IVP Batch Program User's Manual For Pairing

By

Henry Herr
Office of Hydrologic Development
National Weather Service

Table of Contents

1.0 Overview

The IVP Batch Program serves two functions:

(1) constructing forecast-observed data pairs,
(2) calculating verification statistics, and
(3) generating graphical products.

This user's manual provides instructions for the first function, creating data pairs that are stored in the vfypairs and vfyprocpairs tables of the archive database.

1.1 Using This Manual

This manual provides a description of the format of an IVP Batch Program input batch file for pairing. Refer to Section 7.0 for instructions on how to put together a batch input file. Look over the examples prior to running the IVP Batch Program for generating pairs. Refer to the Section on the batch commands, as needed, in order to learn how to specify a particular batch command value.

2.0 Notation and Definitions

This section provides definitions used throughout this section and the remaining sections. Additional definitions to those in this section will be provided as needed. The definitions are as follows:

- *input token*: The `<token>` portion of a batch file input line. All tokens are displayed in `this font.`
- *token value*: The `<value>` portion of a batch file input line. All values are displayed in `this font.`
- *batch command*: An input token that is used to specify a parameter. Its token value is stored by the batch file processor. A batch command does not result in any calculations or output (except logging output written to the terminal). Batch commands are displayed in `this font.`
- *batch action*: An input token that triggers a particular action, such as querying the archive database for pairs or performing calculation of verification statistics. The token value specifies parameters of that action. Batch actions are displayed in `this font.`
- *data pair*: A forecast-observed pair, defined in the vfypairs or vfyprocpairs table of the archive database.

3.0 Execution

Before executing the IVP Batch Program, be sure that the tables vfyruninfo, rivercrit, and location are all populated correctly for each location for which verification is to be done. The vfyruninfo table is populated using the Vfyruninfo Editor. The rivercrit table must be populated in order for the flood stages and flows to be found. The location table must be populated in order for the rfc to be identified for a location, given by the field rfc. If no rfc is found, then the rfc is assumed to be "NONE".

To execute the Verification Batch Program enter:

```
cd $(get_apps_defaults verify_dir)/scripts
ivpbatch [-c] <batch_file_name>
```

where `<batch_file_name>` is the name of the batch file. If the first letter of the name of the batch file is either '.' or '/', then the file name is assumed to be fully specified, relative to the current directory. Otherwise, it is assumed to be specified relative to the directory given by the apps-defaults token "vsys_input". Use the `-c` option only if executed within a cron.

4.0 Apps-defaults Tokens

The following apps-defaults tokens are used by the IVP Batch Program:

- `adb_name` : <depends upon RFC>
- `util_rls` : $(util_dir)/bin/RELEASE
- `verify_dir` : /rfc_arc/verify
- `vsys_dir` : $(verify_dir)
- `vsys_input` : $(vsys_dir)/input
- `vsys_output` : $(vsys_dir)/output
- `vsys_files` : $(vsys_dir)/files
- `rax_pghost` : ax <optional; if not present, it uses the `PGHOST` environment variable>
- `pguser` : pguser <optional; if not present, it uses the `PGUSER` environment variable>
- `pgport` : 5432 <optional; if not present, it uses the `PGPORT` environment variable>

Each of the above directories must exist for the IVP Batch Program to run properly. It is also recommended that the following directory be created for each user:

$(vsys_input)/$LOGNAME

If this recommendation is followed, then the apps-defaults site file should override the setting of vsys_input as follows:

- `vsys_input` : $(vsys_dir)/input/$(LOGNAME)

All of these directories should be constructed prior to running the IVP Batch Program.

5.0 Files and Directory Structure

All of the directories corresponding to apps-defaults tokens in Section 4.0 must exist. In addition, the following files must be in place for the IVP to execute properly:

```
$(util_rls)/rfc.ohd.jar
$(sys_java_dir)/lib/ChartDirector.jar
/usr/share/java/pg74.215.jdbc3.jar
$(vsys_dir)/scripts/ivpbatch
```

6.0 Batch File Format

Each line of the batch file corresponds to a command or a parameter setting. All lines must be of the format:

```
<token> = <value>
```

with the following restrictions:

1. The token is not case sensitive.
2. Any number of spaces or tabs may be placed before or after the token and before or after the value.
3. A space, new-line (carriage return), tab, or pound ('#') marks the end of a value or token.
4. Double quotes must be placed around the value if it is to contain tabs, spaces, or pounds, but the value may never contain a new-line. For example,

    ```
    my_name = john doe
    ```

    has a token of "my_name" and a value of "john", whereas

    ```
    my_name = "john doe"
    ```

    has a token of "my_name" and a value of "john doe". If a new-line is encountered, it is treated as the closing double-quote.
5. The character '#', unless it is within double quotes, is used to indicate a comment. All characters after a '#' are ignored.
6. The equal sign ('=') must not be used as part of a value.

If a line is found which does not follow this format or specifies an unrecognized token, an error message will be generated. Blank lines are ignored.

7.0 Instructions

The following steps can be used to setup a batch input file for constructing pairs:

1. Use the Vfyruninfo Editor GUI to define locations to be included in data pairing. See the *Vfyruninfo Editor User's Manual* for more details.
2. Decide on the time interval for which pairing is to be done. Set the time frame in the batch file using the `START_TIME` and `END_TIME` commands.
3. Determine the desired pairing window (in hours) and set the pairing window by using the `PAIRING_WINDOW` command. The pairing window is the maximum allowable difference between the validtime of the forecast and the obstime of the observed in any forecast-observed data pair.
4. Determine the locations to use for the analysis. A location is defined by the location id (`LOCATION` command), physical element (`PE` command), duration (`DUR` command), extremum (`EXTREMUM` command), and forecast type source (`FCST_TS` command). Specify the desired locations by using the appropriate commands above accordingly. If a command is not specified, it is assumed to be "`ALL`", implying that it will not restrict what locations are paired. Note that any location that does not have a corresponding entry in the vfyruninfo table will not be included in pairing. Step 1 created the required entries.
5. Decide if persistence forecasts are to be generated and specify `PERSISTENCE` accordingly.

6. Decide if this execution of the pairing algorithm is to use observed data with a SHEF type that is 'R', or some type other than 'R'. Specify the OBS_TYPE command as "RAW" for 'R' data, or "PROCESSED" for data that is not 'R'.
7. Initiate pairing by calling the BUILD_PAIRS action.

8.0 Batch Commands

A batch command sets a parameter that is used by a batch action (see Section 8).

This section provides an alphabetical listing of all of the available batch commands. Acceptable values will be listed for each command, as well as the default if the command is not specified or if the command's token value is "<DEFAULT>". Note that all token values below are placed within quotes, although that may not be necessary (see item 4 in Section 6). The following are the batch commands:

DUR = "<DUR1>,<DUR2>,...,<DURn>"
Description: Defines a list of durations. Pairing will only be performed for locations which have a duration within this list.
Acceptable Values: A list of 1-character SHEF duration codes or "ALL" to allow for any duration. The list must be comma separated and, if you have spaces within the list, must be within double quotes.
Default Value: "ALL".

END_TIME = "<DATE>"
Description: Defines the end date/time for the pairing run. The end time can be absolute or can be relative to the current system time. Any data pair included in statistics calculation must have a valid time prior or equal to the end date/time.
Acceptable Values: If the date is absolute, then it must be of one of these formats:

- "CCYY-MM-DD" (assumes time of 23:59:59 that day),
- "CCYY-MM-DD hh:mm:ss",
- "CCYY-MM-DD hh:mm:ss TZC",
- "MMDDCCYY:hh".

If the date is relative, then it must of the following format:

"* [<+ or -> <quantity> <unit> <quantity> <unit> ...]".

Everything in '[ ]' is optional. The <quantity> must be a positive integer and the <unit> must be either "WEEKS" ("WEEK" or "WK"), "DAYS" ("DAY" or "DY"), or "HOURS" ("HOUR" or "HR").
Default Value: "*" (the current system time).

EXTREMUM = "<EXT1>,<EXT2>,...,<EXTn>"
Description: Defines a list of extremums. Pairing will only be performed for locations which have an extremum within this list.
Acceptable Values: A list of valid 1-character SHEF extremum codes or "ALL" to allow for any extremum. The list must be comma separated and, if you have spaces within the list, must be within double quotes.

Default Value: "ALL".

FCST_TS = "<TS1>,< TS1>,...,< TS1>"

Description: Defines a list of forecast type sources. Pairing will only be performed for locations which have a type source within this list.

Acceptable Values: A list of valid 2-character SHEF type source codes or "ALL" to allow for any type source. The list must be comma separated and, if you have spaces within the list, must be within double quotes.

Default Value: "ALL".

LOCATION = "<LID1>,<LID2>,...,<LIDn>"

Description: Defines a list of locations for which forecast-observed data pairing is desired.

Acceptable Values: A list of location ids (lids) or "ALL" to generate pairs for all lids (and physical elements) in the vfyruinfo table of the archive database. The list must be comma separated and, if you have spaces within the list, must be within double quotes. Any location in the list must have at least one entry in the vfyruninfo table or it will be ignored.

Default Value: "ALL"

OBS_TYPE = <RAW/PROCESSED>

Description: Defines if raw or processed tables will be searched for data to pair with forecasts. See Section 12.0 for further information about the raw and processed tables. Pairs constructed with a setting of "RAW" will be placed in vfypairs, and "PROCESSED" will be placed in vfyprocpairs.

Acceptable Values: Either "RAW" or "PROCESSED".

Default Value: "RAW"

NOTE: If the observed data to be used in pairing for a location has a SHEF type that is NOT *'R', then set this value to "PROCESSED" even if the value is normally posted via the archive raw decoder.*

PAIRING_WINDOW = <number of hours>

Description: Defines the maximum allowable difference between the validtime of the forecast and the obstime of the observation in a forecast-observed data pair. Also specifies the maximum allowable difference between the basistime of a forecast and the obstime of an observed for that observed to be considered as a persistence forecast (see PERSISTENCE command). This must be specified in hours.

Acceptable Values: Any positive integer value.

Default Value: "6"

PE = "<PE1>,<PE2>,...,<PEn>"

Description: Defines a list physical elemens. Pairing will only be performed for locations which have a physical element within this list.

Acceptable Values: A list 2-character SHEF physical element codes or "ALL" to allow for any valid physical element. The list must be comma separated and, if you have spaces within the list, must be within double quotes.

Default Value: "ALL".

`PERSISTENCE = <ON/OFF>`

Description: Determines if persistence forecast pairs are to be generated for the data pairs constructed. A persistence forecast is a forecast that uses the most recent observed value when the forecast was produced (the forecast basistime) for all lead times of the forecast. Within the IVP Batch Program, the obstime of the observed value that becomes the persistence forecast must be within the pairing window (see PAIRING_WINDOW command) of the forecast's basistime.

Acceptable Values: Either "ON" or "OFF".

Default Value: "ON"

> *NOTE: All persistence forecasts are give a forecast TS of "FR" in their vfypairs or vfyprocpairs records.*

`START_TIME = "<DATE>"`

Description: Defines the start date/time for the pairing run. The start date/time can be absolute or can be relative to the current system time. The START_TIME provides a lower bound on the valid time of a forecast that is to be paired with an observed value.

Acceptable Values: See END_TIME above, except that for the format of "CCYY-MM-DD" a time of 00:00:00 is assumed.

Default Value: "* - 14 DAYS" (two weeks prior to current system time).

9.0 Batch Actions

Actions instruct the verification program to do something. That nature of what is done depends on the action given.

The following are the batch actions:

`BUILD_PAIRS = <anything>`

Description: This action causes pairing to be done using the parameters defined before this action is called in the batch file.

Accptable Values: The value is ignored.

Default Value: Does not apply. The value is ignored.

10.0 Special Tokens

A special token is a batch language token that is not related to specifying parameters or performing actions related to the software program.

This section provides an alphabetical listing of all special tokens. The following are special tokens:

@FILE = <filename>

Description: Forces batch processor to process the file specified by `<filename>`. The file is treated just as if its contents were included within the original batch file.
Acceptable Values: Any valid file that will not force the batch processor to enter into an infinite, recursive loop (i.e. if file 1 references file 2, then file 2 can never reference file 1).
Default Value: Does not apply. The file name must be valid.

@+ = <line continuation>

Description: Line continuation token. The batch processor will take the value of this line and append it to the value on the previous line.
Acceptable Values: Depends upon previous lines batch command or action.
Default Value: Depends upon previous lines batch command or action.

11.0 The Pairing Algorithm

The algorithm used to construct the forecast-observed data pairs for verification is as follows:

1. For the location ids (lid), physical elements (pe), durations (dur), extremums (ext), and forecast type sources (ts) specified by the `LOCATION`, `PE`, `DUR`, `EXTREMUM`, and `FCST_TS` commands, load all of the rows from the vfyruninfo table of the archive database and process them to create verification locations, acquiring a sensor preference list for restricting observed type sources and for choosing between competing observed data values. If any observation's type source is not within a verification location's sensor preference list, it will not be used in pairing. If empty, the sensor preference list will not play a role in the pairing algorithm.
2. For each verification location, do the following:
   a. Load the forecast data from the appropriate table (see Section 12.0) for the verification location's lid, pe, dur, ext, and ts, between the start and end times specified in the batch file. The list is sorted in the program by validtime.
   b. Load the observed data from the appropriate raw or processed table, determined by the `OBS_TYPE` command, for the verification location's lid, pe, dur, ext, and sensor preference list, between the start and end times. The start time of the query is based on the user specified start time if no persistence forecasts are to be generated, or the smallest loaded forecast basistime (Step 2a) if persistence forecasts are to be generated. The list is sorted in the program by obstime. The dates involved are adjusted to allow for the pairing window around the forecast validtime. So, if the pairing window is 6 hours, the start time is adjusted backwards by 6 hours and the end time is adjusted forwards by 6 hours.

   NOTE: If the `OBS_TYPE` *is "*`PROCESSED`*, it is possible for two queries to be performed searching for observed data. The first query is of one of the three table pehpsep, pedpsep, and peqpsep, based on the duration and extremum of the verification location being paired. The second query will be of the peoosep table if the first query does not produce any observations.*

   c. For every forecast value loaded from the data base (Step 2a), do the following:
      i. Construct a list of candidate observed records (from the list in Step 2b) based on the pairing window. The search used to find the candidates is shortened using the fact that the observed data is sorted by obstime. Any data with a SHEF qualifier code (shef_qual_code) of 'B', 'E', or 'R' is not considered a candidate.

ii. Find the best candidate based on the shef_qual_code, sensor preference rank, and difference between the forecast validtime and observed obstime (which should be minimized), in that order. The order of preference for the shef_qual_code is as follows: G, M, P, V, S, Z, T, F, and Q.

iii. Construct the pair and add it to the list of data pairs.

d. If persistence forecasts are to be generated, then sort the pairs generated in the previous step by forecast basistime and for every pair, do the following:

i. Construct a list of candidate observed records (from the list in Step 2c) for the persistence forecast based on the pair's forecast basistime, using the pairing window. However, the observed obstime must be before the forecast basistime.

ii. Find the best candidate as is done in step 2d-ii above.

iii. Construct the persistence forecast pair, using the best observed record's obsvalue as the forecast value, "FR" as the forecast type source, the observed record's quality code as the forecast quality code, but all other fields are kept the same as the current working pair.

e. Insert or update the list of data pairs, one at a time, into the vfyprocpairs table if the OBS_TYPE command is 'PROCESSED', or the vfypairs table otherwise. Update is done if a pair is found with a matching primary key. The quality_code value for each inserted or updated pair will be set to 1879048191 (the default 'good' quality).

This algorithm will attempt to construct one persistence forecast pair for every generated forecast pair.

12.0 Archive Database

The pairing software accesses the following tables to acquire system and location parameters:

- Location: Used to acquire the RFC name for a location.
- Rivercrit: Used to acquire critical stages and flows for a verification location.
- Vfyruninfo: Used to identify the verification location's for which verification can be done.
- Vaddadjust: Used to acquire adjustment factors that will be applied to pairs prior to computing statistics.

The software accesses the following raw tables of the archive database:

- Forecast data: pedfsep, pehfsep, peqfsep, pedcsep.
- Observed data: pecrsep, pedrsep, peoosep.

The software accesses the following processed tables of the archive database:

- Observed data: pedpsep, pehpsep, peqpsep.

The raw tables are posted to by the archive raw decoder, while the processed tables are posted to by the archive processed decoders. To see how the decoders determine to which table to post data, look at the following document:

http://www.nws.noaa.gov/oh/rfcdev/docs/C2_datavalnames.pdf.

If processed data is posted to the archive database via the raw decoder, then it will be posted to the peoosep table. As such, when instructed to load observed data from the processed tables (see the OBS_TYPE command), it will search the peoosep table if nothing is found in the processed tables.

The two pairs tables posted to by the **IVP Batch Program** during pairing are:

- vfypairs: stores pair constructed from raw observed values.
- vfyprocpairs: stores pairs constructed from processed observed values.

In both cases, the quality_code field of the table determines if a pair is considered 'good' or 'bad'. Bad pairs are those pairs with a quality_code below the threshold 1073741824, with most having a quality code of 0. Good pairs usually have a quality code of 1879048191.

13.0 Examples

The following examples illustrate how to construct batch files to accomplish particular goals.

Example 1: A batch file to produce pairs for the past two weeks and for all locations within the vfyruninfo table, using the default pairing window of 6 hours.

```
START_TIME = "* - 14 days"
END_TIME   = "*"
Location   = "ALL"  # default is all, so this line can
                    # be left out.
Pairing_Window = 6  # default is 6, so this line can be left out.

# Build the pairs
BUILD_PAIRS = true
```

Example 2: A batch file to produce pairs for the month of May, 2007, and for various locations identified by location id and SHEF code, using the default pairing window of 6 hours and without generating persistence forecasts.

```
# Define pairing parameters for a run during May, 2007.
START_TIME      = "2007-05-01 00:00:00"
END_TIME        = "2007-05-31 23:59:59"
PERSISTENCE     = OFF

# Pair locations AAAAA and BBBBB, where the forecast type source is FF.
Location        = "AAAAA,BBBBB"
FCST_TS         = "FF"
BUILD_PAIRS = true

# Pair for all locations with an extremum of Z and type source of TA.
Location        = ALL
FCST_TS         = TA
Extremum        = Z
BUILD_PAIRS = true
```