Command Reference: TS Alias = ChangeInterval()

Create a new time series by changing a time series data interval

Version 09.10.01, 2010-11-16

A `ChangeInterval()` command creates a new time series by changing the data interval of an existing time series. The majority of the original header data (e.g., description, units) are copied to the new time series. Time series data values have a time scale of accumulated (e.g., volume), mean, or instantaneous. Changing the interval can also result in a change in the time scale (e.g., converting instantaneous values to a mean value). Currently, the time scale for input and output time series is NOT automatically determined from the data type and interval and must be specified as ACCM, MEAN, or INST. Instantaneous values are recorded at the date/time of the value and typically apply to small intervals (e.g. minute and hour). For mean and accumulated time series, the date/time for each value is at the end of the interval for which the value applies.

Irregular time series have a date/time precision and a scale appropriate for the data. For example, irregular minute time series may be used for instantaneous temperature or accumulated precipitation. Irregular day time series may be used for "instantaneous" reservoir level. For regular time series, the data intervals must align so that each larger interval aligns with the end-points of the corresponding smaller intervals (e.g., the ends of 6-hour intervals align with the daily interval).

The following conversions are currently supported, with a description of the conversion process.

Irregular Time Series to Regular Time Series

An irregular time series can be converted to a regular time series. The ability to change from an irregular or regular time series to an irregular time series is not currently implemented. Missing data is handled in different ways depending on the old and new time scales. Each of the follow examples demonstrates how missing data is interpreted.

The following conversion combinations are allowed.

Small Interval ACCM to Large Interval ACCM

When converting from small interval accumulated data to large interval accumulated data, values from the old time series are summed for the new interval-ending date/time from the values in the old intervals prior to this date/time.

The following illustrates the conversion from NHour to NHour (1Hour to 3Hour example):

Day 1, Hour 0 (A)	Day 1, Hour 1 (B)	Day 1, Hour 2 (Missing)	Day 1, Hour 3 (C)	Day 1, Hour 4 (Missing)	Day 1, Hour 5 (Missing)	Day 1, Hour 6 (Missing)
Day 1, Hour 0 =A	Day 1, Hour 3 =B+C			Day 1, Hour 6 = Missing		

Large Interval ACCM to Small Interval ACCM

When converting from large interval accumulated data to small interval accumulated data, values from the old time series are equally divided by the number of intervals prior to this date/time in the new time series since the previous non-missing data.

The following illustrates the conversion from NHour to NHour (3Hour to 1Hour example):

Day 1, Hour 0 (A)	Day 1, Hour 3 (B)			Day 1, Hour 6 (Missing)			Day 1, Hour 9 (C)		
Day 1, Hour 0 =A	Day 1, Hour 1 =B/3	Day 1, Hour 2 =B/3	Day 1, Hour 3 =B/3	Day 1, Hour 4 =C/6	Day 1, Hour 5 =C/6	Day 1, Hour 6 =C/6	Day 1, Hour 7 =C/6	Day 1, Hour 8 =C/6	Day 1, Hour 9 =C/6

Small Interval MEAN or INST to Large Interval MEAN

When converting from instantaneous or mean data to mean data, mean values are calculated for the new interval-ending date/time from the values in the old intervals prior to this date/time.

The following illustrates the conversion from NHour to NHour (1Hour to 3Hour example):

Day 1, Hour 0 (A)	Day 1, Hour 1 (B)	Day 1, Hour 2 (Missing)	Day 1, Hour 3 (C)	Day 1, Hour 4 (Missing)	Day 1, Hour 5 (Missing)	Day 1, Hour 6 (Missing)
Day 1, Hour 0 =A	Day 1, Hour 3 =(B+C)/2			Day 2, Hour 6 = Missing		

Large Interval MEAN or INST to Small Interval MEAN

When converting from large interval mean or instantaneous data to small interval mean data, values from the old time series are copied to the new interval-ending date/time time series.

The following illustrates the conversion from NHour to NHour (3Hour to 1Hour example):

Day 1, Hour 0 (A)	Day 1, Hour 3 (B)			Day 1, Hour 6 (Missing)			Day 1, Hour 9 (C)		
Day 1, Hour 0 =A	Day 1, Hour 1 =B	Day 1, Hour 2 =B	Day 1, Hour 3 =B	Day 1, Hour 4 =C	Day 1, Hour 5 =C	Day 1, Hour 6 =C	Day 1, Hour 7 =C	Day 1, Hour 8 =C	Day 1, Hour 9 =C

Small Interval INST to Large Interval INST

When converting from small interval instantaneous data to large interval instantaneous data, the data is copied directly from the old time series when available. If the data is missing, the most recent previous valid data is used.

The following illustrates the conversion from NHour to NHour (1Hour to 3Hour example):

Day 1, Hour 0 (A)	Day 1, Hour 1 (B)	Day 1, Hour 2 (Missing)	Day 1, Hour 3 (C)	Day 1, Hour 4 (Missing)	Day 1, Hour 5 (D)	Day 1, Hour 6 (Missing)	Day 1, Hour 7 (E)	Day 1, Hour 8 (F)
Day 1, Hour 0 =A			Day 1, Hour 3 =C			Day 1, Hour 6 =D		

Large Interval INST to Small Interval INST

When converting from large interval instantaneous data to small interval instantaneous data, values from the old time series are linearly interpolated to calculate values for the new time series.

The following illustrates the conversion from NHour to NHour (3Hour to 1Hour example):

Day 1, Hour 0 (A)	Day 1, Hour 3 (B)			Day 1, Hour 6 (Missing)			Day 1, Hour 9 (C)		
Day 1, Hour 0 =A	Day 1, Hour 1 =A+ (B-A)* (1/3)	Day 1, Hour 2 =A+ (B-A)* (2/3)	Day 1, Hour 3 =B	Day 1, Hour 4 =B+ (C-B)* (1/6)	Day 1, Hour 5 = B+ (C-B)* (2/6)	Day 1, Hour 6 = B+ (C-B)* (3/6)	Day 1, Hour 7 = B+ (C-B)* (4/6)	Day 1, Hour 8 = B+ (C-B)* (5/6)	Day 1, Hour 9 =C

Regular Time Series to Regular Time Series

ACCM (Accumulation) to ACCM (Accumulation)

Small Interval ACCM (Accumulation) to Large Interval ACCM (Accumulation)

Changing the interval for small interval accumulated data to large interval accumulated data involves summing the small interval data values for the period that overlaps the large interval.

Accumulated data have a timestamp corresponding to the interval-end for the accumulation. Conversions involving time intervals that have zero values (e.g., Hour 0, Minute 0) result in a perceived shift in time because the zero occurs on the boundary between larger intervals. The following examples illustrate the accumulation for common cases. In cases where an accumulation jumps over two or more interval categories (e.g., minute to day), the accumulation occurs as if the two intermediate accumulations occurred in succession. In the following examples, the general representation is shown first, followed by an example where appropriate.

The following illustrates the conversion from NHour to Day (6Hour to Day example, *i* equals the hour multiplier):

Day 1, Hour *0*	Day 1, Hour *i*	Day 1, Hour *2i*	Day 1, Hour *3i*	Day 2, Hour *0*
Day 1, Hour 0	Day 1, Hour 6 (A)	Day 1, Hour 12 (B)	Day 1, Hour 18 (C)	Day 2, Hour 0 (D)
	Day 1 accumulation (A+B+C+D)			

The following illustrates the conversion from NDay to Month (example for a month with 30 days):

Month 1, Day 1 (A1)	…		…	Month 1, Day 30 (A30)
Month 1 accumulation (A1 + … + A30)				

Large Interval ACCM (Accumulation) to Small Interval ACCM (Accumulation)

Changing from large interval accumulation data to small interval mean data involves dividing each accumulated value by the number of new values for that same period of record.

The following illustrates the conversion from Day to 6Hour (Day to 6Hour example, *i* equals the hour multiplier):

Day 1 accumulate (A)			
Day 1, Hour *0*	Day 1, Hour *i*	Day 1, Hour *2i*	Day 1, Hour *3i*
Day 1, Hour 0 = A/4	Day 1, Hour 6 = A/4	Day 1, Hour 12 = A/4	Day 1, Hour 18 = A/4

ACCM (Accumulation) to INST (Instantaneous)

Accumulated to instantaneous is not currently supported.

ACCM (Accumulation) to MEAN

Small Interval ACCM to Large Interval MEAN

See **Small Interval INST (Instantaneous) to Large Interval MEAN.**

Interval ACCM to Same Interval MEAN

Changing the interval from accumulation data to the same interval mean data involves copying the data from the old time series to the new time series (no changes to date values occur).

The following illustrates the conversion from 6Hour to 6Hour (6Hour to 6Hour example, *i* equals the hour multiplier):

Day 1, Hour *0*	Day 1, Hour *i*	Day 1, Hour *2i*	Day 1, Hour *3i*
Day 1, Hour 0 (A)	Day 1, Hour 6 (B)	Day 1, Hour 12 (C)	Day 1, Hour 18 (D)
Day 1, Hour 0 =A	Day 1, Hour 6 =B	Day 1, Hour 12 =C	Day 1, Hour 18 =D

Large Interval ACCM to Small Interval MEAN

See **Large Interval ACCM to Small Interval ACCM.**

INST (Instantaneous) to INST (Instantaneous)

Small Interval INST (Instantaneous) to Large Interval INST (Instantaneous)

Changing the interval for small interval instantaneous data to large interval instantaneous data involves assigning each date in the new time series a value from the corresponding date in the old time series. The HandleMissingInputHow parameter indicates how to interpret a missing value in the old time series. HandleMissingInputHow=KeepMissing will simply assign a missing value for that date/time. HandleMissingInputHow=SetToZero will set the value to 0. Repeat fills the date with data from the last non-missing value. Interpolation and using a non-missing future value may be added in the future.

A special case is the ability to compute a statistic from the sample of values from the input time series, using the Statistic parameter. For example, instantaneous 5 minute temperature data can be converted to 1 day maximum values. In this case, each 1 day sample of values from the input time series is used to compute the statistic. The initial handling of missing data described above is supported and additionally the AllowMissingCount parameter is recognized to control computation of the statistic.

The following illustrates the conversion from NHour to Day (6Hour to Day example where HandleMissingInputHow = Repeat, *i* equals the hour multiplier):

Day 1, Hour *0*	Day 1, Hour *i*	Day 1, Hour *2i*	Day 1, Hour *3i*	Day 1, Hour *0*	Day 1, Hour *i*	Day 1, Hour *2i*	Day 1, Hour *3i*
Day 1, Hour 0 (A)	Day 1, Hour 6 (B)	Day 1, Hour 12 (C)	Day 1, Hour 18 (D)	*Missing data*	Day 1, Hour 6 (E)	Day 1, Hour 12 (F)	Day 1, Hour 18 (G)
Day 1 instantaneous = A				Day 2 instantaneous = D			

Large Interval INST (Instantaneous) to Small Interval INST (Instantaneous)

Small interval instantaneous data is created from larger interval instantaneous data by linearly interpolating between the previous and current large interval data to fill each value in the new time series during that same period of time. If the value in the old time series is missing, the method specified by the user in the HandleMissingInputHow parameter is used.

The following illustrates the conversion from Day to NHour (Day to 6Hour example, *i* equals the hour multiplier):

Day 1 instantaneous (A)				Day 2 instantaneous (B)				Day 3 …
Day 1, Hour *0*	Day 1, Hour *i*	Day 1, Hour 2*i*	Day 1, Hour *3i*	Day 2, Hour 0	Day 2, Hour i	Day 2, Hour 2i	Day 2, Hour 3i	
Day 1, Hour 0 =A	Day 1, Hour 6 =A+ (B-A)* (6/24)	Day 1, Hour 12 =A+ (B-A)* (12/24)	Day 1, Hour 18 =A+ (B-A)* (18/24)	Day 2, Hour 0 =B	Day 2, Hour 6 …	Day 2, Hour 12 …	Day 2, Hour 18 …	
These values are an interpolated value between the Day 1 instantaneous value and the Day 2 instantaneous value using a time of 24 hours.				These values are an interpolated value between the Day 2 instantaneous value and the Day 3 instantaneous value using a time of 24 hours.				

In the future, the ability to repeat input values may be added.

INST (Instantaneous) to ACCM (Accumulation)

Instantaneous to accumulated is not currently supported.

INST (Instantaneous) to MEAN

Small Interval INST (Instantaneous) to Large Interval MEAN

Changing from small interval instantaneous data to large interval mean data involves adding together all the values from the small interval time series over the larger interval for the corresponding time period and then dividing by the number of data values used within this calculation. As in other conversions, `HandleMissingInputHow` is first used to interpret missing data. If `HandleEndpointHow = AverageEndpoints`, the values at each end of the interval are averaged for minute and hour inputs (the parameter does not apply to day, month or year input).

The following illustrates the conversion from NHour to Day (6Hour to Day example with `HandleEndpointHow = IncludeFirstOnly`, *i* equals the hour multiplier):

Day 1, Hour *0*	Day 1, Hour *i*	Day 1, Hour 2*i*	Day 1, Hour *3i*	Day 2, Hour *0*	Day 2, Hour *i*	Day 2, Hour *2i*	Day 2, Hour *3i*
Day 1, Hour 0	Day 1, Hour 6	Day 1, Hour 12	Day 1, Hour 18	Day 2, Hour 0	Day 2, Hour 6	Day 2, Hour 12	Day 2, Hour 18
Value A	B	C	D	E	F	G	H
Day 1 mean= (A+B+C+D)/4				Day 2 mean=(E+F+G+H)/4			

The following illustrates the conversion from NHour to Day (6Hour to Day example with
`HandleEndpointHow = AverageEndpoints`, *i* equals the hour multiplier):

Day 1, Hour *0*	Day 1, Hour *i*	Day 1, Hour *2i*	Day 1, Hour *3i*	Day 2, Hour *0*	Day 2, Hour *i*	Day 2, Hour *2i*	Day 2, Hour *3i*	
Day 1, Hour 0	Day 1, Hour 6	Day 1, Hour 12	Day 1, Hour 18	Day 2, Hour 0	Day 2, Hour 6	Day 2, Hour 12	Day 2, Hour 18	
Value A	B	C	D	E	F	G	H	I
Day 1 mean= ((A+E)/2 +B+C+D) / 4				Day 2 mean=((E+I)/2+F+G+H) / 4				

Interval INST (Instantaneous) to Same Interval MEAN

If `OutputFillMethod = Interpolate`, see **Large Interval INST (Instantaneous) to Small
Interval INST (Instantaneous).** Otherwise, the values are duplicated from the old time series directly to
the new time series.

The following illustrates the conversion from 6Hour to 6Hour (6Hour to 6Hour example with
`OutputFillMethod = Repeat`, *i* equals the hour multiplier):

Day 1, Hour *0*	Day 1, Hour *i*	Day 1, Hour *2i*	Day 1, Hour *3i*	Day 2, Hour *0*
Day 1, Hour 0 (A)	Day 1, Hour 6 (B)	Day 1, Hour 12 (Missing)	Day 1, Hour 18 (D)	Day 2, Hour 0 (E)
Day 1, Hour 0 =A	Day 1, Hour 6 =B	Day 1, Hour 12 =B	Day 1 Hour 18 =D	Day 2, Hour 0 =E

Large Interval INST (Instantaneous) to Small Interval MEAN

If the `OutputFillMethod = Interpolate`, see **Large Interval INST (Instantaneous) to Small
Interval INST (Instantaneous).** The time series are handled in the same way. Otherwise, the values are
duplicated from the old time series directly to the new time series.

The following illustrates the conversion from Day to 6Hour (Day to 6Hour example with
`OutputFillMethod = Repeat`, *i* equals the hour multiplier):

Day 1 instantaneous = A			
Day 1, Hour *0*	Day 1, Hour *i*	Day 1, Hour *2i*	Day 1, Hour *3i*
Day 1, Hour 0 =A	Day 1, Hour 6 =A	Day 1, Hour 12 =A	Day 1, Hour 18 =A
Each of these values is equal to the instantaneous value for that day.			

MEAN to MEAN

Small Interval MEAN to Large Interval MEAN

See **Small Interval INST (Instantaneous) to Large Interval MEAN.**

Large Interval MEAN to Small Interval MEAN

Changing from large interval mean data to small interval mean data involves copying values from the old time series into the new time series for that same period of record.

The following illustrates the conversion from Month to Day (Example for a month with 30):

Month Mean (A)			
Day 1 =A	Day 2 =A	…	Day 30 =A

MEAN to ACCM (Accumulation)

Small Interval MEAN to Large Interval ACCM (Accumulation)

See **Small Interval INST (Instantaneous) to Large Interval MEAN.**

Interval MEAN to Same Interval ACCM (Accumulation)

See **Interval ACCM to Same Interval MEAN.**

Large Interval MEAN to Small Interval ACCM (Accumulation)

See **Large Interval ACCM to Small Interval ACCM.**

MEAN to INST (Instantaneous)

Small Interval MEAN to Large Interval INST (Instantaneous)

Not currently supported.

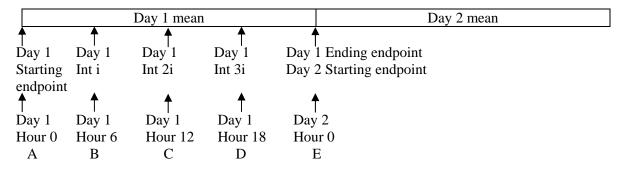**Interval MEAN to Same Interval INST (Instantaneous)**

Not currently supported. The data can be treated equivalently by most commands.

Large Interval MEAN to Small Interval INST (Instantaneous)

Changing the interval for large interval mean to small interval instantaneous data involves calculating a value for each new interval based on trends found in the mean data.   This approach has been adapted from the NWSRFS CHANGE-T operation (see http://www.nws.noaa.gov/oh/hrl/nwsrfs/users_manual/part5/_pdf/533changet.pdf). The following example demonstrates how the data is converted from the old interval to the new interval. A general representation is shown first followed by an example.

The following illustrates the conversion from Day to NHour (Day to 6Hour example, *i* equals the hour multiplier):

Day 1 mean	Day 2 mean

Day 1   Day 1   Day 1   Day 1   Day 1 Ending endpoint
Starting   Int i   Int 2i   Int 3i   Day 2 Starting endpoint
endpoint

Day 1   Day 1   Day 1   Day 1   Day 2
Hour 0   Hour 6   Hour 12   Hour 18   Hour 0
  A   B   C   D   E

In computing instantaneous values, the volume of the original mean time series needs to be maintained. However, the value of the instantaneous endpoints affects the calculated mean value for the previous and subsequent long intervals, since the mean over a longer interval uses both endpoints in its calculation. In the above example, $\text{Average}_{\text{new Day 1}} = ((A+E)/2 + B+C+D) / 4$
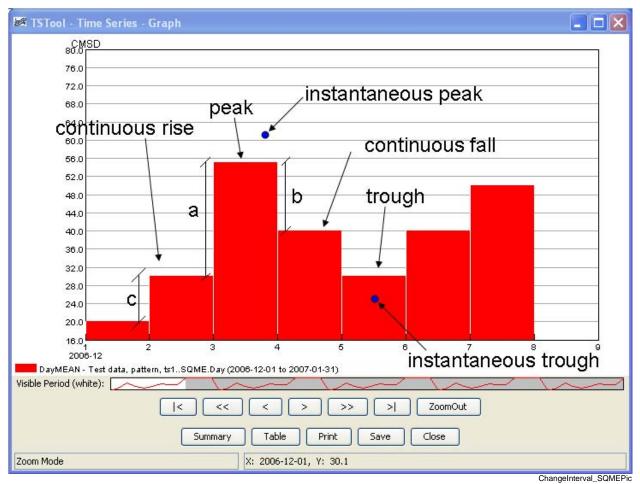
As a result, an iterative technique is required to adjust the initially computed instantaneous values to produce a time series with a volume that is within a specified tolerance of the input mean volume for each time step. The following paragraphs describe how the initial instantaneous time series values are computed, followed by a description of the volume adjustment.

Initial Instantaneous Time Series Calculations

Prior to converting from a large interval mean to small interval instantaneous, special cases are handled associated with missing data:

*   Missing data is initially converted using the method specified by the user in the `HandleMissingInputHow` parameter.
*   If the current input value is still missing, the instantaneous time series is also filled with missing data for each interval that falls in the larger interval.
*   If the previous or next mean values are missing, the current mean value for that interval is copied directly to the instantaneous time series.

The output instantaneous values for each input interval are computed using the current, next, and previous mean values. All three values are useful because together they indicate whether the current value is part of a continuous rise or fall, a peak or trough or simply a continuation of a steady value. These conditions are illustrated in the following figure.

ChangeInterval_SQMEPic

Mean data illustration

- The first condition that may exist is a **peak or trough**. A peak exists when the current value is greater than the previous and next values. A trough is when the current value is less than the next and previous values.
    1. In this case, an *instantaneous peak* (or trough) is calculated. Referring to the above illustration, the magnitude of the peak is calculated by adding (or subtracting for a trough) $\frac{1}{4}$ $(a+b)/2$ to the current mean.
    2. The *time of the instantaneous peak* is initially set to the start date/time of the current interval then shifted forward in time using the following calculation. The number of instantaneous intervals per larger interval is multiplied by $b/(a+b)$. That result is added to the start date/time. The time of the instantaneous peak will not necessarily correspond to the output interval.
    3. The value for the starting endpoint of the interval is set to the *current value minus $\frac{1}{4}$ a*.
    4. The value for the ending endpoint of the interval is set to the *current value minus $\frac{1}{4}$ b*.
    5. The remaining intermediate instantaneous values for the interval are linearly interpolated between the peak (or trough) and both endpoints.
- The second condition that may exist is a **continuous rise or fall**. A continuous rise or fall exists when the current value is between the previous and next values.
    1. In the case of a continuous rise, the starting endpoint of the interval is set to the *current value minus $\frac{1}{4}$ c* (again using the above illustration). In the case of a continuous fall, $\frac{1}{4}$c is added to the current value.

2. The ending endpoint of the interval of a continuous rise is set to the *current value plus¼ c* (minus ¼ c for a continuous fall)
3. The remaining intermediate instantaneous values are calculated based on the following.
    a. The difference between the starting and ending endpoints is computed.
    b. The values *c* and *a* are calculated. The ratio of mean differences is computed: If *c > a*, the mean ratio = *c / a*. If *a > c*, the mean ratio = *a / c*.
    c. If *c* is less than *a*, then the intermediate instantaneous values are computed by adding small but increasing increments to the starting endpoint until the last point of the interval is reached. If *a* is less than *c* then the output values are computed by subtracting small but increasing increments to the last endpoint until the first point of the interval is reached. For intermediate interval *n* from the appropriate endpoint:

$$Increment_n = (1/([number\ intervals] - n) * [endpoint\ difference] * [mean\ ratio])$$

Instantaneous Time Series Volume Adjustment

After the instantaneous values are estimated using the above set of rules, they are adjusted so that the volume over each interval is within a specified tolerance of the input mean volume for each time step. This tolerance is specified with the `Tolerance` parameter. The volume calculation for each interval uses the average of the first and last endpoint.
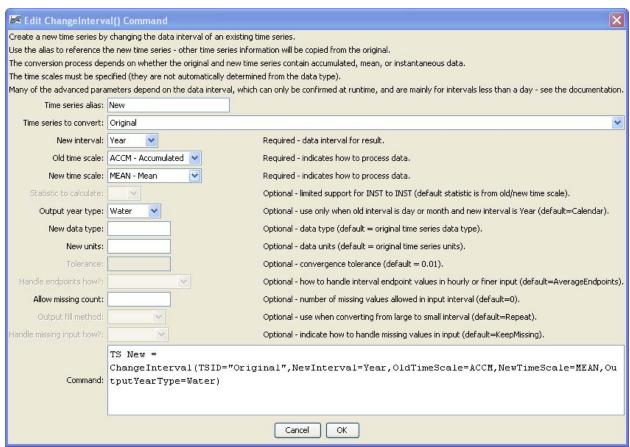
For each time step of the original mean time series, the error of each original interval is computed as:

([mean of current instantaneous values] – [original mean]) / [original mean]

If this volume error is within the tolerance, no adjustment is made for that time interval. If the error is larger than the tolerance, the ratio: *[original mean] / [mean of current instantaneous values]* is computed. For the intermediate instantaneous values, the instantaneous values are adjusted by multiplying each value by this ratio. For the instantaneous endpoint values, since these values affect the mean value of the current and the previous or following time intervals, the endpoint values are not adjusted to the above calculated ratio. Instead, the new endpoint value is computed as the average of the original endpoint value and the original endpoint value multiplied by the above ratio.

The volume error is checked for each original time step. If any adjustments were made, the process is repeated, up to 15 iterations. If the adjustment technique cannot adjust the instantaneous time series such that the corresponding mean volume is within the specified tolerance of the input mean volume within 15 iterations, a warning will be written to the log file.

The following dialog is used to edit the command and illustrates the syntax for the command. This example is converting a monthly volume time series to annual water year (October to September) volumes.

ChangeInterval() Command Editor

The command syntax is as follows:

```
TS Alias = ChangeInterval(Parameter=Value,…)
```

Command Parameters

Parameter	Description	Default
Alias	The alias to assign to the new time series. The time series identifier for the new time series will be the same as the original time series, with the new interval and optionally a new data type (see the `NewDataType` parameter).	None – must be specified.
TSID	The time series identifier or alias for the original (old) time series.	None – must be specified.
NewInterval	The data interval for the new time series, from the provided choices. For example: `6Hour`, `Day`, `Month`, `Year`.	None – must be specified.
OldTimeScale	The time scale for the original time series, one of: `ACCM` – accumulated data `INST` – instantaneous data `MEAN` – mean data	None – must be specified.

	In the future, this parameter may be made optional if the time scale can be determined from the data type.	
NewTimeScale	The time scale for the new time series (see OldTimeScale for possible values). In the future, this parameter may be made optional if the time scale can be determined from the data type.	None – must be specified.
Statistic	Used in the case where INST (small interval) to INST (large interval) conversion is occurring. A sample of values from the input time series, corresponding to the output interval, is determined and used to compute a statistic instead of a simple value transfer. Statistics that are currently supported are MAX and MIN. The HandleMissingInputHow parameter is initially used to adjust missing data and then the AllowMissingCount parameter is used to check whether the statistic can be computed.	The statistic is determined from the old and new time scales.
OutputYearType	The output year type if the output time series has an interval of Year. The output year type can only be specified for input time series having an interval of Day or Month and the output can have a time scale of ACCM (sum the input values) or MEAN (average the input values). The AllowMissingCount parameter is recognized.	Calendar
NewDataType	The data type for the new time series. This will be set in the identifier of the new time series.	Use the data type from the original time series.
NewUnits	The units for the new time series. This will be set in the identifier of the new time series.	Use the units from the original time series.
Tolerance	Currently used when converting large interval MEAN data to small interval INST data. After the new time series is created, the volume of the new time series over each old interval is compared to the old time series for that same interval. If the difference between the two is outside the specified tolerance percentage, then each value in the new time series is adjusted so the totals will match. The endpoints are averaged for this comparison. Additionally, when the adjustment is made, the new starting value is averaged with the ending value of the previous interval so that the previous interval is not overly affected by this calculation.	0.01
Handle EndpointsHow	Indicates how endpoints should be handled when changing from INST to MEAN, small interval to larger interval (daily output or finer), one of: AverageEndpoints – use both endpoint values for new single value IncludeFirstOnly – only use earlier endpoint	Average Endpoints
AllowMissing Count	The number of missing values allowed in the source	0 – do not allow

	interval(s) in order to produce a result. For example, if converting daily data to monthly, a value of 5 would allow <= 5 missing daily values and still compute the result. This capability should be used with care because it may result in data that are not representative of actual conditions.	any missing data in the source data when computing a result.
OutputFill Method	Use to fill output when converting from INST to MEAN, large interval time series to small interval time series, one of: Interpolate – linearly interpolate Repeat – repeat values for the output	Repeat
HandleMissing InputHow	Indicate how to handle missing values in input, one of: KeepMissing – leave data missing Repeat – repeat last non-missing value SetToZero – set values to 0 The missing data is handled on input and the replacement value, if any, is applied to input and used for calculations just as if it was the actual value. The following cases do not use this parameter: • Irregular data • Day and Month input converted to ACCM and MEAN.	KeepMissing

Several example command files follow. The following command creates a Day ACCM time series from a Month ACCM time series:

```
0109.NOAA.Precip.Day~HydroBase
TS 0109Month =
ChangeInterval(TSID="0109.NOAA.Precip.Day",NewInterval=Month,OldTimeScale=ACCM,NewTimeScale=ACCM)
```

The following commands create a 6Hour INST time series from a Day MEAN time series:

```
TS DayMEAN = NewPatternTimeSeries(NewTSID="ts1..SQME.Day",Description="Test data",
SetStart="2006-12-01",SetEnd="2007-01-31",
Units="CMSD",PatternValues="20,30,55,40,30,40,50,45,45,80,80,80,80")

TS 6HourINST =
ChangeInterval(TSID="DayMEAN",NewInterval=6Hour,OldTimeScale=MEAN,NewTimeScale=INST,
NewDataType=CMS)
```

The following commands create a Day MEAN time series from a 6Hour INST time series:

```
TS 6HourInst =
NewPatternTimeSeries(NewTSID="ts2..Flow.6Hour",IrregularInterval=6Hour,Description="Test
data",SetStart="2006-12-15 12",SetEnd="2007-01-29
00",Units="CFS",PatternValues="20,23,56,62,35,42")

TS DayMean2 =
ChangeInterval(TSID="6HourInst",NewInterval=Day,OldTimeScale=INST,NewTimeScale=MEAN,
HandleEndpointsHow=IncludeFirstOnly)
```

The following commands create a 3Hour INST time series from an Irregular (1Hour) INST time series:

```
TS IrregularINST =
NewPatternTimeSeries(NewTSID="ts1..Temp.Irregular",IrregularInterval=1Hour,Description="Test
data",SetStart="2006-12-15 00",SetEnd="2007-01-31 23",Units="DEGF",
PatternValues="20,23,-999,45,-999,-999,56,62,0,-3")

TS 3HourINST =
ChangeInterval(TSID="IrregularINST",NewInterval=3Hour,OldTimeScale=INST,NewTimeScale=INST)
```

This page is intentionally blank.