Appendix: NWSRFS ESP Trace Ensemble Input Type

2013-04-01

Note: The binary NWSRFS ESP ensemble format is a legacy format and may be phased out in the future as alternative formats are implemented. For example, see the Delft FEWS PI XML input type, which is now used by the National Weather Service to disseminate ensemble data.

Overview

The NWSRFS ESP trace ensemble file format stores one or more time series traces of consistent time interval (1 to 24 hours), data type, units, etc. The format has been developed by the National Weather Service (NWS) to support its Ensemble Streamflow Prediction (ESP) software. The ESP system is that portion of the National Weather Service River Forecast System (NWSRFS) that provides the capability of making long-range probabilistic forecasts of streamflow and streamflow related variables. ESP uses conceptual hydrologic/hydraulic models to forecast future streamflow using the current soil moisture, river, and reservoir conditions, with historical meteorological data as input. The ESP procedure assumes that meteorological events that occurred in the past are representative of events that may occur in the future. Each year of historical meteorological data is assumed to be a possible representation of the future and is used to simulate a streamflow trace. ESP produces a probabilistic forecast for each streamflow variable and period of interest. Although often applied to streamflow, the ESP approach and data format can be utilized for other time series data types.

The ESP system can produce two types of stream flow simulations: Conditional and Historic Simulations (CS and HS respectively). Conditional simulation files are produced when ESP repeatedly runs a streamflow simulation, using consistent starting conditions (e.g., the current system states) with a sequence of historical data as inputs. The resulting ensemble of time series traces can be analyzed to determine probabilistic qualities of potential streamflow forecasts. Although often applied to streamflow forecasts, the trace ensembles can also contain other data types (e.g., reservoir levels and releases). For the historical simulation, ESP simulates the entire historical period of record continuously without resetting the initial conditions. The analysis of the historical simulation can be compared to the analysis of the observed flows to assess any bias that might exist in the system. This information is often used to subjectively adjust the conditional simulation before a forecast is made. HS (historical) simulations are used to overcome the limited forecast window in the operational system. Currently the only ESP trace ensemble simulation files compatible with TSTool are conditional simulation files. Therefore, the information below applies to CS files only and do not apply to HS files.

ESP trace ensemble files, when used with NWSRFS, are typically named according to the convention:

Segment.TimeSeries.DataType.Interval.Scenario

For example:

GRCCH.GRCCH.QINE.06.HS
OWSN6.OWSSIMEL.SPEL.06.CS

The data type is specified using standard NWS data types, as found in the NWSRFS *DATAUNIT* system file. The interval is hours. The scenario is typically HS for a historical ESP run and CS for a conditional simulation.

The ESP trace ensemble file is a FORTRAN direct access binary file consisting of header records and time series data. The format of the file is conducive to the operation of ESP software. The file header section consists of one 496-byte record, stored as 124 words of 4-byte data, in the format below. Character data in the header record are stored with spaces padding them out to their maximum length. Consequently, a value of '`ETC`' in the Segment ID, which is an 8-byte character field, will be stored in the file as '`ETC `'. Null characters are not used to pad strings.

The header record only has data through the 412th byte, beyond which the record consists of null values (byte value 00), which should be ignored. The specific format of the header record is described in the following table (data members refer to code variables, to aid developers).

ESP Trace Ensemble File Field Description

Record	Field Number (>= 1)	Field Type	Bytes	Field Starting Byte (in Record)	Data member (from code)	Data Description
1	1	F4	4	0	_format_ver	Format version. e.g., 1.01. There is no logic in existing ESPADP code to do anything differently based on the format version.
1	2	C8	8	4	_seg_id	Segment ID (e.g., '`OWSN6 `').
1	4	C8	8	12	_ts_id	TS ID (e.g., '`OWSSIMEL`').
1	6	C4	4	20	_ts_type	NWS data type (e.g., '`QINE`').
1	7	I4	4	24	_ts_dt	TS time interval in hours (e.g., 6).
1	8	I4	4	28	_simflag	Simulation flag, set to one of: 0 – SIMFLAG_CONDITIONAL 1 – SIMFLAG_HISTORICAL 2 – SIMFLAG_OBSERVED
1	9	C4	4	32	_ts_unit	NWS data units (e.g., '`CMS `')
1	10	I4	4	36	now[0]	Trace file creation month (1-12).
1	11	I4	4	30	now[1]	Trace file creation day (1-31).
1	12	I4	4	44	now[2]	Trace file creation year (4-digit).
1	13	I4	4	48	now[3]	Trace file creation date hours (1-24) and minutes (0-59). The hour is (int)(now[3]/100), the minutes are (int)(now[3]%100).
1	14	I4	4	52	now[4]	Trace file creation date seconds (0-59) and hundredths of seconds 0-99). The seconds are (int))(now[4]/100), the hundredths of seconds are (int)(now[4]%100).
1	15	I4	4	56	_im	Month of the first day of the first time series in the file (1-12).
1	16	I4	4	60	_iy	Year of the first day of the first time series in the file (4-digit).
1	17	I4	4	64	_idarun	Start if the traces as Julian day relative to December 31, 1899, in Zulu time.

Record	Field Number (>= 1)	Field Type	Bytes	Field Starting Byte (in Record)	Data member (from code)	Data Description
1	18	I4	4	68	_ldarun	End of the traces as Julian Day format relative to December 31, 1899, in Zulu time.
1	19	I4	4	72	_ijdlst	Carryover (states) day (1-31)
1	20	I4	4	76	_ihlst	Carryover (states) hour (1-24)
1	21	I4	4	80	_ljdlst	Last day of forecast in Julian Day format relative to December 31, 1899, in Zulu time.
1	22	I4	4	84	_lhlst	Last hour of forecast (1-24), in Zulu time.
1	23	I4	4	88	_n_traces	Number of traces in ESP file.
1	24	I4	4	92	_ncm	Number of conditional months.
1	25	I4	4	96	_nlstz	NWSRFS time zone relative to Zulu time. E.g. –6 = Mountain Standard Time.
1	26	I4	4	100	noutds	NWSRFS daylight savings flag. Either: 0 – No daylight savings time 1 – Daylight savings time
1	27	I4	4	104	_irec	Record number of first trace data, typically 2 (allows for future growth of the header information).
1	28	C4	4	108	_dim	Unit dimensions from NWS data units.
1	29	C4	4	112	_tscale	Time scale of code (INST, MEAN, ACCM)
1	30	C20	20	116	seg_desc	Segment description (from NWSRFS database).
1	35	F4	4	136	xlat	Latitude of segment in decimal degrees
1	36	F4	4	140	xlong	Longitude of segment in decimal degrees
1	37	C8	8	144	fg	Forecast group (from NWSRFS database).
1	39	C8	8	152	cg	Carryover group (from NWSRFS database).
1	41	C8	8	160	rfcname	RFC name (from NWSRFS database).
1	43	C80	80	168	_espfname	Trace file name, without path. (e.g., `GRCCH.GRCCH.QINE.06.HS`).
1	63	C80	80	248	prsf_string	This is the PRSF flag value that is used by the ESPADP NWSRFS application to display hours on the X-axis.
1	83	C80	80	328	_esptext	User comments
1	103	I4	4	408	adjcount	Adjustment counter 0-9. No further information can be determined about this data at this time.
1	104	C84	84	412	Dummy	Dummy contains 84 bytes of null (to be ignored)
2-EOF						Data records – see below.

Some ESP trace ensemble files contain additional records before data records, including analysis information. However, in most cases, the single header record is immediately followed by data records. The data section of the ESP file contains multiple records with 496 bytes of data in 124 4-byte floats. Records are stored in the file by trace in ascending date order. In other words, the first group of data records is for trace1, the second group of data records is for trace2, etc.

Each record contains 124 words of 4-byte floating-point numbers, regardless of the type of trace data interval (24-hour, 6-hour, hourly). Consequently, for 24-hour time series one record will contain:

124 values/1 value-per-day = 124 floating point numbers = 4 months of data per record

For hourly time series, one record will contain:

124 values/24 values-per-day = 5 days of data per record

If the trace period of record covers an entire year (365 or 366 days of data), the number of records for one trace of 24-hour time series will be:

365 days/124 days per record = 3 records per trace

For hourly time series, the number of records will be:

365 days/5 days per record = 73 records per trace

There are some qualifications on the format of the data records. These include:

- All the data values within a trace are sequential. This means that the May 1 data always follow the April 30 data.
- Traces consist of sequential data. February 28 is always followed by data values, whether taken from February 29 or March 1 of historical data.
- Each trace begins at the start of a new record. Therefore, a trace may not fill its final record to the 496th byte.
- If a trace does not fill the end of its final record, the end of the record will be filled with null bytes.

Due to these complexities, it is recommended that users not try to read data values from an ESP trace ensemble file – software like TSTool can read the trace file and translate the data into a format that can be more easily manipulated.

ESP Trace Ensemble Files and Standard Time Series Properties

The standard time series identifier for ESP trace ensemble files is of the form:

```
Location.NWSRFS.DataType.Interval[Trace]~NWSRFS_ESPTraceEnsemble~PathToFile
```

Similar to other input types (e.g., `DateValue`), ESP trace ensemble files support storing multiple time series in the same file. However, whereas other input types typically store time series for different stations or data type, ESP trace ensemble files store one or more traces of time series at the same station, where header information is the same for each time series, and each trace corresponds to a different

historical input period (or year). To uniquely identify each trace, a time series sequence number (trace) is used in the time series identifier. This sequence number is the historical year for the start of each trace. For example, valid time series identifiers are:

```
GRCCH.NWSRFS.SQIN.6[1950]~NWSRFS_ESPTraceEnsemble~PathToFile
GRCCH.NWSRFS.SQIN.6[1951]~NWSRFS_ESPTraceEnsemble~PathToFile
…
GRCCH.NWSRFS.SQIN.6[2003]~NWSRFS_ESPTraceEnsemble~PathToFile
```

When reading a trace ensemble file:

- Each trace is converted to a separate time series, where the period is consistent with the overlapping period (e.g., the carryover date +1 interval forward through the end of a simulation).
- Each time series has redundant header information matching the ensemble header information.
- Each time series is given a sequence number (a time series property) corresponding to the starting year for the trace input data. For example, if the period that the traces are being generated is 2002-01-01 to 2002-12-31 and historical data starting in 1950 are being used, then the first trace time series will have a period 2002-01-01 to 2002-12-31 and a sequence number 1950. The second trace time series will have a period 2002-01-01 to 2002-12-31 and a sequence number 1951.
- Each time series is assigned an alias of the format `Segment_Trace_Seq`, where the segment identifier is taken from the file, "`_Trace_`" is a literal string, and sequence number is as described above.
- The location is set to the segment identifier.
- The data source is set to NWSRFS, because the NWSRFS system has generated the simulated time series.
- Data type is assigned based on the value in the file.
- Data units are assigned based on the value in the file.
- The interval is set based on the value in the file.
- The input type is set to `NWSRFS_ESPTraceEnsemble`.
- The input name is set to the ESP trace ensemble file name.

Some header information may be lost when converting between different time series input types (i.e., if the time series traces read from an ESP trace ensemble file are saved to a DateValue file. The following table illustrates how the internal ESP trace header information is stored in time series data after the trace file is read. In many cases, the specific trace ensemble header data are simply saved to the text comments in the time series.

Similarly, converting other data formats to an ESP trace file any header requires calculating some values and defaulting others. Although most values (such as _ncm, and _idarun) are computed and inserted into the trace header, some header information cannot be assigned without user input. TSTool does allow for some of these lost header fields to be inserted through options in the `WriteNwsrfsEspTraceEnsemble()` command.

ESP Trace Ensemble File Data and Associated Time Series Data

Data	Time Series Trace Data
_format_ver	Stored in TS comments and general time series property.
_seg_id	Stored in TS comments and general time series property.
_ts_id	Stored in TS comments and general time series property.

Data	Time Series Trace Data
_ts_type	Stored in TS comments and general time series property.
_ts_dt	Data Interval, and general time series property
_simflag	Stored in TS comments and general time series property.
_ts_unit	Data Units, Data Units Orig, and general time series property
now[0]	Stored in TS comments and general time series property.
now[1]	Stored in TS comments and general time series property.
now[2]	Stored in TS comments and general time series property.
now[3]	Stored in TS comments and general time series property.
now[4]	Stored in TS comments and general time series property.
_im	Stored in TS comments and general time series property.
_iy	Stored in TS comments and general time series property.
_idarun	Stored in TS comments and general time series property, used to calculate Date1 and Date1 Original: Convert Julian Hours from [((_idarun – 1) * 24) + (_ihlst – 1)] to DateTime
_ldarun	Stored in TS comments and general time series property, used to calculate Date2 and Date2 Original: Convert Julian Hours from [((_ldarun – 1) * 24) + (_lhlst – 1)] to DateTime
_ijdlst	Stored in TS comments.
_ihlst	Stored in TS comments and general time series property, used to calculate Date1 and Date1 Original: Convert Julian Hours from [((_idarun – 1) * 24) + (_ihlst – 1)] to DateTime
_ljdlst	Stored in TS comments.
_lhlst	Stored in TS comments and general time series property, used to calculate Date2 and Date2 Original: Convert Julian Hours from [((_ldarun – 1) * 24) + (_lhlst – 1)] to DateTime
_n_traces	Implicitly stored as the number of time series and general time series property.
_ncm	Stored in TS comments and general time series property.
_nlstz	Stored in TS comments and general time series property.
noutds	Stored in TS comments and general time series property.
_irec	Stored in TS comments and general time series property.
_dim	Stored in TS comments and general time series property.
_tscale	Stored in TS comments and general time series property.
seg_desc	Description, and general time series property
xlat	Stored in TS comments and general time series property.
xlong	Stored in TS comments and general time series property.
fg	Stored in TS comments and general time series property.
cg	Stored in TS comments and general time series property.
rfcname	Stored in TS comments and general time series property.
_espfname	TS Identifier and general time series property.
prsf_string	Stored in TS comments and general time series property.
_esptext	Stored in TS comments and general time series property.
adjcount	Stored in TS comments and general time series property.

Limitations

ESP trace ensemble files have the following limitations:

- The format is specific to hourly data.
- Multi-year trace periods are difficult to process.
- The binary, fixed record length format is not as flexible as text-based time series structures.