

# *IVP Batch Program User's Manual For Pairing*

## *1.0 Overview*

The IVP Batch Program serves two functions:

- (1) constructing forecast-observed data pairs and
- (2) calculating verification statistics.

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

## *2.0 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 and physical elements to be included in data pairing. See the Vfyruninfo Editor User's Manual for more details.
2. Decide on the data 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 and set the locations using the `LOCATIONS` command. Note that any location that does not have a corresponding entry in the `vfyrinfo` table will not be included in pairing. Step 1 created the required entries.
5. Initiate pairing by calling the `BUILD_PAIRS` action.

## *3.0 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 **bold**.
- *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 **bold**.
- *data pair*: A forecast-observed pair, defined in the vfypairs table of the archive database.

## 4.0 Execution

Before executing the IVP Batch Program, be sure that the tables vfyruninfo, riverstat, 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 riverstat table must be populated in order for the flood stage to be found, given by the field fs, if the flood stage is used in determining categories. 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.

## 5.0 Apps-Defaults Tokens

In order to properly access the vfyruninfo table within the archive database, the user's environment must be setup to read from the archive database and access the java system code. This requires setting the apps-defaults token "adb\_name" to be the name for the archive database, making sure the INFORMIXSERVER is set to the correct server, and setting the apps-defaults token "sys\_java\_dir" to the correct value. This should be part of the national apps-defaults file.

Also, the following apps-defaults tokens are used by the IVP Batch Program (with their national file setting given):

- vsys\_input : \$(vsys\_dir)/input
- vsys\_output : \$(vsys\_dir)/output

All input batch files are assumed to exist within the directory corresponding to apps-defaults token "vsys\_input", unless specified otherwise. All output files will be generated in the directory corresponding to token "vsys\_output" and in a subdirectory corresponding to the user's LOGNAME environment variable. The directory will be created if it does not exist.

## 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. Any 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 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:

**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",
- "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:

"\* [ $\leftarrow$  or  $\rightarrow$  <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).

**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"

**OBSERVED\_TABLE = <table name>**

*Description:* Defines which archive database table is to be used to collect observed data: pecrsep or pehpsep.

*Acceptable Values:* Either "pecrsep" or "pehpsep".

*Default Value:* "pecrsep"

**PAIRING\_WINDOW = <number of hours>**

*Description:* Defines the maximum allowable difference between the validtime of the forecast and the obstime of the observed in a forecast-observed data pair. This must be specified in hours.

*Acceptable Values:* Any positive integer value.

*Default Value:* "6"

**PERSISTENCE = <ON/OFF>**

*Description:* Determines if persistence forecast pairs are to be generated for the data pairs constructed.

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

*Default Value:* "ON"

**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.

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

## 8.0 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.

*Acceptable Values:* The value is ignored.

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

## 9.0 The Pairing Algorithm

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

1. For the locations involved, load all of the entries from the `vfyrinfo` table of the archive database.
2. For each location and each physical element (pe) found in the `vfyrinfo` table for that location, do the following:
  - a. Process the entries for that location and pe from Step 1 into
    - (i) a list of forecast type sources to pair,
    - (ii) a list of acceptable extremums (currently fixed to “Z”), and
    - (iii) a sensor preference list for choosing between competing observed data values.

If any of these lists are empty (empty is denoted by a “\_” in the `vfyrinfo` table), then that list will play no role in querying for forecast/observed data from the database.

- b. Load the forecast data from the `pedfsep` table for the location, pe, forecast type sources, and acceptable extremums between the start and end times specified in the batch file. The list is sorted in the program by `validtime`.
- c. Load the observed data from the `OBSERVED_TABLE` table for the location, pe, and acceptable extremums between an adjusted 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 2b) 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.

- d. For every forecast value loaded from the data base (Step 2b), do the following:
  - i. Construct a list of candidate observed records (from the list in Step 2c) 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.
  - 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, E, T, F, Q, B, and R.
  - iii. Construct the pair and add it to the list of data pairs.
- e. 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 (the observed obstime must be before the forecast basistime, however).
  - 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, the observed record's quality code as the forecast quality code, but all other fields are kept the same as the current working pair.
- f. Insert or update the list of data pairs into the vfypairs table. Update is done if a pair is found with a matching vfypairs primary key.

## 10.0 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 October, 2003, and for the locations "AAAAA", "BBBBB", and "CCCCC", using a pairing window of 1 day with persistence forecasts turned off:

---

```
START_TIME = "2003-10-01 00:00:00"
END_TIME   = "2003-10-31 23:59:59"
Location   = "AAAAA, BBBBB, CCCCC"
Pairing_Window = 24 # Must be in hours.
Persistence = off
BUILD_PAIRS = anything
```

---