASE', which are synonymous. - an output time series alias defined in the TIMESERIES definition section <u>1</u>/
ENDREACH	Keyword indicating end of reach parametric information.

Carryover transfer:

No carryover is saved.

The required carryover for a reach is dependent on the routing Method used within the reach. Since LAGK is the only routing Method available, carryover related to a Reach component is described in the Method LAGK description.

[Next] [Previous] [Bookmarks] [Top]

Parameters for each NODE Component

A node is used as a confluence point of flows and as a checkpoint for certain stage / discharge relationships found in Method MAXSTAGE. It can also be a location for diversion or augmentation when using a Lookup3 method. Minimum remainder discharges can constrain diversions to maintain a minimum discharge from the node, subject to sufficient inflow to the node.

<u>Keyword</u>	<u>Definition and Format</u>
NODE	Indicates the start of node parametric information; 'NODE' must be followed by a node-type identifier defined in the topology section.
[CONSTANT]	Indicates specification of a constant to be used in the RULES definition section. These constants should not be used more than once in an English unit parameterization. Otherwise the conversion to RES-J internal units (SI units) will be erroneously performed more than once. 'CONSTANT' must be followed by: <ul style="list-style-type: none">- 'component identifier.constant identifier'- the constant's value
TSINPUT	Indicates the existence of an inflow time series to be applied at the top of the node. 'TSINPUT' must be followed by: <ul style="list-style-type: none">- time series keyword 'INFLOW'- an input time series alias defined in the TIMESERIES definition section <u>1/</u> <p>Any node located at the top of the system topology tree must have at least one usage of 'TSINPUT'. Nodes elsewhere in the system may use 'TSINPUT' to define a local inflow, if desired, with outflow from another component acting as the primary (and necessary) inflow to the reach. Outflow from upstream components must not be defined using TSINPUT. Multiple uses of TSINPUT are acceptable.</p>
[TSOUTPUT]	Indicates that one of the node state time series will be output from RES-J. 'TSOUTPUT' must be followed by: <ul style="list-style-type: none">- time series keyword 'OUTFLOW', 'DIVERSION' or 'INFLOW'- an output time series alias defined in the TIMESERIES definition section <u>1/</u>; multiple usages of TSOUTPUT are acceptable
[INITIALDISCHARGE]	Initial discharge at the node; value must follow. Legacy keyword "DISCHARGE" also accepted, but discouraged. <u>4/</u>

[PREVIOUSDISCHARGE] Discharge at the node one step prior to carryover save; value must follow 4/

[INITIALINFLOW] Initial inflow into the the node; value must follow. This value does not include any augmentation flow associated with the Lookup3 Method. Augmentation is included in the DIVERSION keywords.

[PREVIOUSINFLOW] Inflow to the node one step prior to carryover save; value must follow. This value does not include any augmentation flow. 4/

[INITIALDIVERSION] Initial diversion from the node; value must follow. A negative value is acceptable, representing an augmentation of flow at the node (associated with the Lookup3 method).

[PREVIOUSDIVERSION] Diversion from the node at the node one step prior to carryover save; value must follow. A negative value is acceptable, representing an augmentation of flow at the node (associated with the Lookup3 method). 4/

[MINREMAINDER] Indicates the start of a date / minimum remaining discharge pair list. This minimum remaining discharge will be used to constrain, as possible, diversions from the node to meet any discharge requirements immediately below the node.

Beginning on the next line after 'MINREMAINDER', list any number of date / minimum remaining discharge pairs, one pair per line. Each date should be unique. All values must be non-negative. It is recommended that the list of pairs begin with 1/1_00:00 (midnight on the morning of January 1) and end with 12/31_23:00 (11:00 p.m. on December 31). Doing so will facilitate lookup procedures on the list and aid the user in ensuring that the transition over year's end will function as desired.

If 'MINREMAINDER' is not defined, the value will default to 0.

[ENDMINREMAINDER] Indicates the end of the 'MINREMAINDER' list.

[RATINGCURVEID] NWSRFS Rating Curve ID; value must follow (a rating curve id can not be specified if a TABLE is also defined, see below); a valid Rating Curve ID is <= 8 characters long

[TABLE] Indicates the start of the stage/discharge

rating curve table. TABLE must be followed by the 'RATING_CURVE' keyword. Beginning on the next line after 'RATING CURVE' create a table of stage elevation / discharge pairs describing the channel hydraulic conditions at the node. A rating curve TABLE cannot be specified if a RATINGCURVEID is specified.

[ENDTABLE] Indicates the end of the rating curve table.

ENDNODE Indicates the end of node parametric information

If both a RATINGCURVEID and a Rating TABLE are specified at a NODE, a NODE DEFINITION INPUT ERROR will be output.

Carryover transfer:

INITIALDISCHARGE, PREVIOUSDISCHARGE, INITIALINFLOW, PREVIOUSINFLOW, INITIALDIVERSION and PREVIOUSDIVERSION values are saved. Transfer of these data are direct, without transformation:

```
DISCHARGE(NEW) = DISCHARGE(OLD)
PREVIOUSDISCHARGE(NEW) = PREVIOUSDISCHARGE(OLD)
INITIALINFLOW(NEW) = INITIALINFLOW(OLD)
PREVIOUSINFLOW(NEW) = PREVIOUSINFLOW(OLD)
INITIALDIVERSION(NEW) = INITIALDIVERSION(OLD)
PREVIOUSDIVERSION(NEW) = PREVIOUSDIVERSION(OLD)
```

1/ The time series identifier, data type and time interval can be used in place of the alias to identify a time series.

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

Rules Definition

The simulation rules are specified by grouping one or more Methods (a Method list) with an expression. The expression evaluates a system state, such as a reservoir inflow, outflow or pool elevation. Only when the expression evaluates to true do the Methods associated with the expression execute. Because system solution proceeds from upstream components to downstream components, actual execution of Methods listed in the RULES section may vary from the order in which they are listed. For a given component, however, each Method listed in the RULES section will test its conditional expression (and execute, if TRUE) in the order the component's Methods are listed in the RULES section.

Although the syntax suggests a one-to-many relationship between an expression and the Methods in its list, RES-J actually parses the RULES into a one-to-one type relationship. Each Method explicitly listed under an expression will receive a copy of the expression. If a Method is explicitly listed more than once in the RULES definition section, the Method's execution will be conditional on the last expression assigned to the Method. Any Method implied in the RULES

section by explicitly listing a combo-type Method such as SETSUM, SETMIN or SETMAX (the implied Method having been defined as part of the combo-type Method in the parameter definition section) does not require an expression to be executed. Execution of the implied Method depends solely on the execution of the explicitly listed combo-type Method. Any Method defined in the parameters section which is not explicitly listed or implied in the RULES definition section will never execute, as it is never assigned an expression capable of evaluating to TRUE. States, used in evaluation of the expression, are updated following solution of a loss-type Method such as RAINEVAP or upon complete solution of the system for a given timestep. Release and withdrawal-type Methods do not update the state. 1/

PREVIOUS* refers to the value at the beginning of the previous timestep; STARTING* refers to the value at the start of the current timestep; ENDING* refers to the value at the end of the current timestep. See Chapter II.4-RES-J [[Hyperlink](#)] for complete discussion of states and how and when they are updated.

The RULES defined in this section are not a progression of state checks. Note that while the concept is in the format of a series of 'If-Then' statements, there is neither an 'Else' statement nor default condition applicable if none of the expressions evaluate to true. Use of the 'TRUE' conditional expression, which always evaluates to true, can be used to provide a type of default condition when placed at the top of the RULES definition.

<u>Keyword</u>	<u>Definition and Format</u>
RULES	Indicates the start of the Rules Section
[Indicates the beginning of an expression
	A simple expression is a comparison of two values through use of a conditional.
	Values can be references to component states, component-defined constants, numerical constants, summations of any combination of two states or constants, dates and 'TRUE'. <u>2/</u>
	Valid values include:
	Reservoir states:
	PREVIOUSPOOL, STARTINGPOOL, ENDINGPOOL, PREVIOUSINFLOW, STARTINGINFLOW, ENDINGINFLOW, PREVIOUSRELEASE, STARTINGRELEASE, ENDINGRELEASE, PREVIOUSWITHDRAW, STARTINGWITHDRAW and ENDINGWITHDRAW. The states POOL and RELEASE are also valid. Because these states refer to the same values as STARTINGPOOL and STARTINGRELEASE, respectively, their use is discouraged. (Augmentation values are inherent in the WITHDRAW variables.) <u>3/</u>

Keyword

Definition and Format

Format is '<ReservoirName>.<state>'.

Example: 'ReservoirA.PREVIOUSPOOL'

Node states:

PREVIOUSINFLOW, STARTINGINFLOW,
ENDINGINFLOW, PREVIOUSDISCHARGE,
STARTINGDISCHARGE, ENDINGDISCHARGE,
PREVIOUSDIVERSION, STARTINGDIVERSION,
ENDINGDIVERSION. (Augmentation values are
inherent in the DIVERSION variables.) 4/
Format is '<NodeName>.<state>'.

Example: 'Node.PREVIOUSDISCHARGE'

Component-defined constants as defined in the
Component parameter section:

These constants should only be used once
in a rule set parameterized using English
units. Otherwise the conversion to RES-J
internal units (SI units) will be
erroneously performed more than once.

Example: 'ReservoirA.BottomOfSpillway'

Numerical constants:

Any numerical value. Units associated with
the value are determined by the UNITS
state (ENGLISH or METRIC) and the type of
units associated with the preceding value.

4/

Examples: 98.3, -15.4.

Sums or Differences:

Two or more values can be summed or their
difference can be calculated using the '+'
or '-' signs, respectively, to form a
single value. 5/

Example:

ResA.PREVIOUSINFLOW -
ResA.STARTINGINFLOW

Dates:

The keyword 'DATE' refers to the current
date time the model is solving.

A user specified constant can also be
used. 6/

TRUE:

The keyword 'TRUE' defines an expression
that always evaluates to true.

Conditionals are symbolic references of equality,
non-equality, less-than, less-than-or-equal-to,
greater-than and greater-than-or-equal-to.

Valid conditionals respective to the description
above are:

'==', '!=', '<', '<=', '>' and '>='

Examples of simple expressions include:

[TRUE]

[ResA.STARTINGPOOL >= 95]

[ResB.ENDINGPOOL < ResB.MinLockageElev]

Keyword

Definition and Format

```
[NodeA.STARTINGDISCHARGE > NodeA.PREVIOUSDISCHARGE]  
[NodeB.STARTINGDISCHARGE - \  
  NodeA.PREVIOUSDISCHARGE > 100]  
[DATE > 06/01]
```

Complex expressions can be constructed using logical operators.

Valid logical operators are '&&' and '||' representing AND and OR logic, respectively. Parentheses can be used to group portions of the complex expression, but should be limited in their use. Line continuation symbols ('\') can also be used for long expressions. The following syntax example demonstrates construction of complex expressions from multiple expressions. 7/

```
[ ResA.STARTINGPOOL > 280 || \  
  (ResA.PREVIOUSINFLOW > 1000 && \  
  NodeB.STARTINGDISCHARGE < 450) ]
```

] Indicates the end of an expression

:: Indicates the beginning of a Method. '::' must be followed by:

- S A component identifier defined in the topology section
- S A Method identifier defined in the parameters section

ENDRULES Indicates the end of the Rules definition

- 1/ Note that prior to NWSRFS release 22, a given component's states were updated immediately following its solution. Because of this, an expression based on the states of an upstream component might evaluate differently for the Methods associated with upstream components than it would for Methods associated with downstream components within a time step.
- 2/ Expressions can test states unrelated to the objects owning the Methods to be executed if the expression is true. Be warned that use of the MAXSTAGE Method may create a copied subsystem with expressions linked to components which are not included in the subsystem. This will cause the system to fail.
- 3/ ENDING* states are not defined until after the component solves for the current timestep-only components downstream of the object should test the state. An exception to this limitation is ENDINGINFLOW which becomes available as soon as all inflows to a reservoir become known (meaning upstream components have solved for outflow).
- 4/ Note that the units with these numbers could only be determined by parsing the previous value. Therefore, numerical constants should

Keyword Definition and Format

generally follow a Component state whose dimensionality can be determined (i.e.: PREVIOUSPOOL has units of length). Do not begin an expression with a numerical constant. Always begin with a component state, date or TRUE.

- 5/ Note that no check for appropriate dimensionality (consistency of units) is made.
- 6/ All dates are considered with equal precision. For example, 2/2 will be interpreted as 2/2_00:00. Date expressions are evaluated using the fractional portion of the date following conversion to decimal years. To specify a date constant use one of the following formats:
 - '2-digit_month'/'2-digit_day'
 - '2-digit_month'/'2-digit_day'_'2-digit_hour'
 - '2-digit_month'/'2-digit_day'_'2-digit_hour':'2-digit_minute'
- 7/ No line in the input file should extend beyond column 72 because it may not be decoded correctly.

The following is an example of a set of rules:

```
RULES
[(Madden.STARTINGpool > 120) && (DATE > 10/15) ]
    #Condition which, if satisfied,
    #will cause execution of these
#three methods.
RAINEVAP Madden mad_rel      #Each method must be identified
LAGK YourReach lagk      #using the method type,
SETWITHDRAW Gatun gat_with  #component name and method name.

[(Gatun.previouspool < Gatun.Identity)]
    #This conditional will be
    #evaluated next. If satisfied,
    #it will cause execution of
    #these three methods and
    #overwrite any previous TRUE
    #RULES. Notice the use of the
    #CONSTANT value, Gatun.Identity.
SETELEVATION Madden mad_elev #Each method must be identified
LAGK YourReach lagk      #using the method type,
SETWITHDRAW Gatun gat_with  #component name and method name.

[(Gatun.STARTINGRELEASE < 3000.)]
    #This conditional will be
    #evaluated next. If satisfied,
    #it will cause execution of the
    #next method and overwrite any
    #previous TRUE RULES.
SETRELEASE Gatun power      #Each method must be identified
    #using the method type,
```

#component name and method name.

ENDRULES

[Next] [Previous] [Bookmarks] [Top]

RES-J Method Syntax

A system of reservoir, reach and node components is able to mass balance the water in a network at every time step solving Methods that are specific to each individual component. Table 3 [Bookmark] lists the Methods in RES-J and shows the type of component to which they apply, as well as the computational variable.

A combination Method is one that can act on several Methods that solve for the same type of output variable. For example, a SETMAX can be applied to a SETRELEASE Method and a MAXDECREASE Method since both solve for release. SETMAX will select the largest of the two computed releases. Similarly, a SETSUM Method could be applied to multiple SETWITHDRAW Methods and it would sum the individual withdrawal values.

While the necessary parametric data varies significantly from Method to Method some syntax is common across all Methods:

1. Methods must be uniquely declared using the Method keyword, component identifier that it acts upon and the Method identifier (e.g. RAINEVAP Madden mad_re). This declaration indicates the beginning of this Method's definition and its end must be declared by ENDRAINEVAP.
2. Method-specific time series data in the input deck may have a distribution applied to them. If a distribution is used, there must be one distribution value for each time step in a day (e.g., 4 values for a 6-hour time step) and the values must sum to 1.0. If no distribution is specified, a uniform distribution is assumed.

The following Sections describe each Method.

The following is an example of a complete RES-J input deck:

```

RES-J      GAT&MAD

TIMESERIES
  TIMESTEP      01
  INPUT  GATUN  MAP  01  GAT_MAP      # GATUN PRECIPITATION
  INPUT  MADDENSM SQIN 01  MAD_INFLOW  # LOCAL INFLOWS
  INPUT  GATUNSM SQIN 01  GAT_INFLOW
  INPUT  MADPOOL PELV 01  MAD_OBS_POOL # OBSERVED POOL11G
  INPUT  GATPOOL PELV 01  GAT_OBS_POOL
  INPUT  MADSPILL RQSW 01  MAD_OBS_SPILL # OBSERVED SPILL
  INPUT  GATSPILL RQSW 01  GAT_OBS_SPILL
  INPUT  MADPOWER RQIE 01  MAD_OBS_POWER # OBSERVED POWER
  INPUT  GATPOWER RQIE 01  GAT_OBS_POWER
  INPUT  MADMUNI DQIN 01  MAD_OBS_MUNI # OBSERVED MUNICIPAL DEMAND
  INPUT  GATMUNI DQIN 01  GAT_OBS_MUNI
  INPUT  GATUNLOX DQIN 01  GAT_OBS_LOCK # OBSERVED GATUN LOCK DEMAND
  INPUT  PEDMGLOX DQIN 01  PED_OBS_LOCK # OBSERVED PEDRO MIGUEL LOCK DEMAND
  OUTPUT GATWITHD SDQI 01  GAT_SIM_WITH # SIMULATED WITHDRAW
  OUTPUT MADWITHD SDQI 01  MAD_SIM_WITH

```

```

OUTPUT  GATRELES RQIE 01 GAT_SIM_REL      # SIMULATED RELEASE
OUTPUT  MADRELES RQIE 01 MAD_SIM_REL
OUTPUT  GATPOOL  PELE 01 GAT_SIM_POOL     # SIMULATED POOL
OUTPUT  MADPOOL  PELE 01 MAD_SIM_POOL
ENDTIMESERIES

```

TOPOLOGY

```

RESERVOIR      MADDEN
REACH          CHAGRES BELOW MADDEN
RESERVOIR      GATUN BELOW CHAGRES
ENDTOPOLOGY

```

PARAMETERS

UNITS ENGLISH

RESERVOIR MADDEN

```

CONSTANT MADDEN.DROUGHT_ELEVATION_205 205.0
CONSTANT MADDEN.FLOOD_ELEVATION_253 253.0
TSINPUT  INFLOW MADDENSM SQIN 01
TSOUTPUT RELEASE MAD_SIM_REL
TSOUTPUT WITHDRAW MAD_SIM_WITH
TSOUTPUT POOL MAD_SIM_POOL
TABLE    ELEV_STOR
  72.0    0.0
  201.0   184068.0
  202.0   189922.0
  203.0   195914.0
  ...
  272.0   913797.0
  273.0   927319.0
  274.0   940840.0
ENDTABLE
INITIALPOOL      247.0
INITIALRELEASE   1200
INITIALWITHDRAW  1000
ENDRESERVOIR

```

REACH CHAGRES

ENDREACH

RESERVOIR GATUN

```

CONSTANT GATUN.DROUGHT_ELEVATION_81P5 81.5
CONSTANT GATUN.FLOOD_ELEVATION_88 88.0
TSINPUT  INFLOW GAT_INFLOW
TSOUTPUT RELEASE GAT_SIM_REL
TSOUTPUT WITHDRAW GAT_SIM_WITH
TSOUTPUT POOL GAT_SIM_POOL
TABLE    ELEV_STOR
  40.0    0.0
  77.0   3393274.0
  77.5   3440588.0
  78.0   3487925.0
  ...
  98.5   5666244.0
  99.0   5721423.0
  99.5   5776607.0
  100.0  5831807.0
ENDTABLE
INITIALPOOL      85.5
INITIALRELEASE   1200
INITIALWITHDRAW  1000
ENDRESERVOIR

```

MADDEN COMPONENT METHODS

DAILY MUNICIPAL DEMANDS FOR MADDEN RESERVOIR. THESE VALUES REPRESENT
THE AVERAGE DAILY WITHDRAWAL IN CFS/D AND WERE COMPILED USING THE

AVERAGE DEMAND OVER THE YEARS 1994-1996.

SETWITHDRAW MADDEN MUNICIPAL DEMAND

TSINPUT OBSERVEDWITHDRAW MAD_OBS_MUNI

VALUES

ELEV 0.0 ENDELEV

01/01 189.0

02/01 189.0

03/01 192.0

04/01 193.0

05/01 192.0

06/01 187.0

07/01 187.0

08/01 185.0

09/01 187.0

10/01 179.1

11/01 183.0

12/01 184.0

ENDVALUES

BLENDTBL 0 # NO BLENDING

BLENDTS 0 # NO BLENDING OF TIME SERIES VALUES

NORMAL # NO INTERPOLATION BETWEEN DATES

ENDSETWITHDRAW

RELEASE WATER FROM MADDEN RESERVOIR IN ORDER TO KEEP POOL ELEVATION AT A
CONSTANT 205 FT. THIS ELEVATION REPRESENTS THE MINIMUM POOL ELEVATION AT
WHICH MADDEN LAKE SHOULD BE KEPT.

SETELEVATION MADDEN POOLELEVATION_205

VALUES

01/01 205.0

ENDVALUES

BLENDTBL 0

BLENDTS 0

NORMAL

ENDSETELEVATION

RELEASES FOR POWER AT MADDEN RESERVOIR. RELEASES FOR POWER ARE MADE BASED
BOTH ON TIME OF YEAR AND CURRENT POOL ELEVATION. THE SETRELEASE METHOD
BELOW ASSUMES FULL POWER IS APPROXIMATELY 3500 CFS/D. NO ALLOWANCES FOR
DIURNAL POWER HAS BEEN INCLUDED WITHIN THIS METHOD YET. THIS TABLE IS A
VERY ROUGH ESTIMATE OF WATER USE IN GENERATION BY ANALYSIS OF THE YEARS
1987-1996 WATER USE IN GENERATION. THE RELEASE FOR 215 AND BELOW IS BASED
ON THE ALL TIME MONTHLY LOW FOR THE 10 YEAR PERIOD.

SETRELEASE MADDEN POWER

TSINPUT OBSERVEDRELEASE MADPOWER RQIE 01

VALUES

ELEV 215 217 222 228 233 236 243 247 249 251 252 ENDELEV

01/01 1249 1250 1250 1300 1400 1600 1800 2000 2200 2400 2600

02/01 1476 1500 1700 1800 1900 2000 2100 2200 2400 2600 2800

03/01 1441 1500 1700 1800 1900 2000 2100 2200 2400 2600 2800

04/01 954 1200 1400 1600 1800 2000 2200 2400 2600 2800 3000

05/01 1302 1500 1700 2000 2000 2300 2400 2400 3000 3000 3000

06/01 763 1502 1500 1700 2000 2000 2300 2400 2400 3000 3000

07/01 1639 1700 1700 1800 1900 2000 2100 2200 2300 2400 2500

08/01 1019 1200 1400 1600 1800 2000 2200 2400 2600 2800 3000

09/01 1606 1800 2000 2200 2400 2600 2800 3000 3200 3500 3500

10/01 1645 1800 2000 2200 2400 2600 2800 3000 3200 3500 3500

11/01 1981 1800 2000 2200 2400 2600 2800 3000 3200 3500 3500

12/01 1506 1800 2000 2200 2400 2600 2800 3000 3200 3500 3500

ENDVALUES

BLENDTBL 0

BLENDTS 0

NORMAL

ENDSETRELEASE

CRITICAL FLOODING PERIOD FOCUS ON SAVING THE DAM AT MADDEN BASED ON FLOOD
CONTROL MANUAL PMF CASE SCENARIO. REST IS BASED ON HISTORICAL SPILLS AT
MADDEN (1987-1996).

```

SETRELEASE MADDEN SPILL
  TSINPUT OBSERVEDRELEASE MAD_OBS_SPILL
  VALUES
    ELEV      0 250.0 252.5 255.00 255.78 263.00 ENDELEV
    01/01  0 250 2000 75000 168000 226000
  ENDVALUES
  BLENDTBL 0
  BLENDTS 0
  NORMAL
  ENDSETRELEASE

```

```

SETSUM MADDEN POWER&SPILLS
  SETRELEASE MADDEN POWER
  SETRELEASE MADDEN SPILL
  ENDSETSUM

```

```

ADJUST MADDEN MADDEN_ADJUST
  OBSERVEDPOOL MAD_OBS_POOL
  BLENDTS 5
  ENDADJUST

```

```
# GATUN COMPONENT METHODS
```

```
# THE RAINEVAP METHOD ACCOUNTS FOR DIRECT RAINFALL AND EVAPORATION OVER THE
# GATUN LAKE SURFACE. AVERAGE DAILY EVAPORATION VALUES HAVE BEEN TAKEN FROM
# DAILY PAN VALUES.
```

```

RAINEVAP GATUN GATUN_RAINEVAP
  TSINPUT OBSERVEDPRECIP GAT_MAP
  VALUES
    01/15 0.127
    02/15 0.148
    03/15 0.154
    05/15 0.104
    06/15 0.091
    07/15 0.094
    08/15 0.093
    09/15 0.097
    10/15 0.095
    11/15 0.093
    12/15 0.101
  ENDVALUES
  ENDEVAP

```

```
# DAILY MUNICIPAL DEMANDS FOR GATUN RESERVOIR. THESE VALUES REPRESENT THE
# AVERAGE DAILY WITHDRAWAL IN CFS/D AND WERE COMPILED USING THE 1987-1996
# HISTORICAL DATA.
```

```

SETWITHDRAW GATUN MUNICIPAL_DEMAND
  TSINPUT OBSERVEDWITHDRAW GAT_OBS_MUNI
  VALUES
    ELEV      80.0 ENDELEV
    01/01  122.2
    02/01  127.1
    03/01  119.0
    04/01  128.7
    05/01  113.9
    06/01  122.3
    07/01  119.8
    08/01  119.8
    09/01  113.7
    10/01  115.2
    11/01  122.9
    12/01  121.8
  ENDVALUES
  BLENDTBL 0
  BLENDTS 0
  NORMAL
  ENDSETWITHDRAW

```

AVERAGE LOCKS DEMAND FOR GATUN LOCKS. VALUES BELOW 81.5 FT REPRESENT 80%
 # OF THE AVERAGE WATER USE FOR THE GATUN LOCKS OVER THE YEARS 1992-1995.
 # VALUES FOR POOL ELEVATIONS ABOVE 81.5 REPRESENT 100% OF THE AVERAGE WATER
 # USE FOR THE GATUN LOCKS OVER THE YEARS 1992-1996.
 # VALUES ARE REPRESENTED IN CFS/D.

SETWITHDRAW GATUN GATUNLOCKS
 TSINPUT OBSERVEDWITHDRAW GAT_OBS_LOCK
 VALUES

	ELEV	0	81.5	ENDELEV
01/01	1266.0		1581.0	
02/01	1257.0		1570.0	
03/01	1259.0		1573.0	
04/01	1223.0		1530.0	
05/01	1156.0		1445.0	
06/01	1113.0		1393.0	
07/01	1124.0		1404.0	
08/01	1146.0		1431.0	
09/01	1104.0		1379.0	
10/01	1215.0		1518.0	
11/01	1158.0		1448.0	
12/01	1190.0		1486.0	

ENDVALUES
 BLENDTBL 0
 BLENDTS 0
 NORMAL
 ENDSETWITHDRAW

AVERAGE LOCK DEMAND FOR THE PEDRO MIGUEL LOCKS. THESE VALUES REPRESENT 100%
 # OF THE AVERAGE WATER USE FOR THE PEDRO MIGUE LOCKS OVER THE YEARS 1992-1996.
 # VALUES ARE REPRESENTED IN CFS/D.

SETWITHDRAW GATUN PEDROMIGUELLOCKS
 TSINPUT OBSERVEDWITHDRAW PED_OBS_LOCK
 VALUES

	ELEV	0.0	81.5	ENDELEV
01/01	1235.1		1543.8	
02/01	1268.8		1586.1	
03/01	1284.3		1605.4	
04/01	1247.8		1559.8	
05/01	1177.7		1472.1	
06/01	1125.9		1407.4	
07/01	1131.4		1414.3	
08/01	1144.6		1430.7	
09/01	1082.1		1352.7	
10/01	1164.0		1455.0	
11/01	1129.3		1411.7	
12/01	1216.8		1521.1	

ENDVALUES
 BLENDTBL 0
 BLENDTS 0
 NORMAL
 ENDSETWITHDRAW

SETSUM GATUN MUNI&LOCKS
 SETWITHDRAW GATUN MUNICIPAL_DEMAND
 SETWITHDRAW GATUN GATUNLOCKS
 SETWITHDRAW GATUN PEDROMIGUELLOCKS
 ENDSETSUM

SETRELEASE GATUN POWER
 TSINPUT OBSERVEDRELEASE GAT_OBS_POWER
 VALUES

	ELEV	0	84.95	85.25	85.65	86.15	87.55	87.75	ENDELEV
01/01	0	0	0	0	0	0	0	0	
04/01	0	0	0	1500	1500	1500	3000		
05/01	0	1500	1500	1500	3000	4500	4500		
10/01	0	0	1500	1500	3000	3000	4500		

```

11/01 0 0 0 0 1500 3000 4500
ENDVALUES
BLENDTBL 0
BLENDTS 0
NORMAL
ENDSETRELEASE

SETRELEASE GATUN SPILL
TSINPUT OBSERVEDRELEASE GAT_OBS_SPILL
VALUES
ELEV 0 87.60 87.75 88.00 89.14 92.55 ENDELEV
01/01 0 26000 52000 182000 268000 329000
ENDVALUES
BLENDTBL 0
BLENDTS 0
NORMAL
ENDSETRELEASE

SETSUM GATUN POWER&SPILLS
SETRELEASE GATUN POWER
SETRELEASE GATUN SPILL
ENDSETSUM

ADJUST GATUN GATUN_ADJUST
TSINPUT OBSERVEDPOOL GAT_OBS_POOL
BLENDTS 5
ENDADJUST

# CHAGRES COMPONENT METHODS

# CURRENTLY THE REACH TIME BETWEEN MADDEN AND GAMBOA IS APPROXIMATELY 9 HOURS.
# CURRENTLY, NO ATTENUATION HAS BEEN IMPLEMENTED WITHIN THIS IMPLEMENTATION
# OF THE LAG/K METHOD.
LAGK CHAGRES CHAGRESLAGK
LAG 9 # 9 HOURS
K 0
COINFLOW
VALUES
1000 # EACH CARRYOVER VALUE MUST HAVE SEPARATE LINE OF DATA
1000 # SEPARATE LINE OF DATA
1000
1000 #Must have LAG/TIMESTEP+1
1000 #10 carryover values for the Inflow
1000
1000
1000
1000
ENDVALUES
ENDCOINFLOW
ENDLAGK

ENDPARAMETERS

RULES

# ALL THE RULES EXPRESSIONS ARE EXECUTED FOR EVERY TIME STEP TO DETERMINE IF THE
#EXPRESSION IS TRUE OR FALSE. IF THE EXPRESSION IS TRUE, ALL THE METHODS UNDER
#THIS EXPRESSION WILL BE EXECUTED. THE METHODS EXECUTED IN AN EARLIER TRUE
#EXPRESSION WILL BE OVERWRITTEN BY THE METHODS FROM A LATER TRUE EXPRESSION.

[TRUE]

# THE FOLLOWING METHODS ARE EXECUTED FOR EVERY TIME STEP:

# METHODS FOR MADDEN RESERVOIR

# SETWITHDRAW ACCOUNTS FOR WATER REMOVED FROM THE RESERVOIR FOR MUNICIPAL

```

```

# CONSUMPTIVE DEMAND
::SETWITHDRAW MADDEN MUNICIPAL_DEMAND

# SETSUM ACCOUNTS FOR WATER WHICH IS RELEASED FROM MADDEN EITHER THROUGH
# GENERATION OR BY SPILL.
::SETSUM MADDEN POWER&SPILLS
#::SETRELEASE MADDEN POWER
#::SETRELEASE MADDEN SPILL

# ADJUST MADDEN'S SIMULATED RELEASE AND POOL ELEVATIONS WITH OBSERVATIONS
# WHEN POSSIBLE
::ADJUST MADDEN MADDEN_ADJUST

# LAG/K ROUTES WATER FROM MADDEN TO GATUN VIA THE CHAGRES REACH
::LAGK CHAGRES CHAGRESLAGK

# RAINEVAP ACCOUNTS FOR DIRECT RAINFALL AND EVAPORATION OVER THE GATUN LAKE
# SURFACE.
::RAINEVAP GATUN GATUN_RAINEVAP

# SETSUM ACCOUNTS FOR WATER WHICH IS WITHDRAWN FROM GATUN EITHER FOR
# MUNICIPAL CONSUMPTION OR LOCK USAGE
::SETSUM GATUN MUNI&LOCKS
#::SETWITHDRAW GATUN MUNICIPAL_DEMAND
#::SETWITHDRAW GATUN GATUNLOCKS
#::SETWITHDRAW GATUN PEDROMIGUELLOCKS

# SETSUM ACCOUNTS FOR WATER WHICH IS RELEASED FROM GATUN EITHER THROUGH
# GENERATION OR BY SPILL.
::SETSUM GATUN POWER&SPILLS
#::SETRELEASE GATUN POWER
#::SETRELEASE GATUN SPILL

# ADJUST GATUN'S SIMULATED RELEASE AND POOL ELEVATIONS WITH OBSERVATIONS
# WHEN POSSIBLE
::ADJUST GATUN GATUN_ADJUST

##### SEVERE DROUGHT CONDITIONS #####
# MADDEN LAKE IS ABOVE THE 205 FOOT POOL LEVEL, GATUN LAKE IS BELOW THE 81.5
# FOOT POOL LEVEL. UNDER THIS SCENARIO, MADDEN RESERVOIR MAKES RELEASES FOR
# MUNICIPAL DEMANDS AND RELEASES WATER TO GATUN RESERVOIR UNTIL THE 205 POOL
# LEVEL IS ACHIEVED AT MADDEN. GATUN RESERVOIR IS LIMITED TO MUNICIPAL
# WITHDRAWALS AS WELL AS WITHDRAWALS FOR RESTRICTED GATUN LOCK AND RESTRICTED
# PEDRO MIGUEL LOCK OPERATION.
[ ((MADDEN.POOL > MADDEN.DROUGHT_ELEVATION_205) && \
  (GATUN.POOL < GATUN.DROUGHT_ELEVATION_81P5)) ]
  # MADDEN RESERVOIR
  ::SETELEVATION MADDEN POOLELEVATION_205

ENDRULES

ENDRES-J

```

Name Type Description
[\[Back\]](#) [\[Next\]](#) [\[Previous\]](#) [\[Bookmarks\]](#) [\[Top\]](#)

Table 1. RES-J Methods

<u>Name</u>	<u>Type</u>	<u>Description</u>
ADJUST	Reservoir	Adjusts reservoir output [Bookmark]
BALANCE	Reservoir	Balances reservoir storage for multiple reservoir systems [Bookmark]
CALCINFLOW	Reservoir	Calculates reservoir inflow using mass balance [Bookmark]
LAGK	Reach	Routes water through a reach according to the LAGK routing Method [Bookmark]
LOOKUP3	Reservoir or Node	Provides flexible definition of release, withdrawal or augmentation at a Reservoir and diversion or augmentation at a Node as a function of date, current time step values of input time series, and / or component states. [Bookmark]
MAXDECREASE	Reservoir	Generates maximum daily decrease in reservoir discharge [Bookmark]
MAXINCREASE	Reservoir	Generates maximum daily increase in reservoir discharge [Bookmark]
MAXSTAGE	Reservoir	Generates a maximum release from a reservoir given a maximum allowable discharge downstream [Bookmark]
PASSFLOW	Reservoir	Passes the reservoir inflow directly as the reservoir discharge [Bookmark]
RAINEVAP	Reservoir	Accounts for direct meteorological effects on a reservoir lake surface area [Bookmark]
SETELEVATION	Reservoir	Prescribes reservoir pool elevation [Bookmark]
SETMAX	Reservoir	Selects the maximum release of any two or more specified reservoir Methods [Bookmark]
SETMIN	Reservoir	Selects the minimum release of any two or more specified reservoir Methods [Bookmark]

<u>Name</u>	<u>Type</u>	<u>Description</u>
SETRELEASE	Reservoir	Generates a single reservoir release based on multiple criteria [Bookmark]
SETSUM	Reservoir	Sums prescribed releases [Bookmark]
SETWITHDRAW	Reservoir	Generates a single reservoir withdrawal based on multiple criteria [Bookmark]
SPILLWAY	Reservoir	Augments a single reservoir's previously determined release according to the pseudo-implicit solution of pool elevation versus spill [Bookmark]

Table 2. Time series keyword parameters

<u>Applicable Types</u>	<u>Time Series Keywords</u>	<u>Description</u>	<u>Input/Output</u>
All components	INFLOW	Inflow	Input
All components	TOTALINFLOW	Simulated Inflow	Output
CALCINFLOW Method	POOL	Pool Elevation	Input
CALCINFLOW Method	RELEASE	Release	Input
CALCINFLOW Method	WITHDRAW	Withdrawal	Input
CALCINFLOW Method	INFLOW	Calculated Inflow	Output
LOOKUP3 Method	OBSERVEDWITHDRAWAL	Observed reservoir Withdrawal	Input
LOOKUP3 Method	OBSERVEDDIVERSION	Observed node Diversion	Input
LOOKUP3 Method	OBSERVEDAUGMENTATION	Observed reservoir Or Node Augmentation	Input
LOOKUP3 Method	ROWS	Index to table rows	Input Index
LOOKUP3 Method	COLUMNS	Index to table Columns	Input Index
LOOKUP3 Method	RELEASE	Release	Input
LOOKUP3 Method	WITHDRAW	Withdrawal	Input
LOOKUP3 Method	INFLOW	Calculated Inflow	Output
RAINEVAP Method	OBSERVEDEVAP	Evaporation	Input
RAINEVAP Method	OBSERVEDPRECIP	Precipitation	Input
SETELEVATION, ADJUST Methods	OBSERVEDPOOL	Observed reservoir pool elevation	Input
SETRELEASE, ADJUST, LOOKUP3 Methods	OBSERVEDRELEASE	Observed reservoir release	Input
SETWITHDRAW Method	OBSERVEDWITHDRAW	Observed reservoir withdrawals	Input
Reservoir	POOL	Pool elevation	Output

component			
Reservoir component	RELEASE	Reservoir release	Output
Reservoir component	STORAGE	Reservoir storage	Output
Reservoir component	WITHDRAW	Reservoir withdrawals	Output
Reach component	INFLOW	Reach inflow	Input
Reach component	OUTFLOW/RELEASE	Reach outflow	Output
Node component	INFLOW	Node inflow	Input
Node component	DIVERSION	Diversion from Node	Output
Node component	OUTFLOW	Node outflow	Output

Table 3. RES-J Methods, components and variables

<u>Method Type</u>	<u>Component Acted Upon</u>	<u>Solves For</u>
ADJUST	Reservoir	Release
BALANCE	Reservoir	Release
CALCINFLOW	Self and Reservoir	Inflow
LAGK	Reach	Outflow
LOOKUP3	Reservoir	Release, Withdrawal (Augmentation)
	Node	Diversion, (Augmentation)
MAXDECREASE	Reservoir	Release
MAXINCREASE	Reservoir	Release
MAXSTAGE	Reservoir	Release
PASSFLOW	Reservoir	Release
RAINEVAP	Reservoir	Loss
SETELEVATION	Reservoir	Release
SETMAX	Reservoir, Reach	Combination
SETMIN	Reservoir, Reach	Combination
SETRELEASE	Reservoir	Release
SETSUM	Reservoir, Reach	Combination
SETWITHDRAW	Reservoir	Withdrawal
SPILLWAY	Reservoir	Release (spill augmented) and pool elevation

V.3.3-RES-J-ADJUST JOINT RESERVOIR REGULATION OPERATION METHOD ADJUST

Description

Method ADJUST is applied to a reservoir to adjust the release and/or pool elevation to observed values.

Input

<u>Keyword</u>	<u>Definition</u>
ADJUST	Keyword specifying the beginning of input for an ADJUST Method; must be followed by: <ul style="list-style-type: none">- component identifier- Method identifier
TSINPUT	Keyword specifying an input time series; must be followed by: <ul style="list-style-type: none">- 'OBSERVEDPOOL' or 'OBSERVEDRELEASE'- time series alias
[BLENDTS]	Causes blending of observed time series. Blending occurs from the last non-missing observed release to the release prescribed by the last Method active which prescribed a reservoir release over a given number of time steps. 'BLENDTS' must be followed by: <ul style="list-style-type: none">- an integer number of time steps over which to blend- an optional integer number representing the next time step counter within an ongoing blend; defaults to the blend value + 1 (representing a complete blend); this value is used to initialize carryover <u>1</u>/
[ADJSIM]	Keyword used to explicitly define whether the Adjust Method will adjust the simulation each timestep or will only adjust the states being saved into carryover (leaving simulation untouched). Defaults to adjusting the simulation at each timestep (equivalent to 'ADJSIM ON'). 'ADJSIM' must be followed by: <ul style="list-style-type: none">- 'ON' or 'OFF' <u>2</u>/ <u>3</u>/
ENDADJUST	

1/ BLENDTS is ignored when ADJSIM is set to OFF. No blending is done.

- 2/ When preparing the carryover array containing the states of the reservoir, ADJSIM OFF will replace the simulated states with observed states in input observation pool and / or release time series as available. If a time series does not exist or is missing data for either the current timestep (carryover date) or the previous timestep (previous date), the simulated value is preserved.
- 3/ An ADJUST Method containing 'ADJSIM OFF' should be included in the rules section under the [TRUE] conditional expression. Because the Method affects nothing directly in the simulation but does its work 'between' time steps, evaluation of the conditional expression is not considered. The Method will activate at each carryover save date.

Example

```
ADJUST RESERVOIRNAME ADJUSTNAME
  TSINPUT OBSERVEDRELEASE ALIAS
                                     #RELEASE TIME SERIES TO WHICH THE
                                     #SIMULATED TIME SERIES IS ADJUSTED
  BLENDS 3                           #BLENDS THE OBSERVED RELEASE INTO
                                     #THE SIMULATED RELEASE OVER
                                     #THREE TIME STEPS.
  ADJSIM ON                           #THE SIMULATION WILL BE ADJUSTED AT
                                     #EACH TIMESTEP.
ENDADJUST
```

Keyword Definition

[Back] [Next] [Previous] [Bookmarks] [Top]

V.3.3-RES-J-BALANCE JOINT RESERVOIR REGULATION OPERATION METHOD
BALANCE

Description

Method BALANCE computes the release from a reservoir in an attempt to balance the available flood storage in a system of reservoirs. The calculated release is then subject to minimum and maximum limits.

Input

Keyword Definition

BALANCE	Keyword indicating the start of input for Method Balance. Must be followed by: <ul style="list-style-type: none">- component identifier (reservoir 'owning' this BALANCE Method)- Method identifier
VOLUME	Keyword indicating that balancing should be performed by distributing the available storage evenly across the BALANCE reservoirs. Specify either VOLUME or PERCENT, but not both.
PERCENT	Keyword indicating that balancing should be performed by applying the percent of the total flood storage still available (over all BALANCE reservoirs) to the 'owning' reservoir. Specify either VOLUME or PERCENT, but not both.
ACE	Keyword pseudonym for the Attenuation Constant for Evacuation. The keyword should be followed by a value, generally between 0.0 and 1.0, used as a coefficient to scale the evacuated change in storage required to reach balance. While BALANCE determines what storage is required for balancing it does not understand what impacts the release required to reach that target storage will have on the balance of the system. ACE reduces the zeal of BALANCE causing it to approach a balanced condition over more than one timestep, thus preventing oscillations in the reservoir system. A suggested first-try value for ACE can be calculated as the model timestep divided by the longest lag time in the BALANCE reservoir system from the owning reservoir to the upstream-most or downstream-most reservoir. <u>1/ 2/</u>
BALRES	Keyword specifying one of the reservoirs in the BALANCE system. Must be followed by:

<u>Keyword</u>	<u>Definition</u>
VALUES	- component name 3/ Keyword indicating the start of a list of values for the previously identified reservoir.
[LOWER POOL]	Keyword should be followed by the lowest pool elevation to be considered as part of the flood storage pool for the reservoir. Defaults to minimum elevation in the elevation-storage table. Specify either LOWER POOL or LOWER STORAGE, but not both. <ul style="list-style-type: none"> - If balancing in PERCENT mode then LOWER POOL (or LOWER STORAGE) is required for all reservoirs. - If balancing in VOLUME mode then LOWER POOL (or LOWER STORAGE) is only required for the owning reservoir.
[LOWER STORAGE]	Keyword should be followed by the lowest pool storage to be considered as part of the flood storage pool for the reservoir. Default to minimum storage in the elevation-storage table. Specify either LOWER POOL or LOWER STORAGE, but not both. <ul style="list-style-type: none"> - If balancing in PERCENT mode then LOWER STORAGE (or LOWER POOL) is required for all reservoirs. - If balancing in VOLUME mode then LOWER STORAGE (or LOWER POOL) is only required for the owning reservoir.
[UPPER POOL]	Keyword should be followed by the highest pool elevation to be considered as part of the flood storage pool for the reservoir. Default to maximum elevation in the elevation-storage table. Specify either UPPER POOL or UPPER STORAGE for all reservoirs (independent of balancing mode), but not both.
[UPPER STORAGE]	Keyword should be followed by the highest pool storage to be considered as part of the flood storage pool for the reservoir. Default to maximum storage in the elevation-storage table. Specify either UPPER POOL or UPPER STORAGE for all reservoirs (independent of balancing mode), but not both.
MINRELEASE	Keyword should be followed by the minimum release from the reservoir that the BALANCE Method will specify.
MAXRELEASE	Keyword should be followed by the maximum release from the reservoir that the BALANCE Method will specify.
ENDVALUES	Keyword indicating the end of the list of values for the downstream reservoir.

<u>Keyword</u>	<u>Definition</u>
ENDBALRES	Keyword indicating the end of the parameters for the reservoir. <u>3/</u>
Repeat the above keywords, BALRES through ENDBALRES for each reservoir in the BALANCE system.	
ENDBALANCE	Keyword indicating the end of the parameters for the Balance Method.

- 1/ ACE can also be specified using its full name:
ATTENUATION_CONSTANT_FOR_EVACUATION
- 2/ ACE is only used when the current pool elevation is above the target pool elevation for a balanced state. Thus it reduces releases from the owning reservoir which often cause a downstream reservoir's pool elevation to rise, throwing the system beyond balance and introducing an oscillation into the system. One can imagine a case where an upstream reservoir releases a large pulse of water to arrive at balance with a downstream reservoir only to find that the pulse caused the downstream reservoir's pool elevation to rise, preventing the balanced condition.
- 3/ BALRES can also be specified using the historical syntax keyword RESERVOIR. Use of RESERVOIR is discouraged, however, as it requires special handling to avoid confusion with the keyword used to specify a reservoir component. ENDRESERVOIR can also replace ENDBALRES although its use is also discouraged.

Example

```

BALANCE OWNER_RESERVOIRNAME BALANCENAME
  VOLUME          #OR PERCENT MODE OF BALANCING.
  ACE 0.33                #VALUE WILL MULTIPLY AN
                          #EVACUATION TO PREVENT
                          #OSCILLATION.
  BALRES DSRESNAME      #NAME OF A RESERVOIR.
    VALUES
      LOWER POOL 135    #OPTIONAL - WILL USE THE LOWEST
                        #ELEVATION FROM DSRESNAME'S
                        #ELEV_STOR BY DEFAULT.
                        #ALSO, NOT REQUIRED FOR VOLUME
                        #MODE BALANCING SINCE DRESNAME
                        #IS NOT THE OWNING RESERVOIR.

                        UPPER POOL 140    #OPTIONAL - WILL USE THE
                        #HIGHEST ELEVATION FROM
                        #DSRESNAME'S ELEV_STOR BY
                        #DEFAULT.
      MINRELEASE 12.4
      MAXRELEASE 500.
    ENDVALUES
  ENDBALRES
  BALRES OWNER_RESERVOIRNAME #NAME OF THE RESERVOIR ON WHICH
                              #THIS METHOD WAS DECLARED.
    VALUES
      LOWER STORAGE 12004342 #UPPER AND LOWER BOUNDS CAN BE

```

UPPER STORAGE 18922000 #IN TERMS OF
MINRELEASE 11.8 #POOL ELEVATION OR STORAGE.
MAXRELEASE 450.
ENDVALUES
ENDBALRES
ENDBALANCE

Keyword

Definition

[Back] [Next] [Previous] [Bookmarks] [Top]

V.3.3-RES-J-CALCINFLOW JOINT RESERVOIR REGULATION OPERATION METHOD
CALCINFLOW

Description

Method CALCINFLOW determines the average inflow over a time step to a reservoir based on a mass balance including 'observed' pool elevation, release, withdrawal and rain / evaporation net loss. The calculated inflow can be used immediately for simulation or simply output from Operation RES-J while proceeding with traditional simulation methods.

Method CALCINFLOW solves for the average inflow over the timestep using instantaneous pool elevations, releases, withdrawals (from input timeseries) and net loss over the timestep (from a parameterized RainEvap method). If one or more of these variables is missing then no inflow will be calculated. The inflow value is calculated as a mean value over a timestep despite the general use of instantaneous values throughout Operation RES-J. This was done to avoid oscillations inherent in the calculation of an instantaneous inflow which is a function of the instantaneous inflow from the previous time step.

A calculated inflow can be constrained according to a minimum acceptable inflow or by maximum increases and / or decreases. Any calculated inflow that is not applied to the value stored on the inflow timeseries is stored for application to the mass balance at the next timestep.

Once determined and stored on the inflow timeseries, the calculated inflow value may be transferred to the owning Reservoir component for immediate use in simulation. This immediate use will overwrite any other inflow to the Reservoir (from upstream components or local inflows) in simulation of the reservoir this timestep. If the inflow is not calculated, simulation will proceed with whatever value was determined from upstream sources. Inflow calculated by this method is average inflow over the time step. Any use of this value in the current simulation assumes that it is instantaneous inflow.

Input

Keyword

Definition

TSINPUT

Identifies a time series to be used for mass balance calculation of inflow. 'TSINPUT' must be followed by:

- time series keyword 'POOL', 'WITHDRAW' or 'RELEASE'. CALCINFLOW requires a pool time series. If value of zero will be used for the withdrawal or release variable if no withdrawal

<u>Keyword</u>	<u>Definition</u>
	<p>or release time series, respectively, is provided.</p> <ul style="list-style-type: none"> - an input time series alias defined in the TIMESERIES definition section
[TSOUTPUT]	<p>Requests that the time series containing the calculated inflow be output from the method. 'TSOUTPUT' must be followed by:</p> <ul style="list-style-type: none"> - time series keyword 'INFLOW' - an input time series alias defined in the TIMESERIES definition section
[USEFORSIM]	<p>Indicates that any inflow calculated by the method should immediately override the inflow existent on the owning reservoir. USEFORSIM should be followed by:</p> <ul style="list-style-type: none"> - UseForSim keyword 'YES' or 'NO' <p>If this 'USEFORSIM' is not specified, the calculated inflow will NOT be used. If the keyword following 'USEFORSIM' is absent, CALCINFLOW assumes the user wishes to use the calculated inflow.</p>
[TABLE]	<p>Indicates the start of the maximum change in inflow constraint table. 'TABLE' must be followed by:</p> <ul style="list-style-type: none"> - table keyword 'MAXCHANGE' - three columned table where: column #1 contains an index of the previous calculated inflow, column #2 represents the maximum allowable increase in calculated inflow from the previous time step and column #3 represents the maximum allowable decrease in calculated inflow. Column #1 may contain any real number, while columns #2 and #3 should be positive values. - ENDTABLE keyword specifying the end of the table. <u>1/</u>
[INTERPOLATE]	<p>Indicates that lookup using the MaxChange table should include interpolation of values. If not specified, no interpolation will be done. <u>1//</u></p>
[USELOSS]	<p>Indicates that the Reservoir's loss value be used in CalcInflow mass balance. This assumes that a previously run (this time step) loss method (RainEvap) has assigned an appropriate net loss value to the Reservoir. If 'USELOSS' is not defined then the loss term will default to 0.</p>
[MININFLOW]	<p>Keyword indicating the use of a minimum calculated inflow constraint. Value must follow.</p>
STARTPOOL	<p>Starting pool elevation; value must follow.</p>

<u>Keyword</u>	<u>Definition</u>
[STARTWITHDRAW]	Starting withdrawal; value must follow; keyword may also be 'STARTWITHDRAWAL'
[STARTRELEASE]	Starting release; value must follow.
[STARTINFLOW]	Starting calculated inflow; value must follow.
[REMAININGVOL]	Volume calculated as part of an inflow value but not applied due to constraints in inflow. This value will be applied as part of the mass balance at the next time step. A carryover value, the user should assign this variable a value of 0 when (re)parameterizing. Carryover transfer will manage this value as necessary. <u>1/</u>
ENDCALCINFLOW	Keyword indicating the end of the parameters for the reservoir.

1/ See Section II.4-RES-J-CALCINFLOW [[Hyperlink](#)] for a description of calculated inflow constraints and remaining volume.

Carryover Transfer

Carryover values saved are REMAININGVOL, STARTINFLOW, STARTPOOL, STARTRELEASE, STARTWITHDRAWAL. The REMAININGVOL, STARTINFLOW and STARTPOOL values are transferred directly without transformation. The STARTRELEASE and STARTWITHDRAWAL values will also transfer directly, unless the previous parameterization did not contain the release or withdrawal timeseries', respectively. In this case, no transfer of these values will occur:

REMAININGVOL(NEW) = REMAININGVOL(OLD)

STARTINFLOW(NEW) = STARTINFLOW(OLD)

STARTPOOL(NEW) = STARTPOOL(OLD)

if TSINPUT RELEASE(OLD) exists:

STARTRELEASE(NEW) = STARTRELEASE(OLD)

otherwise:

STARTRELEASE(NEW) = STARTRELEASE(NEW)

if TSINPUT WITHDRAWAL(OLD) exists:

STARTWITHDRAWAL(NEW) = STARTWITHDRAWAL(OLD),

otherwise:

STARTWITHDRAWAL(NEW) = STARTWITHDRAWAL(NEW)

Example

```

CALCINFLOW  JORDAN      CALCIN-TRY
  TSINPUT POOL      OBS_POOL
  TSINPUT WITHDRAW OBS_WDRW

```

<u>Keyword</u>	<u>Definition</u>		
TSINPUT RELEASE	OBS_RELS		
TSOutput Inflow	CalcIN		
UseForSim	NO		
Table MaxChange			
	#PrevCalcIn	MaxIncrease	MaxDecrease
	-100000	10000	1000
	-10000	1000	100
	-1000	100	10
	-100	10	1
	100	10	1
	1000	100	10
	10000	1000	100
	100000	10000	1000
EndTable			
Interpolate			
MinInflow	0		
StartPool	209.7		
StartRelease	374.0		
StartWithdrawal	440		
StartInflow	100		
RemainingVol	0		
ENDCALCINFLOW			

Keyword

Definition

[Back] [Next] [Previous] [Bookmarks] [Top]

V.3.3-RES-J-LAGK JOINT RESERVOIR REGULATION OPERATION METHOD LAGK

Description

Method LAGK performs Lag and K routing for a Reach component. This method has been revised and enhanced to replicate the functionality of the LAG-K operation, including variable lag and variable K.

Each reach component must have a routing Method defined. LAGK is the only routing Method available.

Input

Keyword

Definition

LAGK

Keyword indicating the start of the input for Method LAGK. Must be followed by:
- component identifier
- Method identifier

[LAG]

Keyword used to specify either a constant lag, or the beginning of variable lag (inflow vs. lag). If a constant lag is desired, 'LAG' must be followed by:
- integer lag time, in hours

If a variable lag is desired, the 'LAG' keyword becomes optional, but still recommended. If used, it stands alone on a given line. The next line should have 'TABLE INFLOW_LAG'. Following the table parameterization, 'ENDTABLE' and 'ENDLAG' keywords should be used.

[K]

Keyword used to specify either a constant k storage coefficient, or the beginning of variable k (outflow vs k). If a constant k is desired, 'K' must be followed by:
- single value

If a variable k is desired, the 'K' keyword becomes optional, but still recommended. If used, it stands alone on a given line. The next line should have 'TABLE OUTFLOW K'. Following the table parameterization, 'ENDTABLE' and 'ENDK' keywords should be used.

<u>Keyword</u>	<u>Definition</u>
[TABLE]	<p>Keyword indicating the start of a variable parameter lag or k table. Must be followed by:</p> <ul style="list-style-type: none"> - 'INFLOW_LAG' or 'OUTFLOW_K' - Subsequent lines listing either inflow/lag or outflow/k pairs, one per line. The pairs should be in ascending order based on inflow or outflow. These values need not be integer values. - 'ENDTABLE' keyword indicating the end of the table. <p>If 'LAG' 'ENDLAG' keywords are used, 'TABLE INFLOW_LAG' and associated parameters must be within the 'LAG' 'ENDLAG' block. Similarly, if 'K' 'ENDK' keywords are used, 'TABLE OUTFLOW_K' must be within the 'K' 'ENDK' block.</p>
[ENDLAG]	<p>Keyword indicating the end of the LAG information. Recommended when 'TABLE INFLOW_LAG' specifies a variable lag.</p>
[ENDK]	<p>Keyword indicating the end of the K information. Recommended when 'TABLE OUTFLOW_K' specifies a variable K.</p>
COINFLOW	<p>Keyword indicating the start of the inflow carryover information. The VALUES / ENDVALUES sub-section must be contained within the COINFLOW / ENDCOINFLOW block. <u>1/</u></p>
VALUES	<p>Keyword indicating the start of a list of carryover values. 'VALUES' must be followed by a list of carryover values, beginning on the subsequent line, and 'ENDVALUES', constituting a sub-section within the COINFLOW / ENDCOINFLOW block.</p> <p>The required number of inflow values can be determined by comparing the computational time step and the lag.</p> <p>If lag is constant, determine the number of values as follows:</p> <ul style="list-style-type: none"> - If $(lag \text{ mod } timestep) = 0$, then the number of carryover values = $(lag \% timestep) + 1$ S If $(lag \text{ mod } timestep) > 0$, then the number of carryover values = $(lag \% timestep) + 2$ <p>('%' represents the integer division of the first number by the second). ('mod' represents the remainder of the integer division of the first number by the second).</p>

<u>Keyword</u>	<u>Definition</u>
	<p>If the lag is variable with inflow, determine the number of inflow values as being the larger of two calculations:</p> <ul style="list-style-type: none"> - Use the constant lag calculation above where "lag" is the largest lag in the INFLOW_LAG table. - Count the number of entries (rows) in the INFLOW_LAG table.
[ENDVALUES]	Keyword indicating the end of the list of values, the end of the VALUES / ENDVALUES section. This section must end within the COINFLOW / ENDCOINFLOW block.
[ENDCOINFLOW]	Keyword indicating the end of the inflow carryover information (the COINFLOW / ENDCOINFLOW block).
[INITIALOUTFLOW]	Indicates the initial discharge from the reach; value must follow. Defaults to 0.0. If K = 0, this value must be consistent with the COINFLOW values or RES-J will force it to be so.
[INITIALSTORAGE]	Indicates the initial storage value used in the K calculation.
[INITIALLAGGEDINFLOW]	Indicates the initial lagged inflow used in the K calculation.
[KFTWLOSS]	<p>Keyword indicating that the Fort Worth recession loss calculation should be performed. 'KFTWLOSS' must be followed by:</p> <ul style="list-style-type: none"> - value representing the minimum discharge threshold, above which the calculation will be made. - loss factor value applied to the recession portion of the hydrograph. <p>See also the LAG-K operation documentation for further discussion of this parameter and its usage.</p>
ENDLAGK	Keyword indicating the end of the LAG/K Method parameters.

Carryover Transfer

Carryover values saved are inflow, initial outflow, initial storage and initial lagged inflow values.

Inflow carryover consists of a pre-defined number of inflows with the 'oldest' inflow value occurring first, proceeding to the inflow 'current' at the time of carryover output. These values are un-

Keyword Definition

lagged inflows to the reach. Thus, only a change in Lag time will affect the inflow carryover.

The inflow carryover transfer is a function of the number of required carryover values as defined by the lag. For each lag (OLD and NEW), the number of carryover values required (nValsNEW, nValsOLD) is calculated as described above in the VALUES section.

Using nValsNEW and nValsOLD as decision criteria, the following carryover rules apply:

If nValsNEW == nValsOLD, then
 $Q[i](NEW) = Q[i](OLD)$ for $0 \leq i < nValsOLD$

If nValsNEW < nValsOLD, then
 diffN = nValsOLD - nValsNEW and
 $Q[i](NEW) = Q[i+diffN](OLD)$ for $0 \leq i < nValsNEW$
 In essence, this trims off the 'oldest' inflows that with the new, shorter lag time, would have already passed through the reach.

If nValsNEW > nValsOLD, then
 diffN = nValsNEW - nValsOLD and
 $Q[i](NEW) = 0.000000$ for $0 \leq i < diffN$ and
 $Q[i+diffN](NEW) = Q[i](OLD)$ for $0 < i \leq nValsOLD$
 In essence, this files the 'oldest' (unavailable) inflows with 0.000000 although these unavailable inflows would still be traveling through the reach with the new, longer lag time.

where Q is the array of inflow values to the LAGK Method as stored in carryover
 I is the index used to access values in Q
 (NEW) is the new carryover array
 (OLD) is the old carryover array

Outflow carryover consists of the INITIALOUTFLOW value only. Transfer of the carryover is direct, without transformation:

$$INITIALOUTFLOW(NEW) = INITIALOUTFLOW(OLD)$$

Storage carryover consists of the INITIALSTORAGE value only. Transfer of the carryover is direct, without transformation:

$$INITIALSTORAGE(NEW) = INITIALSTORAGE(OLD)$$

Lagged inflow carryover consists of the INITIALLAGGEDINFLOW value only. Transfer of the carryover is direct, without transformation:

$$INITIALLAGGEDINFLOW(NEW) = INITIALLAGGEDINFLOW(OLD)$$

Examples

LAGK RIVER ROUTING1

```

LAG 36          # Constant LAG
K 0            # Constant K
COINFLOW
VALUES         # (LAG mod 6hr timestep) = 0
  70.          # Therefore, we need 36%6+1 = 7 values.
  60.
  50.
  40.
  30.
  20.
  10.
ENDVALUES     # End of Values section
ENDCOINFLOW   # End of COINFLOW section
ENDLAGK       # End of LagK parameterization

```

```

LAGK          RIVER          ROUTING2
LAG           # Start of LAG section
              # No value, so look for table
              # Start of variable lag table
TABLE INFLOW_LAG # Largest lag is 15.
  76.  12      # 4 pairs in the table
  114. 8
  228. 10
  1500 15
ENDTABLE     # End of variable lag table
ENDLAG      # End of lag section
K 0         # Constant K
COINFLOW
VALUES      # Variable lag, so we need larger of 2 values:
  0.1       # (maxLag mod 6): (15 mod 6) = 3 > 0
  0.1       # Therefore, 15%6+2 = 4 values, or
  0.1       # 4 pairs in the INFLOW_LAG table.
  0.1       # They are both 4, so we need 4 values.
ENDVALUES
ENDCOINFLOW
ENDLAGK

```

```

LAGK          RIVER          ROUTING3
LAG           # Start of LAG section
              # No value, so look for table
              # Start of variable lag table
TABLE INFLOW_LAG # Largest lag is 12
  1.00 12      # 8 pairs in the table
  2.00 10
  3.00 9
  4.00 10
  10.00 12
  20.00 7
  30.00 6
  40.00 8
ENDTABLE     # End of variable lag table
ENDLAG      # End of lag section
K 0         # Constant K
COINFLOW
VALUES      # Variable lag, so we need larger of 2 values:
  0.1       # (maxLag mod 6): (12 mod 6) = 0

```

```

0.1          # Therefore,  $12\%6+1 = 3$  values, or
0.1          # 8 pairs in the INFLOW_LAG table.
0.1          # Take the larger, so we need 8 values.
0.1
0.1
0.1
0.1
0.1
0.1
0.1
ENDVALUES
ENDCOINFLOW
ENDLAGK

LAGK          RIVER          ROUTING4
K
TABLE OUTFLOW_K      # Variable K
  0 2
 25 1
 75 7
 80 10
100 5
200 2
250 1
350 8
550 9
ENDTABLE
ENDK
LAG
TABLE INFLOW_LAG    # Variable lag
  1.00 22
  2.00 18
  3.00 14
  4.00 10
 10.00 12
 20.00  8
 30.00  7
 40.00 12
ENDTABLE
ENDLAG
COINFLOW
VALUES          # Variable lag and K, so use larger of 2
values:
0.001          # (maxLag mod 6):  $(22 \text{ mod } 6) = 4$ 
0.001          # Therefore,  $22\%6+1 = 4$  values or
0.001          # 8 pairs in the INFLOW_LAG table.
0.001          # We need 8 values.
0.001
0.001
0.001
0.001
0.001
0.001
0.001
0.001
ENDVALUES
ENDCOINFLOW
ENDLAGK

```

V1.3.3-RES-J-LOOKUP3 JOINT RESERVOIR REGULATION OPERATION METHOD
LOOKUP3

Description

Method LOOKUP3 is very flexible and powerful. The method defines a release, withdrawal, diversion or augmentation based on an observed value or a lookup table, as a function of two other values. The lookup table is used when observed values are not available. The method will be applicable to RES-J reservoir and node components. If applied to a reservoir, the calculated LOOKUP3 value may be parameterized as a withdrawal or a release from, or an augmentation of flows into the reservoir. If applied to a node, the defined value for the LOOKUP3 Method may be parameterized as a diversion from, or an augmentation of flows at the node. To define the value, the method uses a lookup table with columns indexed by one independent variable and rows indexed by another independent variable. The value of each independent variable may be defined by:

Keyword Definition

Current time step date,
Current time step value of an input time series, or
RES-J component states (including total inflow to a reservoir or node,
reservoir pool elevation, outflow from a reservoir or node, withdrawal
from a reservoir and diversion from a node).

Input

Keyword Definition

LOOKUP3 Indicates the start of parametric information for the
LOOKUP3 method. 'LOOKUP3' must be followed by:

- S A reservoir-type component or a node-type component
defined in the TOPOLOGY definition section.
- S An identifier for this method.

[TSINPUT] Indicates the existence of an observed time series.
'TSINPUT' must be followed by one of the following time
series type keywords to define the type of variable the
time series represents. The variable type must be
consistent with the TABLEVAR keyword defined later in the
parameterization.

- 'OBSERVEDWITHDRAW(AL)' for reservoir component.
- 'OBSERVEDDIVERSION' for a node component.
- 'OBSERVEDRELEASE' for a reservoir component.
- 'OBSERVEDAugMENT(ATION)' for both node and reservoir
components.

'TSINPUT' and the time series type keyword must be
followed by:

- S An input time series alias defined in the TIMESERIES
definition section.

Negative values in the time series are not valid due to
the potential confusion of a legitimate -999 value as
representing a MISSING value. Therefore, if a value in
the time series is less than 0, it will be treated as
missing and solution will proceed to the lookup table.

However, 'TSINPUT' can be used twice in one method
parameterization if the two time series types are either
'OBSERVEDWITHDRAW(AL)' and 'OBSERVEDAugMENT(ATION)' at a
reservoir or 'OBSERVEDDIVERSION' and
'OBSERVEDAugMENT(ATION)' at a node. In this fashion, a
'TSINPUT OBSERVEDWITHDRAW' time series with positive
values can be used to withdraw water from the reservoir
for certain time steps and a 'TSINPUT OBSERVEDAugMENT'
time series with positive values can be used to augment
available water in the reservoir for certain time steps.

Keyword Definition

Similarly, diversion and augmentation time series may both be parameterized at a node.

In parameterizations where TSINPUT keywords are used for both withdrawals and augmentations or diversions and augmentations, any value treated as missing from either time series will cause both time series values to be treated as missing, resulting in use of the lookup table for solution at the current time step. If positive values exist in both time series for the same time step, the difference between the OBSERVEDWITHDRAW (or OBSERVEDDIVERSION) and the OBSERVEDDAUGMENTATION will be calculated and the result will be handled according to the positive / negative nomenclature described in the TABLEVAR section below.

[TSINDEX] Indicates that an external time series be used as one of the independent variables in applying the lookup table. This keyword can be used more than once if both indices are to be associated with time series. 'TSINDEX' must be followed by:

S 'COLUMNS' or 'ROWS' specifying which axis of the table with which to associate this time series.

S An input time series alias defined in the TIMESERIES definition section.

COLUMNVAR Indicates what variable the columns in the lookup table represent. 'COLUMNVAR' must be followed by one of the following:

- 'DATE' meaning the column indexes are to be indexed by date and accessed using the date/time of the current time step.

- 'INPUT_TS' meaning the columns are to be accessed using the current time step value in an input time series specified previously using the 'TSINDEX COLUMNS' keywords. Depending on the data type of the input time series, its value could be snow water equivalent, pool elevation, inflow, outflow, etc.

'INPUT_TS' may be followed by a unit type used in defining the column headings. RES-J will convert the column indexes from this unit type to the standard FCST units for the time series type. If no code is defined, the column indexes will not be converted; RES-J will assume they are already in the standard forecasting units. For example "COLUMNVAR INPUT_TS IN" might be used in association with a snow water equivalent time series, specifying that your column indexes will be defined in "inches". RES-J will convert the table column index values from inches to millimeters (the FCST units for the SWE datatype). (See "I.4-DATAUNIT: Data Unit Codes" for acceptable unit types and codes and "I.4 DATATYPE-TS: Calibration System Historical Data and Operational

Keyword Definition

Forecast System Processed Data Base Data Type Codes" for FCST Units associated with a given time series data type.)

- A component state expression such as <ComponentID>.<ComponentState> (e.g.: RESA.STARTINGPOOL, NODEX.ENDINGINFLOW). Other ComponentStates are listed as below. A reservoir or node component state also available in the RULES section.

These include the following reservoir states:
PREVIOUSPOOL, STARTINGPOOL, ENDINGPOOL,
PREVIOUSINFLOW, STARTINGINFLOW, ENDINGINFLOW,
PREVIOUSRELEASE, STARTINGRELEASE, ENDINGRELEASE,
PREVIOUSWITHDRAW, STARTINGWITHDRAW, ENDINGWITHDRAW,
PREVIOUSAUGMENT(ATION), STARTINGAUGMENT(ATION),
ENDINGAUGMENT(ATION).

These also include the following node states:
PREVIOUSINFLOW, STARTINGINFLOW, ENDINGINFLOW,
PREVIOUSDISCHARGE, STARTINGDISCHARGE,
ENDINGDISCHARGE, PREVIOUSDIVERSION,
STARTINGDIVERSION, ENDINGDIVERSION,
PREVIOUSAUGMENT(ATION), STARTINGAUGMENT(ATION),
ENDINGAUGMENT(ATION).

Complex state expressions can also be used, as described in the RULES section documentation (e.g.: RESA.STARTINGPOOL - 35.0, NODEA.ENDINGINFLOW - NODEA.STARTINGINFLOW)

ROWVAR Indicates what variable is represented by the rows in the lookup table. 'ROWVAR' must be followed by one of the following:

S 'DATE' meaning the row indexes are to be indexed by date and accessed using the date/time of the current time step.

S 'INPUT_TS' meaning the rows are to be accessed using the current time step value in an input time series specified previously using the 'TSINDEX ROWS' keywords. Depending on the data type of the input time series, its value could be snow water equivalent, pool elevation, inflow, outflow, etc.

'INPUT_TS' may be followed by a unit type used in defining the row headings. RES-J will convert the row indexes from this unit type to the standard FCST units for the time series type. If no code is defined, the row indexes will not be converted; RES-J will assume they are already in the standard forecasting units. For example "ROWVAR INPUT_TS IN" might be used in association with a snow water equivalent time series, specifying that your row indexes will be defined in "inches". RES-J will

Keyword Definition

convert the table row index values from inches to millimeters (the FCST units for the SWE datatype). (See "I.4-DATAUNIT: Data Unit Codes" for acceptable unit types and codes and "I.4 DATATYPE-TS: Calibration System Historical Data and Operational Forecast System Processed Data Base Data Type Codes" for FCST Units associated with a given time series data type.)

S A component state expression such as <ComponentID>.<ComponentState> (e.g.: RESA.STARTINGPOOL, NODEX.ENDINGINFLOW). Other ComponentStates are listed as below. A reservoir or node component state also available in the RULES section.

These include the following reservoir states:
PREVIOUSPOOL, STARTINGPOOL, ENDINGPOOL,
PREVIOUSINFLOW, STARTINGINFLOW, ENDINGINFLOW,
PREVIOUSRELEASE, STARTINGRELEASE, ENDINGRELEASE,
PREVIOUSWITHDRAW, STARTINGWITHDRAW, ENDINGWITHDRAW,
PREVIOUSAUGMENT(ATION), STARTINGAUGMENT(ATION),
ENDINGAUGMENT(ATION).

These also include the following node states:
PREVIOUSINFLOW, STARTINGINFLOW, ENDINGINFLOW,
PREVIOUSDISCHARGE, STARTINGDISCHARGE,
ENDINGDISCHARGE, PREVIOUSDIVERSION,
STARTINGDIVERSION, ENDINGDIVERSION,
PREVIOUSAUGMENT(ATION), STARTINGAUGMENT(ATION),
ENDINGAUGMENT(ATION).

Complex state expressions can also be used, as described in the RULES section documentation (e.g.: RESA.STARTINGPOOL - 35.0, NODEA.ENDINGINFLOW - NODEA.STARTINGINFLOW)

TABLEVAR Indicates what variable is calculated by the lookup table or defined in the time series specified previously by the 'TSINPUT' keyword. The 'TABLEVAR' keyword and associated parameters must be defined after both the ROWVAR and COLUMNVAR parameters. 'TABLEVAR' must be followed by one of the following keywords:

S 'WITHDRAW(AL)' from a reservoir (positive values are removed from the reservoir; negative values are added to the reservoir).

S 'RELEASE' from a reservoir (all values must be positive).

S 'AUGMENTATION' to a reservoir or a node (positive values are added to the reservoir or node; negative values are removed from the reservoir or node).

S 'DIVERSION' from a node (positive values are removed from the node; negative values are added to the node).

Keyword Definition

The LOOKUP3 method as applied to a reservoir falls into one of two method-types: withdrawal type or release type. In this sense it can be combined with other methods (SetWithdraw, SetRelease, SetElevation, Balance, etc.) of the same type. The LOOKUP3 method-type is defined by the TABLEVAR sub-keyword.

If the sub-keyword is RELEASE, the method becomes a release-type method. As a release-type method its value may be applied to the owning reservoir's release variable or passed to a Combo method (SETSUM, SETMIN, SETMAX) containing other release-type methods.

If the sub-keyword is WITHDRAWAL or AUGMENTATION, the method becomes a withdrawal-type method. The ultimate value calculated by a withdrawal-type method may be assigned to the reservoir withdrawal variable or passed to a Combo method (SETSUM, SETMIN, SETMAX) containing other withdrawal-type methods. A preliminary value is calculated in the same fashion as all LOOKUP3 methods. How this preliminary value is finalized to become the withdrawal-type method's ultimate value depends on the TABLEVAR sub-keyword, however.

If the sub-keyword is WITHDRAWAL, the ultimate value equals the preliminary value (positive quantities exiting the reservoir, negative quantities entering the reservoir).

If the sub-keyword is AUGMENTATION, the sign of the preliminary value is reversed prior to its assignment as the ultimate value. Therefore, a calculated AUGMENTATION value with positive sign will be applied to the withdrawal variable as a negative quantity--both representing water entering the reservoir. Alternatively, a calculated AUGMENTATION value with negative sign will be applied to the withdrawal variable as a positive quantity--both representing water exiting the reservoir.

If applied to a node, the LOOKUP3 method is a diversion-type method, independent of the TABLEVAR sub-keyword. Similarly, however, the sub-keyword defines how the ultimate value is calculated from the preliminary value. If the sub-keyword is DIVERSION, the ultimate value equals the preliminary value (positive quantities exiting the node, negative quantities entering the node). If the sub-keyword is AUGMENTATION, the sign of the preliminary value is reversed prior to its assignment as the ultimate value. Therefore, a calculated AUGMENTATION value with positive sign will be applied to the diversion variable as a negative quantity--both representing water entering the node. Alternatively, a calculated AUGMENTATION value with negative sign will be applied to the diversion variable as a positive quantity--both representing water exiting the node.

Keyword Definition

A table containing both positive and negative values could be useful in a synthetic node, a node representing combined results from several diversion/augmentation nodes. A synthetic node could be useful to simplify the modeling efforts. INTERPOLATE will generally be a preferred option to use in a synthetic node.

VALUES Indicates the start of the lookup table.

On the next line after 'VALUES', begin a table by listing column index variables. Beginning on the line after the last of N column indices, define the remainder of the table row by row, with a row index followed by N values. The table may have any number of columns and rows. The meaning of the column indexes is defined by the COLUMNVAR keyword. The meaning of the row indexes is defined by the ROWVAR keyword. In the subsequent example, ROWVAR would be parameterized as DATE. COLVAR might be INPUT_TS or a component state.

Example:

```
VALUES
      10 20 30 40
01/01  5 10 15 20
      06/15 25 35 45 55
12/31  5 10 15 20
```

ENDVALUES Indicates the end of the lookup table.

[INTERPOLATE] Indicates that values between table-defined columns and / or rows will be linearly interpolated. If 'INTERPOLATE' is omitted, no interpolation will occur; the column (row) chosen will be determined as the last column (row) indexed by a value equal to or less than the current value of the indexing variable. 'INTERPOLATE' must be followed by one of the following keywords:

- 'COLUMNS' indicates interpolation only between columns.
- 'ROWS' indicates interpolation only between rows.
- 'ALL' or 'BOTH' indicates interpolation between both columns and rows

No extrapolation beyond the parameterized table will occur. If the independent value being used in the lookup table is less than the smallest column (row) index or greater than the largest column (row) index, the smallest or largest column (row) will be used, respectively, without extrapolation.

[WEEKLYVARIATION] Indicates that values taken from the lookup table will be scaled according to a weekly pattern. 'WEEKLYVARIATION' must be followed by:

Keyword Definition

- Number of time steps per day * 7 multiplier values.

The actual value used in the computation for a given day is the table value multiplied by the specified multiplier value for that day. These values are applied following any interpolation prior to any blending.

[BLENDTS] Causes blending from the observed time series. Blending occurs from the last non-missing value in the time series to a value in the lookup table, over a given number of time steps. 'BLENDTS' must be followed by:

- An integer number of time steps over which to blend

[BLENDTBL] Causes blending within the lookup table. Blending occurs from previously-table-specified value to a newly-table-specified value when an index is crossed from one time step to the next, over a given number of time steps. 'BLENDTBL' must be followed by:

- An integer number of time steps over which to blend

Table blending is intended to smooth the transition from one table-prescribed release to another. Therefore, use of 'INTERPOLATE' may nullify 'BLENDTBL' functionality. For example, if the column indexing value crosses over a column index from one time step to the next, blending would occur. However, if 'INTERPOLATE COLUMNS' or 'INTERPOLATE ALL' has been defined, blending would be nullified as the transition across the columns would have already been smoothed by interpolation.

[TOCOMP] Indicates that withdrawals/diversion from the owning component will be applied as inflows to other components. 'TOCOMP' must be followed by:

- A component ID that will receive the withdrawal or diversion.

- A transfer node keyword: 'INSTANTANEOUS' indicates that the withdrawal should be applied to the component this time step, to be used in that component's solution. RES-J verifies that the component receiving the transferred flow will be solved after the reservoir owning this LOOKUP3 method. If it will solve before the owning reservoir, 'INSTANTANEOUS' parameterization is not allowed. 'NEXTSTEP' indicates that the flow should be applied to the component for use in its solution at the next time step.

Use of the TOCOMP keyword, sending water TO a COMPONENT cannot be reversed by negative values that suggest taking water FROM a COMPONENT. Any simulation value implying that water is being transferred from the TOCOMP-specified component into the LOOKUP3 owning component will be replaced by a value of 0.

A negative value may also result during TOCOMP keyword usage at the end of a current time step when a LOOKUP3

Keyword Definition

method using TOCOMP is coupled with a LOOKUP3 AUGMENTATION type method within a SETSUM method. To avoid negative value difficulties, parameterize the SETSUM method with the LOOKUP3 AUGMENTATION type method before the LOOKUP3 method using the TOCOMP keyword. As the sum of the diversions and augmentations is partitioned, and some water is diverted, each method's value is considered within a running-total of flow context. If the running-total is negative (meaning an augmentation) when a LOOKUP3 TOCOMP type method is considered, its diversion cannot be handled properly. It will be revised to 0 and a "NOTE" (non-error, non-warning) message output.

[INITIALTRANSFER] Indicates the value transferred to the 'TOCOMP' component as a result of last time step's solution. If the TOCOMP mode is set to NEXTSTEP, this keyword MUST be included in the parameterization. The INITIALTRANSFER value is applied to the receiving component at the beginning of the first solution time step. 'INITIALTRANSFER' may be followed by:

- The initial transfer to the component. If not provided, initial transfer will default to 0. Any negative value also will be replaced with 0.

ENDLOOKUP3 Indicates the end of parametric information for the Method LOOKUP3.

Carryover Transfer

LOOKUP3 carryover values are BLENDTS, BLENDTBL, COLINDEX, ROWINDEX and LASTVALUE (INITIALTRANSFER) values. No transfer of the blend and indexing parameters is made. Transfer of LASTVALUE (INITIALTRANSFER) is direct, without transformation:

$$\text{LASTVALUE (NEW)} = \text{LASTVALUE (OLD)}$$

Examples

In the following examples, blue lettering highlights keywords.

1) The following example shows a LOOKUP3 implementation specifying release as a function of the owning component's pool elevation and total inflow.

LOOKUP3 RESA REL_EXAMPLE1

#No observed time series so skip the TSINPUT OBSERVEDRELEASE line.

#No time series as independent variables so skip TSINDEX COLUMNS

```

#and TSINDEX ROWS lines.
#Columns will be indexed by starting pool elevation at RESA.
COLUMNVAR RESA.STARTINGPOOL
#Rows will be indexed by ending inflow to RESA.
ROWVAR RESA.ENDINGINFLOW
#Output from the table will be release.
TABLEVAR RELEASE
VALUES
      200 210 220 230
    0   0   5  50  50
  100   0  10  50 100
  250   0  50 150 250
  800   0 100 450 800
ENDVALUES
#Interpolate for inflows (rows) but stair-step on columns.
INTERPOLATE ROWS
#If RESA.STARTINGPOOL is 210 and RESA.ENDINGINFLOW is 100, then RELEASE
#will be 10.
#If RESA.STARTINGPOOL is 215 and RESA.ENDINGINFLOW is 50, then we use
#the column indexed by 210 and interpolate the rows indexed by 0 and
#100, so the RELEASE will be 7.5.
#No concern for weekly variation, so skip WEEKLYVARIATION line.
#No input time series so skip the BLENDTS line.
#Blend over 4 time steps any movement across columns.
BLENDTBL 4
#Because the method applies to reservoir release, the TOCOMP line and the
#INITIALTRANSFER line do not apply. Skip them.

```

ENDLOOKUP3

2) The following example shows a LOOKUP3 implementation specifying release as a function of downstream discharge at a node and date.

```

LOOKUP3 RESA REL_EXAMPLE2
#No observed time series so skip the TSINPUT OBSERVEDRELEASE line.
#No time series as independent variables so skip TSINDEX COLUMNS
#and TSINDEX ROWS lines.
#Columns will be indexed by inflow to a node. In this fashion we
#will create an efficient, if not quite so precise, substitute for
#the MaxStage method.
COLUMNVAR DOWNSTR_NODE.ENDINGINFLOW
#Rows will be indexed by date to account for flood and non-flood

```

#season maximum discharges (900 in flood season, 600 in non-flood).

ROWVAR DATE

#Output from the table will be release.

TABLEVAR RELEASE

VALUES

	0	200	500	900
01/01	900	700	400	0
03/16	600	400	100	0
08/16	900	700	400	0
12/31	900	700	400	0

ENDVALUES

#Interpolate flows but stair-step dates.

INTERPOLATE COLUMNS

#If NODE.STARTINGDISCHARGE is 500 and current timestep date is
#03/16_00, then RELEASE will be 100.

#If NODE.STARTINGDISCHARGE is 700 and current timestep date is
#06/01_00, then we interpolate between the columns indexed by 500 and
#900 and use the row indexed by 03/16, so the release is 50.

#No need for weeklyvariation or blending.

#Because the method applies to reservoir release, the TOCOMP line
#and the INITIALTRANSFER line do not apply. Skip them.

ENDLOOKUP3

3) The following example shows a LOOKUP3 implementation that might work for a pumped storage project. When an observed time series has a value for water pumped from the reservoir to the upstream reservoir, use it as withdrawal. Otherwise, the withdrawal pumped back is calculated as a function of pool elevation at the pumped storage reservoir and the current time step value of some externally calculated time series.

LOOKUP3 RESA WITH_EX1

#The observed time series for withdrawal (pump back) to the generation
#reservoir.

TSINPUT OBSERVEDWITHDRAW OBS_WITHDR

#The independent variable used for indexing rows is a time series.

#Here we specify the time series and that its current time step value
#will index the rows in the table.

TSINDEX ROWS INDEP_TS

#Columns will be indexed by the starting pool elevation at a reservoir
#called PUMPEDRES.

COLUMNVAR PUMPEDRES.STARTINGPOOL

```

#Rows will be indexed by current time step values in the time series
#identified above with TSINDEX.
ROWVAR INPUT_TS
#Output from the table will be withdrawal.
TABLEVAR WITHDRAW
VALUES
    250 255 260 265
    0 30 20 10 0
    100 75 50 25 0
    1000 150 100 50 0
    10000 300 250 200 0
ENDVALUES
#Interpolate throughout the table.
INTERPOLATE ALL
#Scale the value according to the time step within the week
WEEKLYVARIATION 1.5 0.5 0.5 1.5 \ # actual withdrawal for day1, period1=
    1.0 0.0 0.0 1.0 \ # 1.5 * table-values
    1.0 0.0 0.0 1.0 \
    1.0 0.0 0.0 1.0 \ # actual withdrawal for day1, period2=
    1.0 0.0 0.0 1.0 \ # 0.5 * table-values
    1.0 0.0 0.0 1.0 \
    1.0 0.5 0.5 1.0
#No need for blending.
#Send the withdrawal to the reservoir called PUMPEDRES, and apply
#the flow as inflow next time step.
TOCOMP PUMPEDRES NEXTSTEP
#Initialize the TOCOMP transfer as 0.
INITIALTRANSFER 0.0
ENDLOOKUP3
4) The following example shows a diversion from a node. When an
observed time series has a value for diversion, use it. Otherwise,
divert from the node as a function of an externally calculated time
series (perhaps one representing soil moisture) and date.
LOOKUP3 NODEA DIV_EXAMPLE4
#The observed time series for diversion:
TSINPUT OBSERVEDDIVERSION OBS_DIVERS
#The independent variable used for indexing columns is a time series.
#Here we specify the time series and that its current time step value
#will index the columns in the table.

```

TSINDEX COLUMNS INDEP_TS
#Columns will be indexed by current time step values in the time series
#identified above with TSINDEX. Perhaps we would need to convert the
#column index units, but we don't specify any base units. Therefore, no
#conversion will occur.

COLUMNVAR INPUT_TS
#Rows will be indexed current time step date.

ROWVAR DATE
#Output from the table will be diversion.

TABLEVAR DIVERSION
VALUES

	0	200	800	1000
01/01	0	0	0	0
04/16	400	400	200	0
07/01	500	500	300	0
09/16	400	400	200	0
10/16	0	0	0	0
12/31	0	0	0	0

ENDVALUES
#Interpolate across columns only

INTERPOLATE COLUMNS
#If current time step value of the INDEP_TS time series is 800 and
#current timestep date is 07/01_00, then diversion will be 300.
#If current time step value of the INDEP_TS time series is 900 and
#current timestep date is 07/15_00, then we interpolate between the
#columns indexed by 800 and 1000 and use the row indexed by 07/01, so
#the diversion is 150.
#Scale the value according to the time step within the week.

WEEKLYVARIATION 1.0 0.5 0.5 1.0 \
1.0 1.0 1.0 1.0 \
1.0 1.0 1.0 1.0 \
1.0 1.0 1.0 1.0 \
1.0 1.0 1.0 1.0 \
1.0 1.0 1.0 1.0 \
1.0 1.0 1.0 1.0

#Blend both from observed values to the table and within the table.

BLENDTS 8
#BLENDTBL only has meaning across rows as we already interpolate

#between columns.

BLENDTBL 4

#Divert from the node, but not to any specific component. Therefore,
#the TOCOMP and INITIALTRANSFER lines do not apply. Skip them.

ENDLOOKUP3

5) The following example shows an augmentation of flow to a node as a function of an externally calculated time series (perhaps one representing consumptive demand) and date. The values calculated by this method are applied at the Node with positive quantities as increases to flow at the node (augmentation) and negative quantities as decreases (negative augmentation, or diversion) at the node. The negative augmentation quantities might physically represent a loss from the node either as a physical diversion at the node or a loss from the node through seepage.

LOOKUP3 NODEA DIV_EXAMPLE5

#The independent variable used for indexing columns is a time series.
#Here we specify the time series and that its current time step value
#will index the columns in the table.

TSINDEX COLUMNS INDEP_TS

#Columns will be indexed by current time step values in the time series
#identified above with TSINDEX.

COLUMNVAR INPUT_TS

#Rows will be indexed by date.

ROWVAR DATE

#Output from the table will be augmentation. Positive values will
#signify addition of water to the node. Negative values will signify
#removal of water from the node.

TABLEVAR AUGMENTATION

#Note that during the high irrigation period, we are augmenting
#using a negative value. Since augmentation represents water being
#added to the total at the node, a negative augmentation represents
#water being removed from the node.

VALUES

	0	200	800	1000
01/01	35	35	40	50
08/01	-20	-15	15	25
09/16	25	25	30	40
10/16	35	35	40	50
12/31	35	35	40	50

ENDVALUES

#Interpolate across columns and rows.

INTERPOLATE ALL

#No need to vary according to the time step within the week.
#Skip the WEEKLYVARIATION line.
#With INTERPOLATE ALL, no need to blend.
#Because this applies to a node augmentation, the TOCOMP and
#INITIALTRANSFER lines do not apply. Skip them.

ENDLOOKUP3

6) The following example is a re-parameterization of example 5 from an augmentation of flow to a node as a diversion of flow from a node. The purpose of this example is to highlight the meaning of positive and negative values in the table, depending on the type of application. The values calculated by this method are applied at the Node with positive quantities as decreases of flow at the node (diversion) and negative quantities as increases (negative diversion, or augmentation) at the node. The negative diversion quantities might physically represent return flows to the node. It is probably clear that this parameterization is somewhat confusing and less than ideal. Most of the table is negative with the concept of negative diversions being somewhat distasteful. The augmentation parameterization in example 5 is preferable for the functionality desired. Nevertheless, this parameterization is acceptable.

LOOKUP3 NODEA DIV_EXAMPLE6

#The independent variable used for indexing rows is a time series.
#Here we specify the time series and that its current time step value
#will index the columns in the table.

TSINDEX COLUMNS INDEP_TS

#Columns will be indexed by current time step values in the time series
#identified above with TSINDEX.

COLUMNVAR INPUT_TS

#Rows will be indexed by date.

ROWVAR DATE

#Output from the table will be diversion. Positive values will
#signify removal of water from the node. Negative values will signify
#addition of water to the node.

TABLEVAR DIVERSION

#Note that for periods except during the high irrigation period, we are
#diverting using negative values. Since diversion represents water
#being taken from the total at the node, a negative diversion
#represents water being added the node.

VALUES

0 200 800 1000
01/01 -35 -35 -40 -50

```
08/01  20  15 -15  -25
09/16 -25 -25 -30  -40
10/16 -35 -35 -40  -50
12/31 -35 -35 -40  -50
```

ENDVALUES

#Interpolate across columns and rows.

INTERPOLATE ALL

#No need to vary according to the time step within the week.

#Skip the WEEKLYVARIATION line.

#With INTERPOLATE ALL, no need to blend.

#Because this applies to a node diversion, TOCOMP functionality

#might be considered. However, because we have negative values in the
#table, it is not allowed.

#With no TOCOMP functionality, the INITIALTRANSFER line does not apply.

#Skip it.

ENDLOOKUP3

7) While negative augmentation and negative diversion values in a LOOKUP3 table may be somewhat confusing, functionality exists in RES-J that would allow the modeler to parameterize two independent LOOKUP3 methods with strictly positive values. One would be an augmentation LOOKUP3 and the other would be a diversion LOOKUP3. Execution of the two might be mutually exclusive as a result of conditional expressions in the RULES section. Alternatively, they might be combined in a SETSUM method.

The following example attempts to provide reasoning why the ability to parameterize both positive and negative values into one table, beyond the efficiency of one method versus two. The example presupposes the need to have water entering and exiting the node under different circumstances. Discussion follows the example.

LOOKUP3 NODEA DIV_EXAMPLE

#The independent variable used for indexing columns is a time series.

#This one might represent some availability of water for import.

#Here we specify the time series and that its current time step value

#will index the columns in the table.

TSINDEX COLUMNS INDEP_TS1

#The independent variable used for indexing rows is a time series.

#This one might represent how wet the conditions are wherever diverted

#water might be applied.

#Here we specify the time series and that its current time step value

#will index the columns in the table.

TSINDEX ROWS INDEP_TS2

#Columns will be indexed by current time step values in the time series

#identified above with TSINDEX COLUMNS.

COLUMNVAR INPUT_TS

#Columns will be indexed by current time step values in the time series

#identified above with TSINDEX ROWS.

ROWVAR INPUT_TS

#Output from the table will be diversion. Positive values will

#signify removal of water from the node. Negative values will signify

#addition of water to the node.

TABLEVAR DIVERSION

VALUES

	0	100	200	300	400	500	600	700	800	900	1000	
0	90	90	90	90	90	90	0	-20	-40	-60	-80	-100
10	90	90	90	90	90	90	90	-10	-30	-50	-70	-90
20	90	90	90	90	90	90	90	0	-20	-40	-60	-80
30	70	70	70	70	70	70	70	70	-10	-30	-50	-70
40	70	70	70	70	70	70	70	70	0	-20	-40	-60
50	55	55	55	55	55	55	55	55	55	-10	-30	-50
60	-10	40	40	40	40	40	40	40	40	0	-20	-40
70	-20	-10	40	40	40	40	40	40	40	40	-10	-30
80	-30	-20	-10	10	10	10	10	10	10	10	0	-20
90	-40	-30	-20	-10	10	10	10	10	10	10	10	-10
100	-50	-40	-30	-20	-10	10	10	10	10	10	10	0

ENDVALUES

#Do not interpolate, so skip it.

#No need to vary according to the time step within the week, so skip the

#WEEKLYVARIATION line.

#Skip blending.

#Because this applies to a node diversion, TOCOMP functionality

#might be considered. However, because we have negative values in the

#table any negative value will be replaced with 0 when transfer actually

#occurs.

#With no TOCOMP functionality, the INITIALTRANSFER line does not apply.

#Skip it.

ENDLOOKUP3

In the preceding example, the table is indexed entirely by time series values calculated external to RES-J.

Suppose the user wished to parameterize this as two LOOKUP3 methods.

First, the user would create the two methods. One would be a

diversion method, identical to this one but replacing all the negative

values with zeros. The second would be an augmentation method where

zeros replaced all positive values in this original diversion method and all negative values then were replaced with positive values. To arrive at the desired result, these methods would be executed in a Combo method.

Within a SETSUM combo method, each LOOKUP method produces results that act on the NODE diversion variable. They both execute any time the SETSUM method is active. The SETSUM adds the two values together, treating the diversions as positive quantities and augmentations as negative quantities. Independent of whether the sum is positive or negative, the sum would be placed on the Node diversion variable.

Suppose the example had been different. Suppose the table was simpler and its indexing was not time series based, but available RES-J component states; states that could be queried in RES-J RULES like pool elevation and date. The user could then choose not to couple the LOOKUP3 methods within a Combo method, but have them execute exclusive of each other as defined by RULES conditional expressions.

Description

Method MAXDECREASE computes the reservoir release that is a specified decrease in the flow over the previous release.

Input

<u>Keyword</u>	<u>Definition</u>
MAXDECREASE	Keyword indicating the start of parameters for Method MAXDECREASE. Must be followed by: <ul style="list-style-type: none">- component identifier- Method identifier
DECREASE	Maximum decrease in flow/time step.
ENDMAXDECREASE	Keyword indicating the end of parameters for Method MAXDECREASE.

Example

```
MAXDECREASE ReservoirName MethodName
DECREASE 10.2
ENDMAXDECREASE
```

V.3.3-RES-J-MAXINCREASE JOINT RESERVOIR REGULATION OPERATION METHOD
MAXINCREASE

Description

Method MAXINCREASE computes the reservoir release that is a specified increase in the flow over the previous release.

Input

<u>Keyword</u>	<u>Definition</u>
MAXINCREASE	Keyword indicating the start of parameters for Method MAXINCREASE. Must be followed by: <ul style="list-style-type: none">- component identifier- Method identifier
INCREASE	Maximum increase in flow/time step.
ENDMAXINCREASE	Keyword indicating the end of parameters for Method MAXINCREASE.

Example

```
MAXINCREASE ReservoirName MethodName
  INCREASE 10.2
ENDMAXINCREASE
```

Keyword Definition

[Back] [Next] [Previous] [Bookmarks] [Top]

V.3.3-RES-J-MAXSTAGE JOINT RESERVOIR REGULATION OPERATION METHOD
MAXSTAGE

Description

Method MAXSTAGE iteratively solves for the release of a reservoir until the downstream stage (corresponding to the routed release plus any additional inflows downstream of the reservoir) at a control point (node) is within a specified tolerance of a specified maximum value.

In essence, MAXSTAGE suspends simulation of the RES-J system, copies the affected portion of the system and runs the copy from the current time step forward sufficient time for the prescribed release to reach the control point. Simulation of the copy RES-J system is according to the same rules and Methods as for the original system. The prescribed release is adjusted and simulation repeats until an acceptable release is determined or the maximum number of iterations is reached.

Input

Keyword Definition

MAXSTAGE Indicates the start of parametric information for Method MAXSTAGE. 'MAXSTAGE' must be followed by:

- a reservoir-type component identifier as defined in the TOPOLOGY section
- an identifier for this Method

[TABLE] Indicates the start the stage/discharge rating curve table. 'TABLE' must be followed by:

- table keyword 'RATING_CURVE'

Beginning on the next line after 'RATING_CURVE', create a table of stage elevation / discharge pairs describing the channel hydraulic conditions at the node.

<u>Keyword</u>	<u>Definition</u>
	A rating curve TABLE cannot be specified if the DSCONTROL node has been parameterized with a rating curve, either using the RATINGCURVEID or TABLE keyword. Note: this section's inputs are retained only for backward compatibility.
[ENDTABLE]	Indicates the end of the rating curve table.
[MAXIMUMDISCHARGE]	
Or [MAXIMUMSTAGE]	Maximum discharge (or maximum stage) at the downstream control point. Either (but not both of) MAXIMUMDISCHARGE (or MAXIMUMSTAGE) must be specified. If MAXIMUMDISCHARGE is specified, no Rating Table or Rating Curve ID (at the NODE) is required. If a Rating Table or Rating Curve ID is specified at either the node or in the method they will be ignored. Use of MAXIMUMDISCHARGE option is encouraged. It does not require table look up during iterations and will reduce process time. Units are as described in the previous Parameter Definition section.
MINRELEASE	Minimum allowed reservoir release. This will likely be equal to MINRELEASE as defined in the reservoir parameter definition section, but may be different.
[CRITERION]	Convergence criterion. The value is used as an absolute allowable difference in simulated stage and MAXIMUMSTAGE where the value compared to CRITERION is calculated according to the equation: $D = \text{abs}((\text{MAXIMUMSTAGE} - \text{SIM_STAGE}) / \text{MAXIMUMSTAGE}) * 100$
DSCONTROL	Indicates the downstream control point. 'DSCONTROL' must be followed by: - a node-type component identifier as defined in the TOPOLOGY section
[MAXITERATIONS]	Maximum number of iterations. If not user-defined, MAXITERATIONS defaults to 20.
ENDMAXSTAGE	Indicates the end of MAXSTAGE parametric information.

Keyword

Definition

Error Conditions

Error messages are generated for the following conditions:

1. Both MAXIMUMSTAGE and MAXIMUMDISCHARGE are specified.
2. Neither MAXIMUMSTAGE or MAXIMUMDISCHARGE are specified.
3. The MAXSTAGE method defines a Rating Table, and a Rating Table and/or Rating Curve ID are specified at the NODE as well.
4. The MAXIMUMSTAGE constraint is used, but no Rating Table or Rating Curve ID is specified in the method or at the NODE.

Example

```
MAXIMUMSTAGE ReservoirName MethodName
TABLE rating_curve                #This is the rating curve at the
  0  0                             #downstream control point.
  2 100
  4 150
  6 200
  8 280
 10 350
ENDTABLE
MAXSTAGE      7.5                  #Stage at downstream control point
                                         #not to be exceeded.
MINRELEASE    100                  #Reservoir's minimum release.
CRITERION     2.5                  #Convergence criterion
#(percentage: absolute difference
                                         #in simulated and maximum
                                         #allowable stage divided by the
                                         #maximum allowable stage)
DSCONTROL     NodeName            #downstream control point
MAXITERATIONS 35                  #Optional-defaults to 20
ENDMAXSTAGE
```

V.3.3-RES-J-PASSFLOW JOINT RESERVOIR REGULATION OPERATION METHOD
PASSFLOW

Description

Method PASSFLOW allows the user to pass the reservoir inflow directly as the reservoir discharge. If there is no direct rainfall and evaporation and no withdrawal then the pool will be maintained at the same elevation.

Input

<u>Keyword</u>	<u>Definition</u>
PASSFLOW	Keyword indicating the start of parameters for Method PASSFLOW. Must be followed by: <ul style="list-style-type: none">- component identifier- Method identifier
ENDPASSFLOW	Keyword indicating the end of parameters for Method PASSFLOW.

Example

```
PASSFLOW ReservoirName MethodName  
ENDPASSFLOW
```

Keyword Definition

[Back] [Next] [Previous] [Bookmarks] [Top]

V.3.3-RES-J-RAINEVAP JOINT RESERVOIR REGULATION OPERATION METHOD
RAINEVAP

Description

Method RAINEVAP computes the change in reservoir storage due to precipitation and evaporation over the lake surface.

Input

Keyword Definition

RAINEVAP Keyword indicating the start of parameters for Method RAINEVAP. Must be followed by:

- component identifier
- Method identifier

[TSINPUT] Keyword indicating an input time series. Must be followed by:

- 'OBSERVEDPRECIP' or 'OBSERVEDEVAP'
- time series alias

The values in the timeseries are taken to be the total over the time step. Also note that if you provide an 'OBSERVEDEVAP' timeseries then it is recommended that you input a EVAP - ENDEVAP section (see below). Failure to do so will result in use of a default value of 0 whenever the observed timeseries value is missing.

[EVAP] Keyword indicating that average evaporation values will be specified.

[VALUES] Keyword indicating the start of a list of average daily total evaporation values. Must be followed by a list of date/evaporation pairs. A

<u>Keyword</u>	<u>Definition</u>
	date/evaporation pair may be followed by a diurnal distribution. The table will be interpolated in time to determine the value for application for a given timestep. <u>1/</u>
[ENDVALUES]	Keyword indicating the end of the list.
[ENDEVAP]	Keyword indicating the end of the EVAP input.
ENDRAINEVAP	Keyword indicating the end of the RAINEVAP parameters.

1/ Diurnal distributions are specified using 24/dt values that sum to 1.0.

Example

```

RAINEVAP ReservoirName MethodName
    TSINPUT ObservedPrecip alias    #Observed MAP time series

    TSINPUT ObservedEvap alias      #Observed evaporation time
    #series

    EVAP                               #Beginning of Evaporation data
VALUES                                #evaporation values
    01/01    .25 0.3 0.3 0.25 0.15
    02/01    .08
    03/01    .09
    10/13    0.25
    10/28    0.23
    11/15    0.21
    ENDVALUES

    ENDEVAP                               #End of evaporation
ENDRAINEVAP

```

Keyword Definition

[Back] [Next] [Previous] [Bookmarks] [Top]

V.3.3-RES-J-SETELEVATION JOINT RESERVOIR REGULATION OPERATION METHOD
SETELEVATION

Description

Method SETELEVATION prescribes a reservoir release calculated to achieve a target reservoir pool elevation.

The target elevation is based on an observed value or according to user-specified date/elevations pairs; the pairs are only used when observed values are not available. The parametric syntax for SETELEVATION is described below.

See Section II.4-RES-J-SETELEVATION [[Hyperlink](#)] for a description of how the date/pool elevation pairs are used and the decision algorithm for blending.

Input

Keyword Definition

SETELEVATION Indicates the start of parametric information for Method SETELEVATION. 'SETELEVATION' must be followed by:

- a reservoir-type component identifier as defined in the TOPOLOGY section
- an identifier for this Method

[TSINPUT] Indicates the existence of an observed pool elevation time series. 'TSINPUT' must be followed by:

- time series keyword 'OBSERVEDPOOL'
- an input time series alias defined in the TIMESERIES definition section

<u>Keyword</u>	<u>Definition</u>
VALUES	<p>Indicates the start of the date/elevation pair list.</p> <p>Beginning on the next line after 'VALUES', list any number of date/elevation pairs, one pair per line. Each date should be unique. It is recommended that the list of pairs begin with 1/1_00:00 (midnight on the morning of January 1) and end with 12/31_23:00 (11:00 p.m. on December 31). Doing so will facilitate lookup procedures on the list and aid the user in ensuring that the transition over year's end will function as desired. <u>1/ 2/</u></p>
ENDVALUES	Indicates the end of the list
[INTERPOLATE]	Indicates that elevations between list-defined dates will be linearly interpolated.
[BLENDTS]	<p>Causes blending to occur from the last non-missing value in the observed pool elevation time series to a value in the date/elevation pair list, over a given number of time steps. 'BLENDTS' must be followed by:</p> <ul style="list-style-type: none"> - an integer number of time steps over which to blend. <u>2/</u> - an optional integer number representing the time step number since the current time series blending sequence began. This also corresponds to the time step number since the last non-missing value in the release time series. Defaults to the time series blend value + 1 (representing a complete blend). This value is used to initialize carryover.
[BLENDTBL]	<p>Causes blending of the elevation table. Blending occurs across an elevation explicitly defined on the date/elevation pair list, over a given number of time steps. 'BLENDTBL' must be followed by:</p> <ul style="list-style-type: none"> - an integer number of time steps over which to blend. <u>2/</u> - an optional integer number representing the time step number since the current list blending sequence began. Defaults to 1. This value is used to initialize carryover. <p>List blending is intended to smooth the transition from one list-prescribed release to another.</p>

Keyword

Definition

Therefore, use of 'INTERPOLATE' will nullify 'BLENDTBL' functionality. For example, if the current time step and the previous time step straddle a list-defined date, blending from the previous elevation to the new elevation would occur. If however, 'INTERPOLATE' has been defined, blending would be nullified as the transition across the date would have already been smoothed by interpolation.

ENDSETELEVATION Indicates the end of parameters for Method SETELEVATION.

1/ Dates are specified using one of the formats:

'2-digit_month'/'2-digit_day'

'2-digit_month'/'2-digit_day'_'2-digit_hour'

'2-digit_month'/'2-digit_day'_'2-digit_hour':'2-digit_minute'

2/ See Section II.4-RES-J-SETELEVATION [[Hyperlink](#)] for a description of how the date/elevation pair list is used and the decision algorithm for blending.

Example

SETELEVATION ReservoirName MethodName

TSINPUT ObservedPool alias #Observed time series -
#optional.

VALUES #If observed time series
#specified, these values
#are only used
#if time series value is
#missing. If no observed
#time series is entered,
#then these values are
#used exclusively.

01/01_00:00 135.0

01/01_06:00 136.0

01/01_12:00 134.0

Keyword Definition

01/01_18:00 135.0
01/02_00:00 136.0
01/02_06:00 134.0
01/02_12:00 134.0
01/02_18:00 135.0
01/03_00:00 136.0
02/01 133.5
10/13 124
10/28 125
11/15 120.3

ENDVALUES

INTERPOLATE

#Optional

BLENDTS 3

#BlendTS and Blend values are

BLENDTBL 2

#optional for this method. Note
#also that BLENDTBL value of 2
#would attempt to arrive at a
#new value in 2*6 hours which is
#more than the 6 hour period for
#which we have defined some
#elevations. In these cases the
#blend algorithm would not
#finish prior to beginning
#another blend routine.

ENDSETELEVATION

[Back] [Next] [Previous] [Bookmarks] [Top]

V.3.3-RES-J-SETMAX JOINT RESERVOIR REGULATION OPERATION METHOD SETMAX

Description

Method SETMAX determines the maximum computed output variable of all of the Methods it acts upon.

All of the Methods specified must be of the same type, e.g. release or withdrawal.

Input

<u>Keyword</u>	<u>Definition</u>
SETMAX	Keyword indicating the start of parametric information for Method SETMAX. Must be followed by: <ul style="list-style-type: none">- component identifier- Method identifier- list of Method keywords, component identifiers, Method identifiers <u>1/</u>
ENDSETMAX	Keyword indicating the end of the parametric information for Method SETMAX.

1/ Valid Method keywords are:

- BALANCE
- MAXDECREASE
- MAXINCREASE
- MAXSTAGE
- SETELEVATION
- SETMAX
- SETMIN
- SETRELEASE

- SETSUM
- SETWITHDRAW

Example

```
SETMAX ComponentName MethodName
    #Will determine the maximum release value as computed by these
    #two methods:
    SETRELEASE ComponentName SetReleaseName
    SETELEVATION ComponentName SetElevationName
    ENDSETMAX
```

[Back] [Next] [Previous] [Bookmarks] [Top]

V.3.3-RES-J-SETMIN JOINT RESERVOIR REGULATION OPERATION METHOD SETMIN

Description

Method SETMIN determines the minimum computed output variable of all of the Methods it acts upon.

All of the Methods specified must be of the same type, e.g. release or withdrawal.

Input

<u>Keyword</u>	<u>Definition</u>
SETMIN	Keyword indicating the start of parametric information for Method SETMIN. Must be followed by: <ul style="list-style-type: none">- component identifier- Method identifier- list of Method keywords, component identifiers, Method identifiers <u>1/</u>
ENDSETMIN	Keyword indicating the end of the parametric information for Method SETMIN.

1/ Valid Method keywords are:

- BALANCE
- MAXDECREASE
- MAXINCREASE
- MAXSTAGE
- SETELEVATION
- SETMAX
- SETMIN
- SETRELEASE

- SETSUM
- SETWITHDRAW

Example

```
SETMIN ComponentName MethodName
    #Will determine the minimum release value as computed by these
    #two methods:
    SETRELEASE ComponentName SetReleaseName
    SETELEVATION ComponentName SetElevationName
    ENDSETMIN
```

Keyword

Definition

[Back] [Next] [Previous] [Bookmarks] [Top]

V.3.3-RES-J-SETRELEASE JOINT RESERVOIR REGULATION OPERATION METHOD
SETRELEASE

Description

Method SETRELEASE determines a reservoir release based on an observed value or a release table.

The release table allows releases to be expressed as a function of date and pool elevation. The release table is only used when observed values are not available. The parametric syntax for SETRELEASE is described below.

See Section II.4-RES-J-SETRELEASE [[Hyperlink](#)] For a description of how the release table is used and the decision algorithm for blending.

Input

Keyword

Definition

SETRELEASE	Indicates the start of parametric information for Method SETRELEASE. 'SETRELEASE' must be followed by: <ul style="list-style-type: none">- a reservoir-type component identifier as defined in the TOPOLOGY definition section- an identifier for this Method
[TSINPUT]	Indicates the existence of an observed release time series. 'TSINPUT' must be followed by: <ul style="list-style-type: none">- time series keyword 'OBSERVEDRELEASE'- an input time series alias defined in the TIMESERIES definition section
VALUES	Indicates the start of the release table.

<u>Keyword</u>	<u>Definition</u>
ELEV	<p>Indicates the start of the column header row defining pool elevations for which releases will be specified. 'ELEV' must be followed by:</p> <ul style="list-style-type: none"> - a list of elevation values in ASCENDING order - companion keyword 'ENDELEV', signifying the end of the column header row. <p>RES-J must see all elevation data on the same line as the keyword. If many data points are used and more than one line is desired to input this header row, force a line continuation by ending the input line with ' \' (a blank space followed by a backslash).</p> <p>Beginning on the next line after 'ENDELEV', create a table by listing a date, release values (as many releases as defined elevations) and an optional diurnal distribution. You may have any number of rows in this table, each beginning with a unique date. It is recommended, however, that the table begin with 1/1_00:00 (midnight on the morning of January 1) and end with 12/31_23:00 (11:00 p.m. on December 31). Doing so will facilitate lookup procedures on the table and aid the user in ensuring that the transition over year's end will function as desired. <u>1/</u> <u>2/</u> <u>3/</u></p>
ENDVALUES	Indicates the end of the release table.
[INTERPOLATE]	<p>Indicates that releases between table-defined dates and/or elevations will be linearly interpolated. 'INTERPOLATE' can be followed by the following keywords:</p> <ul style="list-style-type: none"> - 'TIME' indicates interpolation only between table dates. Constant release will be used between elevations. - 'ELEV' indicates interpolation only between table elevations - 'ALL' or 'BOTH' indicates interpolation between both table dates and elevations <p>If INTERPOLATE is omitted or entered with no keyword the default is 'INTERPOLATE TIME'.</p>
[WEEKLYVARIATION]	Indicates that values taken from the SETRELEASE table will be scaled according to a weekly pattern. 'WEEKLYVARIATION' must be followed by:

Keyword

Definition

- Number of timesteps per day * 7 scaling values.

These values are applied following any interpolation and (ill-advised) diurnal variation, but prior to any blending. These values are scaling values, not distribution values that must sum to 1.0 (as in the case of diurnal distributions).

[BLENDTS]

Causes blending of observed time series. Blending occurs from the last non-missing value in the release time series to a value in the release table, over a given number of time steps.

'BLENDTS' must be followed by:

- an integer number of time steps over which to blend 3/
- an optional integer number representing the time step number since the current time series blending sequence began. This also corresponds to the time step number since the last non-missing value in the release time series. Defaults to the time series blend value + 1 (representing a complete blend). This value is used to initialize carryover.

[BLENDTBL]

Causes blending on the release table. Blending occurs across a release explicitly defined on the release table, over a given number of time steps.

'BLENDTBL' must be followed by:

- an integer number of time steps over which to blend 3/
- an optional integer number representing the time step number since the current table blending sequence began. Defaults to 1. This value is used to initialize carryover.

Table blending is intended to smooth the transition from one table-prescribed release to another. Therefore, use of 'INTERPOLATE' may nullify 'BLENDTBL' functionality. For example, if pool elevation crosses over a table-defined elevation from one time step to the next, blending would occur. If however, 'INTERPOLATE ELEV' or 'INTERPOLATE ALL' has been defined, blending would be nullified as the transition across the elevation point would have already been smoothed by interpolation.

<u>Keyword</u>	<u>Definition</u>
ENDSETRELEASE	Indicates the end of parameters for Method SETRELEASE.

- 1/ Dates are specified using one of the following the formats:
'2-digit_month'/'2-digit_day'
'2-digit_month'/'2-digit_day'_'2-digit_hour'
'2-digit_month'/'2-digit_day'_'2-digit_hour':_'2-digit_minute'

- 2/ Date release values may be followed with a diurnal distribution. The diurnal distribution is expressed as a list of 24/dt values which sum to 1.0.

- 3/ See Section II.4-RES-J-SETRELEASE [[Hyperlink](#)] for a description of how the release table is used and the decision algorithm for blending.

Example

```

SETRELEASE ReservoirName MethodName
  TSINPUT ObservedRelease alias #Observed time series - optional
    VALUES                      #If observed time series
                                #specified, these values are
                                #only used if time series value
                                #is missing. If no observed
                                #time series, then these values
                                #are used exclusively.

    #Example of setting multiple release values (one for each
    #elevation value) with the ability to specify a
    #distribution as applied to that date for all three
    #values.
    ELEV  129.0  145.0  165.0  ENDELEV
    01/01  200  265  384  0.2  0.3  0.3  0.2
    02/01  300  320  340
    03/01  367  380  500
    08/01  235  270  382
    11/15  300  325  550
  ENDVALUES

```

<u>Keyword</u>	<u>Definition</u>
INTERPOLATE TIME	#Refers to how table values data are #interpreted.
BLENDTS 3 BLENDTBL 2	#Both BLENDTBL and BLENDTS are #optional
ENDSETRELEASE	

Keyword

Definition

[Back] [Next] [Previous] [Bookmarks] [Top]

V.3.3-RES-J-SETSUM JOINT RESERVOIR REGULATION OPERATION METHOD SETSUM

Description

Method SETSUM sums the computed output variable of all of the Methods it acts upon.

All of the Methods specified must be of the same type, e.g. release or withdrawal.

Input

Keyword

Definition

SETSUM

Keyword indicating the start of parametric information for Method SETSUM. Must be followed by:

- component identifier
- Method identifier
- list of Method keywords, component identifiers, Method identifiers 1/

ENDSETSUM

Keyword indicating the end of the parametric information for Method SETSUM.

1/ Valid Method keywords are:

- SETMAX
- SETMIN
- SETRELEASE
- SETWITHDRAW

Example

Keyword

Definition

SETSUM ComponentName MethodName

#Will determine the summation of the release values

#computed by these two methods:

SETRELEASE ComponentName SetreleaseName_1

SETRELEASE ComponentName SetreleaseName_2

ENDSETSUM

Keyword

Definition

[Back] [Next] [Previous] [Bookmarks] [Top]

V.3.3-RES-J-SETWITHDRAW JOINT RESERVOIR REGULATION OPERATION METHOD
SETWITHDRAW

Description

Method SETWITHDRAW determines a reservoir withdrawal based on an observed value or a withdrawal table.

The withdrawal table allows withdrawals to be expressed as a function of date and pool elevation. The withdrawal table is only used when observed values are not available. Optionally, the withdrawal can be applied as inflow to another component in the system. The parametric syntax for SETWITHDRAW is described below.

See Section II.4-RES-J-SETWITHDRAW [[Hyperlink](#)] for a description of how the withdrawal table is used and the decision algorithm for blending.

Input

Keyword

Definition

SETWITHDRAW

Indicates the start of parametric information for Method SETWITHDRAW. 'SETWITHDRAW' must be followed by:

- a reservoir-type component identifier as defined in the TOPOLOGY definition section.
- an identifier for this Method

[TSINPUT]

Indicates the existence of an observed release time series. 'TSINPUT' must be followed by:

- time series keyword 'OBSERVEDWITHDRAW'
- an input time series alias defined in the TIMESERIES definition section

VALUES

Indicates the start of the withdrawal table.

Keyword

Definition

ELEV

Indicates the start of the column header row defining pool elevations for which withdrawals will be specified. 'ELEV' must be followed by:

- a list of elevation values in ASCENDING order
- companion keyword 'ENDELEV', signifying the end of the column header row.

RES-J must see all elevation data on the same line as the keyword. If many data points are used and more than one line is desired to input this header row, force a line continuation by ending the input line with ' \' (a blank space followed by a backslash).

Beginning on the next line after 'ENDELEV', create a table by listing a date, withdrawal values (as many as withdrawals as defined elevations) and an optional diurnal distribution. You may have any number of rows in this table, each beginning with a unique date. It is recommended, however, that the table begin with 1/1_00:00 (midnight on the morning of January 1) and end with 12/31_23:00 (11:00 p.m. on December 31). Doing so will facilitate lookup procedures on the table and aid the user in ensuring that the transition over year's end will function as desired. 1/ 2/ 3/

ENDVALUES

Indicates the end of the withdrawal table.

[INTERPOLATE]

Indicates that withdrawals between table-defined dates and/or elevations will be interpolated.

'INTERPOLATE' can be followed by:

- an interpolation keyword: 'TIME' indicates interpolation only between table dates; 'ELEV' indicates interpolation only between table elevations; 'ALL' or 'BOTH' indicates interpolation between both table dates and elevations.

The default is 'TIME'.

[BLENDTS]

Causes blending to occur from the last non-missing value in the withdrawal time series to a value in the withdrawal table, over a given number of time steps. 'BLENDTS' must be followed by:

Keyword

Definition

- an integer number of time steps over which to blend 3/
- an optional integer number representing the time step number since the current time series blending sequence began. This also corresponds to the time step number since the last non-missing value in the release time series. Defaults to the time series blend value + 1 (representing a complete blend). This value is used to initialize carryover.

[BLENDTBL]

Causes blending to occur across a withdrawal explicitly defined in the withdrawal table, over a given number of time steps. 'BLENDTBL' must be followed by:

- an integer number of time steps over which to blend. 3/
- an optional integer number representing the time step number since the current table blending sequence began. Defaults to 1. This value is used to initialize carryover.

Table blending is intended to smooth the transition from one table-prescribed withdrawal to another. Therefore, use of 'INTERPOLATE' may nullify 'BLENDTBL' functionality. For example, if pool elevation crosses over a table-defined elevation from one time step to the next, blending would occur. If however, 'INTERPOLATE ELEV' or 'INTERPOLATE ALL' has been defined, blending would be nullified as the transition across the elevation point would have already been smoothed by interpolation.

[TOCOMP]

Indicates that withdrawals will be applied as inflows to another component. 'TOCOMP' must be followed by:

- a component ID that will receive the withdrawal. 4/
- a transfer mode keyword: 'INSTANTANEOUS' indicates that the withdrawal should be applied to the component this timestep, to be used in that component's solution. RES-J verifies that the component receiving the transferred withdrawal will be solved after the reservoir owning this SetWithdraw Method. If it will solve before the owning reservoir, 'INSTANTANEOUS' parameterization is not allowed. 'NEXTSTEP' indicates that

Keyword

Definition

the withdrawal should be applied to the component for use in its solution at the next timestep.

[INITIALTRANSFER] Indicates the value of the withdrawal transferred as a result of last timestep's solution. If the TOCOMP mode is set to NEXTSTEP, this keyword MUST be included in the parameterization. The INITIALTRANSFER value is applied to the receiving component at the beginning of the first solution timestep. 'INITIALTRANSFER' may be followed by:

- the initial transfer to the component. Defaults to 0.

ENDSETWITHDRAW Indicates the end of parameters for Method SETWITHDRAW.

Carryover Transfer

Carryover values saved are BLENDTS, BLENDTBL and INITIALTRANSFER values. No transfer of the blend parameters is made. Transfer of INITIALTRANSFER is direct, without transformation:

$$\text{INITIALTRANSFER(NEW)} = \text{INITIALTRANSFER(OLD)}$$

- 1/ Dates are specified using one of the following formats:
'2-digit_month'/'2-digit_day'
'2-digit_month'/'2-digit_day'_'2-digit_hour'
'2-digit_month'/'2-digit_day'_'2-digit_hour':_'2-digit_minute'
- 2/ Date withdrawal values may be followed with a diurnal distribution. The diurnal distribution is expressed as a list of 24/dt values which sum to 1.0.
- 3/ See Section II.4-RES-J-SETWITHDRAW [\[Hyperlink\]](#) for a description of how the withdrawal table is used and the decision algorithm for blending.
- 4/ The locations of the owner of this SETWITHDRAW Method and the receiving component must be consistent with the transfer mode in

Keyword

Definition

terms of the order of system solution. That is, the withdrawal solved by reservoir A cannot transfer to any component that has previously solved its Methods for the current timestep. Also, the user must ensure that if the receiving component belongs to a MAXSTAGE subtree, the owner of this SETWITHDRAW Method also belongs to the subtree. If the receiving component expects to receive a transfer from a reservoir that does not exist in the subtree, the MAXSTAGE Method will not operate correctly.

Example

```

SETWITHDRAW ReservoirName MethodName
  TSINPUT ObservedWithdraw alias
                                     #Observed time series - optional
VALUES                               #If observed time series
                                     #specified, these values are
                                     #only used if time series value
                                     #is missing. If no observed time
                                     #series, then these values used
                                     #exclusively.
#Example of setting multiple withdrawal values (one for
#each elevation value) with the ability to specify a
#distribution as applied to that date for all three
#values.
ELEV 129.0 145.0 165.0 ENDELEV
01/01 200 265 384 0.2 0.3 0.3 0.2
02/01 300 320 340
03/01 367 380 500
08/01 235 270 382
11/15 300 325 550
ENDVALUES
INTERPOLATE TIME #Refers to how table values are
                 #interpreted.
BLENDTBL 2 #Both BLENDTBL and BLENDTS are
           #optional
BLENDTS 3
TOCOMP NODE_A NEXTSTEP #The withdrawal will become inflow to

```

#NODE_A, next timestep. NODE_A may be
#anywhere within the system, so long
#as any MAXSTAGE method containing
#NODE_A also contains the owner of
#this method.

INITIALTRANSFER 45.0
ENDWITHDRAW

<u>Keyword</u>	<u>Definition</u>
	- an identifier for this Method
TABLE	Keyword indicating start of the elevation-storage table. 'TABLE' must be followed by: <ul style="list-style-type: none"> - table keyword 'ELEV_SPILL' signifying the table's use as an elevation-spill table on the same line <p>Subsequent lines must contain the table data:</p> <ul style="list-style-type: none"> - One elevation and its corresponding spill value pair per line
ENDTABLE	Indicates the end of the elevation-spill table.
INTERVALS	Indicates the number of intervals the simulation time step should be sub-divided into to enable pseudo-implicit solution of the water balance, including the spill term. A fully implicit solution would require an infinitely large value of INTERVALS. Setting INTERVALS to one signifies a fully explicit solution and the results should equal to that from the SETRELEASE method. 'INTERVALS' must be followed by: <ul style="list-style-type: none"> - an integer value
[INITIALSPILL]	Indicates the spill associated with the total discharge (release) from the reservoir at the beginning of the simulation period. This value will be subtracted from the Reservoir's INITIALRELEASE to separate the effects of the SPILLWAY Method from the other release; a value must follow. If INITIALSPILL is not defined it will default to 0.0.
ENDSPILLWAY	Indicates the end of parameters for Method SPILLWAY.

Carryover Transfer

The carryover saved consists of the INITIALSPILL only. Transfer of the carryover is direct, without transformation:

$$\text{INITIALSPILL(NEW)} = \text{INITIALSPILL(OLD)}$$

Keyword

Definition

Example

SPILLWAY ReservoirName MethodName

TABLE ELEV_SPILL

100 0

101 0.1

102 0.3

103 0.6

104 1.0

105 1.5

106 2.1

107 2.8

109 4.4

110 5.4

ENDTABLE

INTERVALS 12

INITIALSPILL 2.1

ENDSPILLWAY

Keyword Definition

[Next] [Previous] [Bookmarks] [Top]

Index

Application	2
Developed By	3
General Syntax Rules	4
Method ADJUST	1
Method BALANCE	1
Method CALCINFLOW	1
Method LAGK	1
Method MAXDECREASE	1
Method MAXINCREASE	1
Method MAXSTAGE	1
Method PASSFLOW	1
Method RAINEVAP	1
Method SETELEVATION	1
Method SETMAX	1
Method SETMIN	1
Method SETRELEASE	1
Method SETSUM	1
Method SETWITHDRAW	1
Method SPILLWAY	1
Organization	3
Parameters Definition	7
Parametric Information for each NODE Component	12
Parametric Information for each REACH Component	11
Parametric Information for each RESERVOIR Component	7
RES-J Method Syntax	18
Rules Definition	14
Table 1. RES€J Methods	26
Table 2. Time series keyword parameters	28
Table 3. RES€J Methods, components and variables	29
Time Series Definition	4

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