NOAA Technical Report NOS CS 16

# ODAAS' Real-time River Information Acquisition for NOS Estuarine Forecast Systems in the Middle-Atlantic Region

Meredith Westington
John G. W. Kelley

January 2003

noaa National Oceanic and Atmospheric Administration

U.S. DEPARTMENT
OF COMMERCE
Donald Evans, Secretary

National Oceanic and
Atmospheric Administration
Conrad C . Lautenbacher, Jr.,
VADM USN (Ret.), Under Secretary

National Ocean Service
Jamison S. Hawkins
Acting Assistant Administrator

Office of Coast Survey
Captain David MacFarland

Coast Survey Development
Laboratory
Bruce B. Parker

## NOTICE

Mention of a commercial company or product does not constitute an endorsement by NOAA. Use for publicity or advertising purposes of information from this publication concerning proprietary products or the tests of such products is not authorized.

NOAA Technical Report NOS CS 16

# ODAAS' Real-time River Information Acquisition for NOS Estuarine Forecast Systems in the Middle-Atlantic Region

Silver Spring, Maryland
January 2003

noaa National Oceanic and Atmospheric Administration

U.S. DEPARTMENT OF COMMERCE
National Ocean Service
Coast Survey Development Laboratory

## 9. SUMMARY AND THE FUTURE

In summary, the following flowchart depicts the process of receiving and archiving river information received from the MARFC.

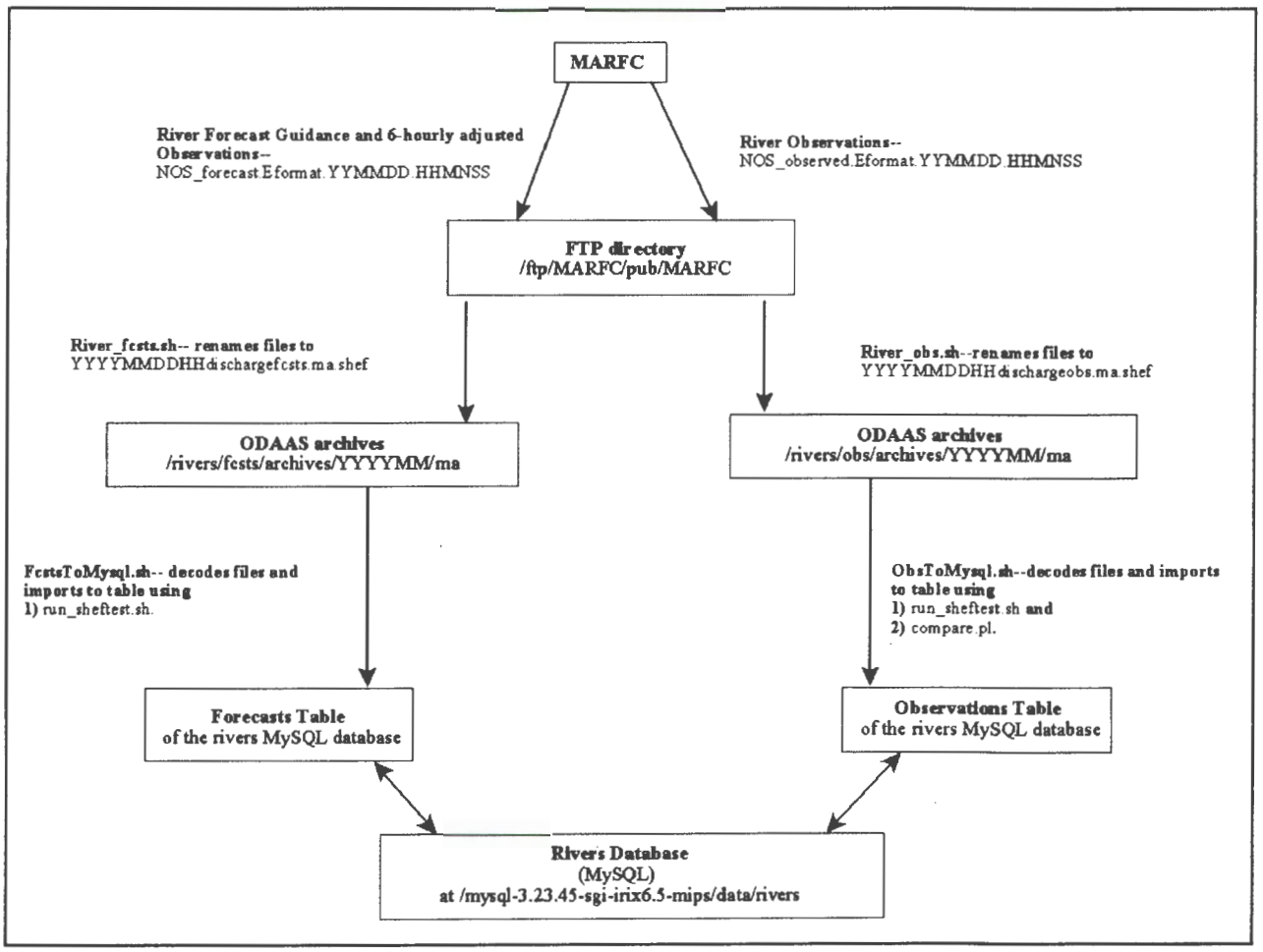


**Figure 9. 1.** Flowchart of river information from the NWS' MARFC, to the ODAAS archives, and to the MySQL rivers database tables, forecasts and observations. YYYY is the year, MM is the month, DD is the day, HH is the hour, MN is the minute, and SS is the second.

The improvements, changes, or additions that will be applied to ODAAS' system for ingesting river information are as follows:

a) Improve the process flow in `compare.pl` by fine-tuning the nested "if" statements within the query that checks for record matches in the database,
b) Obtain river observations and NWS' river forecast guidance for the Patapsco, Mattaponi, and Pamunkey Rivers from the MARFC to support CBOFS and for the Hudson and Hackensack Rivers from the Northeastern River Forecast Center (NERFC) located at Taunton, MA to support NYOFS (See Figure 9.2), and

NOAA Technical Report NOS CS 16

# ODAAS' Real-time River Information Acquisition for NOS Estuarine Forecast Systems in the Middle-Atlantic Region

Silver Spring, Maryland
October 2002

noaa National Oceanic and Atmospheric Administration

U.S. DEPARTMENT OF COMMERCE
National Ocean Service
Coast Survey Development Laboratory

Office of Coast Survey
National Ocean Service
National Oceanic and Atmospheric Administration
U.S. Department of Commerce

The Office of Coast Survey (CS) is the Nation's only official chartmaker. As the oldest United States scientific organization, dating from 1807, this office has a long history. Today it promotes safe navigation by managing the National Oceanic and Atmospheric Administration's (NOAA) nautical chart and oceanographic data collection and information programs.

There are four components of CS:

The Coast Survey Development Laboratory develops new and efficient techniques to accomplish Coast Survey missions and to produce new and improved products and services for the maritime community and other coastal users.

The Marine Chart Division collects marine navigational data to construct and maintain nautical charts, Coast Pilots, and related marine products for the United States.

The Hydrographic Surveys Division directs programs for ship and shore-based hydrographic survey units and conducts general hydrographic survey operations.

The Navigation Services Division is the focal point for Coast Survey customer service activities, concentrating predominantly on charting issues, fast-response hydrographic surveys and Coast Pilot updates.

NOAA Technical Report NOS CS 16

# ODAAS' Real-time River Information Acquisition for NOS Estuarine Forecast Systems in the Middle-Atlantic Region

Meredith Westington
John G. W. Kelley

October 2002

noaa National Oceanic and Atmospheric Administration

U.S. DEPARTMENT
OF COMMERCE
Donald Evans, Secretary

National Oceanic and
Atmospheric Administration
Conrad C . Lautenbacher, Jr.,
VADM USN (Ret.), Under Secretary

National Ocean Service
Jamison S. Hawkins
Acting Assistant Administrator

Office of Coast Survey
Captain David MacFarland

Coast Survey Development
Laboratory
Bruce B. Parker

## TABLE OF CONTENTS

LIST OF FIGURES ..... v
LIST OF TABLES ..... v
LIST OF ACRONYMS ..... vi
ABSTRACT ..... vii
1. INTRODUCTION ..... 1
1.1. Purpose ..... 1
1.2. Importance of Hydrologic Information in Estuarine Modeling ..... 1
2. REQUIREMENTS OF ESTUARINE PREDICTION SYSTEMS ..... 3
2.1. Nowcast Cycle ..... 3
2.2. Forecast Cycle ..... 4
3. RIVER DATA ACQUISITION PROCESS ..... 5
3.1. Introduction ..... 5
3.2. Available stations ..... 5
3.3. Filename convention ..... 7
4. SHEF ENCODED FILES ..... 9
4.1. Introduction ..... 9
4.2. River Observations File ..... 9
4.3. River Forecast Guidance and Adjusted Observations File ..... 10
5. THE SHEF DECODER ..... 13
5.1. Introduction ..... 13
5.2. Decoded River Observations File ..... 13
5.3. Decoded River Forecast Guidance and Adjusted Observations File ..... 14
6. OVERVIEW OF ODAAS ..... 17
6.1. ODAAS Directories ..... 17
6.2. Scripts ..... 17
6.3. Scheduling ..... 19
6.4. Log Files ..... 20
7. OVERVIEW OF MYSQL ..... 21
7.1. Introduction ..... 21
7.2. Database Structure ..... 21
7.3. Database Entry ..... 24
7.4. Navigating Through MySQL ..... 25
8. MONITORING PROCEDURES ..... 27
9. SUMMARY AND THE FUTURE ..... 29
10. ACKNOWLEDGMENTS ..... 31
REFERENCES ..... 31

APPENDIX A. CUSTOMIZING THE SHEF DECODER ..... 33
APPENDIX B. DETAILS OF COMPARE.PL..... 37
APPENDIX C. INSTALLING AND CONFIGURING MYSQL WITH PERL DBI ..... 41
APPENDIX D. STEPS FOR ADDING NEW STATIONS TO THE RIVERS DATABASE .... 45

## LIST OF FIGURES

Figure 2. 1. Names and locations of rivers for which streamflow observations and forecast guidance are needed by NOS' CBOFS and NYOFS ..... 4
Figure 3. 1. Locations of the nine river stations at which the NWS' MARFC provides observations and forecast guidance to NOS ..... 6
Figure 4. 1. Portion of MARFC's SHEF encoded, river observations file ..... 10
Figure 4. 2. Portion of MARFC's SHEF encoded, river forecast file ..... 11
Figure 5. 1. Portion of the SHEF decoded, output of file ..... 14
Figure 5. 2. Portion of the SHEF decoded output of file ..... 15
Figure 9. 1. Flowchart of river information from the NWS' MARFC, to the ODAAS archives, and to the MySQL rivers database tables, forecasts and observations. ..... 29
Figure 9. 2. The names and locations of outstanding rivers for which streamflow observations and forecast guidance are still needed by NOS' CBOFS and NYOFS. ..... 30
Figure A. 1. The original `shpout.f` program packaged with the default SHEF software. ..... 33
Figure A. 2. Changed portion of the `shpout.f` program ..... 34
Figure A. 3. The run_sheftest script was modified to reflect its use and directory location. Note that the changes are highlighted, and this file is the `run_sheftest` routine for the river observations. ..... 35
Figure B. 1. The `compare.pl` script ..... 37

## LIST OF TABLES

Table 2. 1. Real-time river observation requirements for NOS' nowcast and forecast systems. ..... 3
Table 3. 1. The nine river stations listed by NWS station ID, USGS station ID, station name, state, and geographic position. ..... 6
Table 6. 1. Name and Location of ODAAS' River Observations Files. ..... 18
Table 6. 2. Name and Location of ODAAS' River Forecasts and 6-hourly Adjusted Observations Files. ..... 19
Table 6. 3. Location, names, launch times, and sleeping characteristics of the river observations scripts. ..... 20
Table 6. 4. Location, names, launch times, and sleeping characteristics of the river forecasts scripts. ..... 20
Table 7 1. Descriptions of the fields used for the stations, observations, and forecasts tables. ..... 22
Table 7 2. Stations table format of MySQL `rivers` database.. ..... 23
Table 7 3. Observations table format of MySQL `rivers` database. ..... 23
Table 7 4. Forecasts table format of MySQL `rivers` database.. ..... 24

## LIST OF ACRONYMS

| | |
|---|---|
| CBOFS | Chesapeake Bay Operational Forecast System |
| CFS | Cubic Feet per Second |
| COFS | Coastal Ocean Forecast System |
| CSDL | Coast Survey Development Laboratory |
| DBD | Database Driver |
| DBI | Database Interface |
| FTP | File Transfer Protocol |
| GBOFS | Galveston Bay Operational Forecast System |
| MARFC | Middle Atlantic River Forecast Center |
| MLLW | Mean Lower Low Water |
| MMAP | Marine Modeling and Analysis Programs |
| NCEP | National Centers for Environmental Prediction |
| NERFC | Northeast River Forecast Center |
| NOAA | National Oceanic and Atmospheric Administration |
| NOS | National Ocean Service |
| NWS | National Weather Service |
| NYOFS | New York Operational Forecast System |
| ODAAS | Operational Data Acquisition and Archive System |
| PID | Process Identification |
| RDBMS | Relational Database Management System |
| RFC | River Forecast Center |
| QPF | Quantitative Precipitation Forecast |
| SGI | Silicon Graphics, Inc. |
| SHEF | Standard Hydrometeorological Exchange Format |
| SQL | Structured Query Language |
| USGS | United States Geological Survey |
| WGRFC | Western Gulf River Forecast Center |

## ABSTRACT

The National Ocean Service's Marine Modeling and Analysis Programs (MMAP) branch of the Coast Survey Development Laboratory develops real-time estuarine nowcast/forecast systems for predicting water levels, currents, salinity, temperature, and other oceanographic parameters at the U.S. seaports, estuaries, coastal ocean, and Great Lakes. To produce accurate, high-resolution nowcasts and forecasts of estuarine conditions, these prediction systems need to incorporate real-time, meteorological, oceanographic, and hydrological observations and forecasts.

River information, necessary to fulfill the hydrological component of MMAP's accurate, high-resolution prediction models, was recently acquired via NOS' Operational Data Acquisition and Archive System (ODAAS) for the Middle-Atlantic region. The United States Geological Survey (USGS) and NOAA's National Weather Service (NWS) are responsible for providing river flow information to the nation. A real-time reporting system for the national stream gauging network is maintained by the USGS, and the daily river forecast guidance is produced by NWS using the NWS' River Forecast System. In the past, MMAP's acquisition of river information had been an individual effort among the modelers. Recently, a more comprehensive approach has been developed and incorporated into ODAAS to ensure that coastal and estuarine model developers have uniform access to real-time stream flow observations and river model streamflow forecast guidance.

Presently, the development has focused on obtaining river stream flow observations and forecast guidance for major rivers flowing into estuaries in the Middle-Atlantic region. It is anticipated that this same approach will be expanded to other major estuaries and the Great Lakes.

# 1. INTRODUCTION

## 1.1. Purpose

Real-time, estuarine nowcast/forecast systems are developed by the National Ocean Service's Marine Modeling and Analysis Programs (MMAP) branch of the Coast Survey Development Laboratory in order to provide products and services to the coastal marine community and to support safe navigation. After development and final testing, the estuarine prediction models are transferred to NOS' Center for Operational Oceanographic Products and Services (CO-OPS) for operational implementation. CO-OPS provides accurate, high-resolution analyses and short-term forecast guidance of oceanographic variables such as water levels, currents, salinity, and temperature. This guidance is designed to benefit commercial, military, and recreational mariners, the National Weather Service's (NWS') marine forecasters, and other members of the marine community.

MMAP's prediction systems are designed to accurately predict water levels, currents, salinity, temperature, and other oceanographic variables within the United States' seaports, estuaries, coastal ocean, and Great Lakes. Three types of real-time information are necessary for the development of estuarine prediction systems:

(1) oceanographic observations and model forecast guidance,
(2) meteorological observations and model forecast guidance, and
(3) hydrological observations and model forecast guidance.

This technical report describes the data acquisition method that was developed and implemented in NOS' Operational Data Acquisition and Archive System (ODAAS) to acquire real-time river discharge observations and forecast guidance for the Middle-Atlantic Region of the United States (Kelley et al., 2001). More detailed information about the history of this project and the importance of river stream flow information for estuarine prediction systems is available in the technical report, "Acquisition Status of Real-Time River Observations and River Forecast Guidance for Use in NOS Estuarine Nowcast/Forecast System Development" by Thomson (2000).

## 1.2. Importance of Hydrologic Information in Estuarine Modeling

Accurate streamflow or discharge information can be critical when analyzing and predicting oceanographic conditions, especially during extreme weather situations such as flooding or long term drought conditions. Model variables such as water level, salinity, temperature, and currents can be affected by the amount of freshwater flowing into a port or estuary.

To meet the need for hydrologic information, estuarine prediction models have traditionally utilized daily inflows or monthly climatological streamflow data. As prediction models have been refined over time, the importance of near real-time river information has increased. Because there was no comprehensive and centralized approach to acquiring real-time river information at MMAP, model developers had to rely on individual methods for obtaining observations and forecasts for their study areas. In response to this need, a river data acquisition system was added to ODAAS to acquire real-time river discharge observations and NWS' river discharge forecast guidance for the Middle-Atlantic Region of the United States.

## 2. REQUIREMENTS OF ESTUARINE PREDICTION SYSTEMS

### 2.1. Nowcast Cycle

The nowcast cycle of NOS nowcast/forecast systems need near real-time, observed river discharge. To create accurate nowcasts, the models rely on river information that meet three required guidelines. First, the models need to know the amount of water flowing into an estuary; therefore, the observations must contain discharge, which is defined as the volumetric rate of flow in thousands of cubic feet per second (cfs), and not stage height, which is defined as the water level surface referenced to a particular datum. Joseph Ostrowski at the MARFC notes,

> Realize that problems are introduced…with the indirect measurements of discharge. Since stage is observed and the discharges are computed using static rating curves, you never are truly certain of the accuracy of the conversion. The USGS uses a 10% error as their standard answer for the accuracy of their rating curves. And, based on location, some translations are worse than others (personal communication, October 26, 2000).

Second, the data must be received in real-time or near real-time on an hourly to daily basis. The following table shows the variables required for the nowcast cycle, frequency, forecast horizon, and timeliness at which the variables are needed (See Table 2.1).

**Table 2. 1.** Real-time river observation requirements for NOS' nowcast and forecast systems.

| Prediction Cycle | Required Variables | Frequency | Forecast Horizon | Timeliness |
|---|---|---|---|---|
| Nowcast | 1)Estimated Discharge<br>2)Water Temperature<br>3)Salinity | 1 hour | NA | 10 to 20 minutes past the top of the hour |
| Forecast | 1)Discharge<br>2)Water Temperature<br>3)Salinity | 3-6 hours | 48 hours | 2 times per day |

The third requirement of accurate nowcasts refers to the location of river gages in relation to the study area. Ideally, river stations should be positioned at points along each tributary as close as possible to the mouth of the estuary or port, but upstream of any tidal influences. Because an estuarine model needs an estimate of freshwater input for as much of the river's watershed as possible, the location of river gages is critical. It is important to be upstream of tidal influences because a tidal influence makes it impossible to estimate the freshwater flow separate from the tidally-induced flow. Occasionally, it is necessary for models to include additional river stations along tributaries of a main stream, if the major tributaries enter the stream below the gage (Thomson, 2000). Figure 2.1 depicts the names of the rivers for which river discharge observations are needed for the nowcast cycles of NOS' Chesapeake Bay Operational Forecast System (CBOFS) and the New York Operational Forecast System (NYOFS) for the Port of New York and New Jersey.

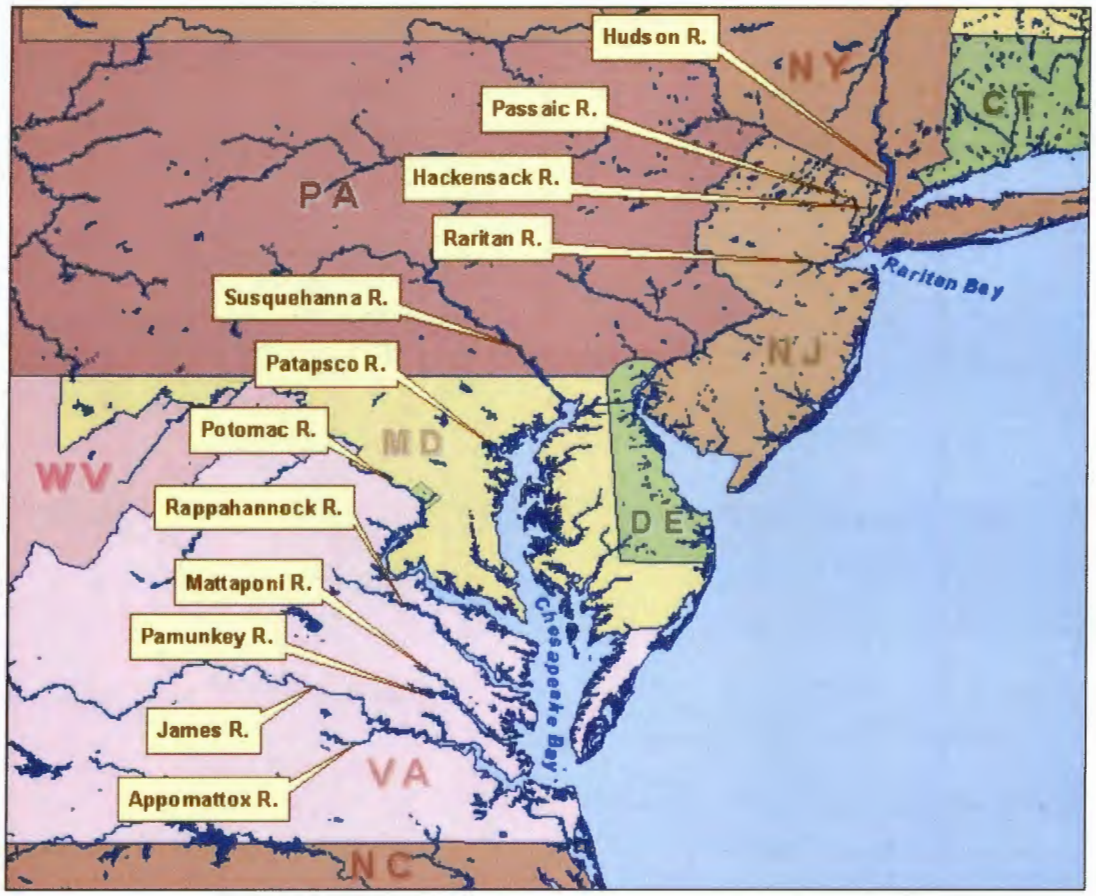


**Figure 2. 1.** Names and locations of rivers from which streamflow observations and forecast guidance are needed by NOS' CBOFS and NYOFS.

## 2.2. Forecast Cycle

The forecast cycle of NOS nowcast/forecast systems require river discharge forecasts. Forecast periods vary by location and are determined by the NWS' River Forecast Center (RFC) responsible for issuing the forecast guidance using the NWS' River Forecast System. The minimum forecast duration of NOS models is 24 hours and all points which forecast on a daily basis meet this minimum requirement. For estuarine models that extend beyond 24 hours, longer term forecast information is required. Another forecast requirement is the need for information to be available at intervals of 3 to 6 hours (See Table 2.1).

Similar to nowcast needs, the location and number of forecast sites are important for the forecast cycles of NOS prediction systems. For estuarine forecasting, the most useful forecast is at the point farthest downstream of a tributary and closest to the port or estuary. Typically, there are fewer river sites where forecast information is available as compared to observational data. The amount of available forecast information depends upon the weather conditions. For example, during flood conditions, stage and discharge forecast guidance is generated at additional sites. The rivers from which streamflow observations and forecast guidance are needed for forecasting estuarine conditions in the Chesapeake Bay and Port of New York/New Jersey are depicted in Figure 2.1.

## 3. RIVER DATA ACQUISITION PROCESS

### 3.1. Introduction

On December 1, 2000, the National Weather Service's Middle Atlantic River Forecast Center (MARFC) in State College, PA began providing river discharge observations and forecast guidance to the National Ocean Service. The MARFC created a discharge generation system which provides daily files containing USGS river gage information and NWS river stream flow forecast guidance from NWS' River Forecast System at points along tributaries flowing into the Chesapeake, Delaware, and Raritan Bays. This information is incorporated into two encoded files: (1) river observations and (2) river model forecast guidance and 6-hourly adjusted values. These two files are encoded using NWS' Standard Hydro-meteorological Exchange Format (SHEF).

Arriving every 4 hours, the river observations file contains USGS observations of stage and estimated stream flow for the past 24 hours at a 1 hour frequency. The streamflow is estimated by MARFC from the observed stage and latest rating curves.

Arriving once a day at 1600 GMT, the river forecast file contains NWS MARFC river forecast guidance of stream flow for the next 48 hours at a 6 hour frequency. The forecast discharge values incorporate 48 hours of quantitative precipitation forecast (QPF). This file also contains adjusted discharges for the previous 24 hours. Adjusted discharges are the discharge values resulting from the MARFC's modeling revisions. Sometimes, stage readings are incorrect due to a bad gage or ice effects in winter, which lead to incorrect discharge values (J. Ostrowski, personal communication, October 12, 2000). The MARFC provides these adjusted values to be used in cases when there is a conflict with observed data values; the adjusted values are presumed to be more accurate.

### 3.2. Available stations

The MARFC provides observations and forecast guidance at USGS river gages to CSDL for use by NOS estuarine nowcast/forecast systems in the Chesapeake Bay and Port of New York/New Jersey (Raritan Bay) and for the NOAA Coastal Ocean Forecast System (COFS). The nine available stations at the MARFC are:
Passaic River at Little Falls, NJ; Raritan River at Bound Brook, NJ; Delaware River at Trenton, NJ; Susquehanna River at Marietta, PA; Schuylkill River at Philadelphia, PA; Potomac River at Little Falls, MD; Rappahannock River at Fredericksburg, VA; James River at Richmond, VA; and Appomattox River at Matoaca, VA.

The following figure depicts the nine locations of the river stations provided to ODAAS by the MARFC, and Table 3.1 indicates the station ID, station name, state, and geographic coordinates.

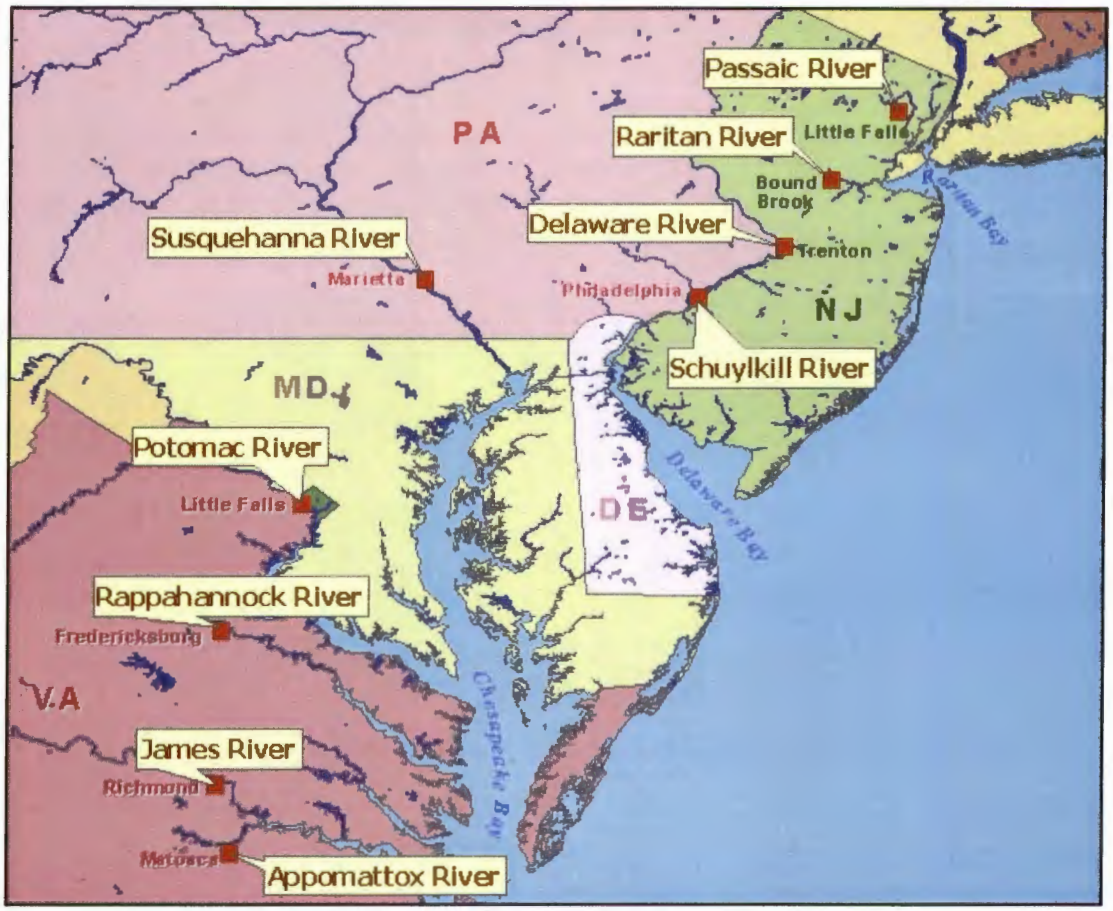


**Figure 3. 1.** Locations of the nine river stations at which the NWS' MARFC provides observations and forecast guidance to NOS. Station details are provided in Table 3.1.

**Table 3.1.** The nine river stations listed by NWS station ID, USGS station ID, station name, state, and geographic coordinates.

| NWS Station ID | USGS Station ID | Station Name | State | Latitude (deg) | Longitude (deg) |
|---|---|---|---|---|---|
| LTFN4 | 01389500 | Passaic River at Little Falls | NJ | 40.88472 | -74.22638 |
| BDKN4 | 01403000 | Raritan River Below Calco Dam at Bound Brook | NJ | 40.55000 | -74.55138 |
| TREN4 | 01463500 | Delaware River at Trenton | NJ | 40.22166 | -74.77833 |
| MRTP1 | 01576000 | Susquehanna River at Marietta | PA | 40.05444 | -76.53111 |
| PADP1 | 01474500 | Schuylkill River at Philadelphia | PA | 39.96777 | -75.18888 |
| BRKM2 | 01646500 | Potomac River at Little Falls near Washington | DC | 38.94972 | -77.12749 |
| FEDV2 | 01668000 | Rappahannock River at Fredericksburg | VA | 38.32222 | -77.51805 |
| RMDV2 | 02037500 | James River near Richmond | VA | 37.56305 | -77.54722 |
| MTCV2 | 02041650 | Appomattox River at Matoaca | VA | 37.22499 | -77.47555 |

### 3.3. Filename convention

MARFC transmits files containing river observations and forecast guidance to a public FTP directory on a restricted NOS/CSDL data server. The observations and forecasts in these files are encoded in SHEF. Files located under this directory are given the following naming convention:

(1) River observations are `NOS_observed.Eformat.YYMMDD.HHMNSS`, where YY is the two-digit year, MM is the month, DD is the day, HH is the hour, MN is the minute, and SS is the second of the delivery time.

(2) River forecasts are `NOS_forecast.Eformat.YYMMDD.HHMNSS`, where YY is the two-digit year, MM is the month, DD is the day, HH is the hour, MN is the minute, and SS is the second of the delivery time.

The "Eformat" refers to a specific type of SHEF encoded format, which is described further in Section 4.1.

From this FTP directory, the observations and forecasts files are renamed and moved to permanent ODAAS archive directories under `/rivers/obs/archives/YYYYMM/ma` and `/rivers/fcsts/archives/YYYYMM/ma`, respectively. The directory name `ma` refers to the Middle Atlantic region and YYYYMM refers to a concatenation of the 4-digit year and 2-digit month. The MARFC files are renamed using the following filename convention:

(1) River observations are renamed to `YYYYMMDDHHdischargeobs.ma.shef`, where YYYY is the 4-digit year, MM is the month, DD is the day, and HH is the hour.

(2) River forecasts are renamed to `YYYYMMDDHHdischargefcsts.ma.shef`, where YYYY is the 4-digit year, MM is the month, DD is the day, and HH is the hour.

After the files are renamed and redirected to the ODAAS archives, they are SHEF decoded, parsed, and added to a database for future ingests by the estuarine prediction models.

# 4. SHEF ENCODED FILES

## 4.1. Introduction

NWS' Standard Hydro-meteorological Exchange Format (SHEF) was created as a standard for the coding of real-time data (Office of Hydrology, 1998). The MARFC provides the river observations and forecast guidance to NOS in SHEF. In addition, NOAA's NWS provides SHEF parsing software to be used in the decoding process. Extensive online documentation is available for the NWS' SHEF decoding software at http://hydrology.nws.noaa.gov/oh/hrl/shef/version_1.3/index.htm.

There are three formats available with SHEF—the .A format for single station, multiple parameter data; the .B format for multiple station, multiple parameter, header driven data; and the .E format for single station, single parameter, and evenly spaced time series data. The MARFC has selected the .E format as a means to disseminate river data to NOS. The next two sections provide a detailed description of the two SHEF encoded files NOS receives from the MARFC.

## 4.2. River Observations File

Figure 4.1 depicts the top portion of an .E format, encoded, river observations file, `NOS_observed.Eformat.YYMMDD.HHMNSS`. The first two lines of the file are NWS' communication headers and are not meaningful for our purposes. To prevent parsing by the SHEF decoding software, text blocks are commented out using ":" symbol. The next 10 lines identify the purpose of the data as hourly discharges from points feeding the Chesapeake, Delaware, and Raritan Bays for use by NOS' nowcast/forecast systems, the provider as the NWS' MARFC, and the file creation date and time as November 17, 2001 at 2030 GMT. The next 7-line block contains descriptive information and data for a particular station. The first two lines of this block describe the station, units of measurement, and timeliness of the data. The first station is identified as along the "Delaware River at Trenton, NJ." The second line is described as follows: "TREN4" is the NWS' station ID, "20011116 Z DH20" indicates that values were observed on November 16, 2001 beginning at 2000 GMT, "DC200111172030" indicates that the file was created on November 17, 2001 at 2030 GMT, "DUE" indicates that the following data values are expressed in English units of thousands of cfs, "QRIRZ" indicates that the following data is instantaneous discharge computed from observed stage, and "DIH01" indicates that the hourly interval between each observation is 1 hour. The ".E" character denotes a line continuation and the encoded format type. The next five lines beginning at ".E1" and ending at ".E5" show each observed data value in thousands of cfs. Each value is separated by "/". The "M" character denotes a missing data value, and when decoded these values become equivalent to "-9999". Data from each of 9 stations is sent from the MARFC. Each station is formatted in the same manner as described above and is appended one after the other in this SHEF encoded, river observations file.

```
TTAA00 CCCC 172030
RHARRMRHA
:
: ESTIMATED HOURLY DISCHARGES FOR POINTS FEEDING
:  CHESAPEAKE, DELAWARE, AND RARITAN BAYS
: FOR USE BY NOS ESTUARINE NOWCAST/FORECAST SYSTEMS
:  AND NOAA COASTAL OCEAN FORECAST SYSTEM
:
: PROVIDED BY NWS MIDDLE ATLANTIC RIVER FORECAST CENTER
:   CREATED 2030 GMT ON 11/17/2001
:
:
: Delaware River at Trenton, NJ
.E TREN4   20011116 Z DH20/DC200111172030/DUE/QRIRZ/DIH01
.E1 /  2.6/  2.6/  2.6/  2.6/  2.6/   M
.E2 /  2.7/  2.7/  2.7/  2.7/  2.7/  2.7
.E3 /  2.7/  2.7/  2.7/  2.7/  2.7/  2.7
.E4 /  2.7/  2.7/  2.7/  2.7/  2.7/  2.7
.E5 /   M
```

**Figure 4.1.** Portion of MARFC's SHEF encoded, river observations file.

## 4.3. River Forecast Guidance and Adjusted Observations File

Figure 4.2 depicts a portion of an encoded, river forecast file, `NOS_forecast.Eformat.YYMMDD.HHMNSS`, with two types of data: forecast guidance and adjusted observations. As described in the river observations file, the first two lines of the file are NWS' communication headers and are not meaningful for our purposes. Following the communication header is an 11-line block, which provides information about the file. The block identifies the type of data as adjusted 6-hourly discharges and 6-hourly discharge forecast guidance, the location of information as points feeding the Chesapeake, Delaware, and Raritan Bay, the purpose as for use by NOS' nowcast/forecast models, the data provider as the NWS' MARFC, and the file creation date and time as November 17, 2001 at 1600 GMT. The next 3-line block contains descriptive information and data for a particular station. The first two lines of this block describe the station, units of measurement, and timeliness of the data. The first station is identified as along the "Delaware River at Trenton, NJ." The second line is described as follows: "TREN4" is NWS' station ID, "20011116 Z DH12" indicates that values were observed on November 16, 2001 beginning at 1200 GMT, "DC200111171600" indicates that the file was created on November 17, 2001 at 1600 GMT, "DUE" indicates the following data values are expressed in English units, "QRIRX" indicates that the file contains adjusted observed values, and "DIH06" indicates the hour interval between each adjusted observation is 6 hours for the past 24 hours. The ".E" character denotes a line continuation and encoded format type. The next line beginning at ".E1" shows each adjusted observed data value in thousands of cfs. Each value is separated by "/". Data values from each of 9 stations are sent from the MARFC. Each station is formatted in the same manner as described above and is appended one after the other in this SHEF encoded, river forecasts file.

Following the adjusted observations is another 3-line block of forecast values for the same station. As with the adjusted observations block, the first 2 lines describe the station, units of measurement, and timeliness of the data. Again, the first station is identified as along the "Delaware River at

Trenton, NJ." The second line is described as follows: "TREN4" is the NWS' station ID, "20011117 Z DH18" indicates that values were forecasted on November 17, 2001 beginning at 1800 GMT, "DC200111171601" indicates that the file was created on November 17, 2001 at 1601 GMT, "DUE" indicates the following data values are expressed in the English units cubic feet per second, "QRIFA" indicates that the file contains instantaneous river forecast discharge out to 48 hours, where the forecast discharges incorporate 48 hours of QPF, and "DIH06" indicates the hour interval between each value is 6 hours. The next line beginning at ".E1" shows each forecast guidance value in thousands of cfs. Each value is separated by "/" and each of the 9 stations are appended to each other in the same manner as described above to create this SHEF encoded, river forecasts file.

```
TTAA00 CCCC 171600
RHARRMRHA
:
: ADJUSTED 6-HOURLY DISCHARGES AND 6-HOURLY DISCHARGE
:  FORECAST GUIDANCE FOR POINTS FEEDING
:  CHESAPEAKE, DELAWARE, AND RARITAN BAYS
: FOR USE BY NOS ESTUARINE NOWCAST/FORECAST SYSTEMS
:  AND NOAA COASTAL OCEAN FORECAST SYSTEM
:
: PROVIDED BY NWS MIDDLE ATLANTIC RIVER FORECAST CENTER
:   CREATED 1600 GMT ON 11/17/2001
:
:
: Delaware River at Trenton, NJ
.E TREN4    20011116 Z DH12/DC200111171600/DUE/QRIRX/DIH06
.E1 /  2.6/  2.6/  2.6/  2.7/  2.7
:
: Delaware River at Trenton, NJ
.E TREN4    20011117 Z DH18/DC200111171601/DUE/QRIFA/DIH06
.E1 /  2.7/  2.7/  2.6/  2.6/  2.7/  2.7/  2.7/  2.7
```

**Figure 4.2.** Portion of MARFC's SHEF encoded, river forecast file.

# 5. THE SHEF DECODER

## 5.1. Introduction

Using NWS' SHEF parsing software the observations and forecast guidance are reformatted for easier inspection and ingest to a MySQL database. The parsing software is installed on an SGI workstation, under the ODAAS directory `/tools/decoders/shefpars` (alias *`shefpars`*). To utilize the SHEF parsing routines, some modifications were made. For example, the script `run_sheftest` was redirected to appropriate local directories. Then, the script was moved, along with a necessary parameter file called `SHEFPARM`, to local observations or forecasts script directories. The following sections are detailed descriptions of files containing decoded observations and forecasts from the MARFC.

## 5.2. Decoded River Observations File

Figure 5.1 depicts a portion of SHEF decoded output for river observations, `2002010320dischargeobs.ma.shef`. The SHEF parsing software was modified (see Appendix A) to create a comma delimited text file. The first line of the file is defined as follows "TREN4" is the station id, "20020102" is the observation date identified as January 2, 2002, "200000" is the observation time identified as 20:00:00 GMT, "20020103" is the file delivery date identified as January 3, 2002, "203000" is the delivery time identified as 20:30:00 GMT, "QR" is the physical element code referring to the value of river discharge, "R" and the following "Z" refer to the type and source values of nonspecific observed reading, "Z" is extremum code valued as null character (no value assigned), "3.000" is the observed data value in thousands of cfs, "Z" is the data qualifier, "-1.00" is the probability code defined as a real number, "0" is the encoded duration code, "0" is the revision code, " " is the source code (left blank for possible future inclusion), and "1" is the time series indicator. The observed data values are collected at 1 hour intervals out to 24 hours.

For more detailed information about the physical element codes for the SHEF decoder, see Table 1 of the SHEF manual (Office of Hydrology, 1998).

```
TREN4  ,20020102,200000,20020103,203000,QR,R,Z,Z,   3.000,Z, -1.00,   0, 0,      ,1
TREN4  ,20020102,210000,20020103,203000,QR,R,Z,Z,   3.000,Z, -1.00,   0, 0,      ,2
TREN4  ,20020102,220000,20020103,203000,QR,R,Z,Z,   3.000,Z, -1.00,   0, 0,      ,2
TREN4  ,20020102,230000,20020103,203000,QR,R,Z,Z,   3.000,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,000000,20020103,203000,QR,R,Z,Z,   3.000,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,010000,20020103,203000,QR,R,Z,Z,   3.100,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,020000,20020103,203000,QR,R,Z,Z,   3.100,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,030000,20020103,203000,QR,R,Z,Z,   3.200,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,040000,20020103,203000,QR,R,Z,Z,   3.200,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,050000,20020103,203000,QR,R,Z,Z,   3.200,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,060000,20020103,203000,QR,R,Z,Z,   3.100,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,070000,20020103,203000,QR,R,Z,Z,   3.000,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,080000,20020103,203000,QR,R,Z,Z,   2.900,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,090000,20020103,203000,QR,R,Z,Z,   2.900,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,100000,20020103,203000,QR,R,Z,Z,   2.800,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,110000,20020103,203000,QR,R,Z,Z,   2.700,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,120000,20020103,203000,QR,R,Z,Z,   2.700,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,130000,20020103,203000,QR,R,Z,Z,   2.700,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,140000,20020103,203000,QR,R,Z,Z,   2.600,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,150000,20020103,203000,QR,R,Z,Z,   2.600,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,160000,20020103,203000,QR,R,Z,Z,   2.700,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,170000,20020103,203000,QR,R,Z,Z,   2.700,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,180000,20020103,203000,QR,R,Z,Z,   2.700,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,190000,20020103,203000,QR,R,Z,Z,   2.800,Z, -1.00,   0, 0,      ,2
TREN4  ,20020103,200000,20020103,203000,QR,R,Z,Z, -9999.000,Z, -1.00,   0, 0,      ,2
PADP1  ,20020102,200000,20020103,203000,QR,R,Z,Z,   0.800,Z, -1.00,   0, 0,      ,1
PADP1  ,20020102,210000,20020103,203000,QR,R,Z,Z,   0.800,Z, -1.00,   0, 0,      ,2
PADP1  ,20020102,220000,20020103,203000,QR,R,Z,Z,   0.800,Z, -1.00,   0, 0,      ,2
PADP1  ,20020102,230000,20020103,203000,QR,R,Z,Z,   0.700,Z, -1.00,   0, 0,      ,2
PADP1  ,20020103,000000,20020103,203000,QR,R,Z,Z,   0.700,Z, -1.00,   0, 0,      ,2
PADP1  ,20020103,010000,20020103,203000,QR,R,Z,Z,   0.700,Z, -1.00,   0, 0,      ,2
PADP1  ,20020103,020000,20020103,203000,QR,R,Z,Z,   0.600,Z, -1.00,   0, 0,      ,2
PADP1  ,20020103,030000,20020103,203000,QR,R,Z,Z,   0.500,Z, -1.00,   0, 0,      ,2
PADP1  ,20020103,040000,20020103,203000,QR,R,Z,Z,   0.500,Z, -1.00,   0, 0,      ,2
PADP1  ,20020103,050000,20020103,203000,QR,R,Z,Z,   0.400,Z, -1.00,   0, 0,      ,2
```

**Figure 5.1**. Portion of the SHEF decoded, output of file `2002010320dischargeobs.ma.shef`.

## 5.3. Decoded River Forecast Guidance and Adjusted Observations File

Figure 5.2 depicts a portion of SHEF decoded output for river forecast guidance and adjusted observations. The SHEF software was modified (see Appendix A) to create a comma delimited text file. The first line of the file is defined as follows "TREN4" is the station id, "20020102" is the forecast date identified as January 2, 2002, "120000" is the forecast time identified as 12:00:00 GMT, "20020103" is the delivery date identified as January 3, 2002, "160000" is the delivery time identified as 16:00:00 GMT, "QR" is the physical element code referring to the value of river discharge, "R" and the following "X" refer to the type and source values of visual, observed reading, "Z" is extremum code valued as null character (no value assigned), "3.000" is the adjusted observed data value in thousands of cfs, "Z" is the data qualifier, "-1.00" is the probability code defined as a real number, "0" is the encoded duration code, "0" is the revision code, " " is the source code (left blank for possible future inclusion), and "1" is the time series indicator. These adjusted observation values are collected at 6 hour intervals out to 24 hours.

Within the same file depicted in Figure 5.2 are the river forecast guidance values. The second highlighted line is defined as follows: "TREN4" is the station id, "20020102" is the forecast date identified as January 2, 2002, "180000" is the forecast time identified as 18:00:00 GMT, "20020103" is the delivery date identified as January 3, 2002, "160000" is the delivery time identified as 16:00:00 GMT, "QR" is the physical element code referring to the value of river discharge, "F" and the following "A" refer to the type and source values of adjusted model, forecast values, "Z" is extremum code valued as null character (no value assigned), "2.500" is the adjusted observed data value in thousands of cfs, "Z" is the data qualifier, "-1.00" is the probability code defined as a real number, "0" is the encoded duration code, "0" is the revision code, " " is the source code (left blank for possible future inclusion), and "1" is the time series indicator. These forecast values are collected at 6 hour intervals out to 48 hours.

```
TREN4  ,20020102,120000,20020103,160000,QR,R,X,Z,  3.000,Z, -1.00,  0, 0,   ,1
TREN4  ,20020102,180000,20020103,160000,QR,R,X,Z,  3.000,Z, -1.00,  0, 0,   ,2
TREN4  ,20020103,000000,20020103,160000,QR,R,X,Z,  3.000,Z, -1.00,  0, 0,   ,2
TREN4  ,20020103,060000,20020103,160000,QR,R,X,Z,  3.100,Z, -1.00,  0, 0,   ,2
TREN4  ,20020103,120000,20020103,160000,QR,R,X,Z,  2.700,Z, -1.00,  0, 0,   ,2
PADP1  ,20020102,120000,20020103,160000,QR,R,X,Z,  0.700,Z, -1.00,  0, 0,   ,1
PADP1  ,20020102,180000,20020103,160000,QR,R,X,Z,  0.800,Z, -1.00,  0, 0,   ,2
PADP1  ,20020103,000000,20020103,160000,QR,R,X,Z,  0.700,Z, -1.00,  0, 0,   ,2
:
:
TREN4  ,20020103,180000,20020103,160000,QR,F,A,Z,  2.500,Z, -1.00,  0, 0,   ,1
TREN4  ,20020104,000000,20020103,160000,QR,F,A,Z,  2.400,Z, -1.00,  0, 0,   ,2
TREN4  ,20020104,060000,20020103,160000,QR,F,A,Z,  2.500,Z, -1.00,  0, 0,   ,2
TREN4  ,20020104,120000,20020103,160000,QR,F,A,Z,  2.700,Z, -1.00,  0, 0,   ,2
TREN4  ,20020104,180000,20020103,160000,QR,F,A,Z,  2.600,Z, -1.00,  0, 0,   ,2
TREN4  ,20020105,000000,20020103,160000,QR,F,A,Z,  2.500,Z, -1.00,  0, 0,   ,2
TREN4  ,20020105,060000,20020103,160000,QR,F,A,Z,  2.600,Z, -1.00,  0, 0,   ,2
TREN4  ,20020105,120000,20020103,160000,QR,F,A,Z,  2.700,Z, -1.00,  0, 0,   ,2
PADP1  ,20020103,180000,20020103,160000,QR,F,A,Z,  0.600,Z, -1.00,  0, 0,   ,1
PADP1  ,20020104,000000,20020103,160000,QR,F,A,Z,  0.600,Z, -1.00,  0, 0,   ,2
PADP1  ,20020104,060000,20020103,160000,QR,F,A,Z,  0.700,Z, -1.00,  0, 0,   ,2
```

**Figure 5.2.** Portion of the SHEF decoded output of file `2002010316dischargefcsts.ma.shef`

## 6. OVERVIEW OF ODAAS

The following sections contain detailed descriptions of the directories, scripts, scheduling, and log files associated with the river information stored in NOS' archive system ODAAS.

### 6.1. ODAAS Directories

The river observations and forecasts are stored in ODAAS directories on a SGI workstation. The river observations are located under the ODAAS directory `/rivers/obs`, and the river forecasts are located under the ODAAS directory `/rivers/fcsts`. Within each directory are subdirectories for archives (`archives/YYYYMM/ma`), log files (`execlog`), dated copies of decoded SHEF files (`recent`), and scripts (`scripts`). There is also a `modlog` file in each river data directory. This file is updated when modifications are made to the script or log files. Tables 6.1 and 6.2 depict the files located under each subdirectory and their purpose.

### 6.2. Scripts

There are two main scripts used to process river observations and forecast data. These scripts are `river_obs.sh` and `ObsToMysql.sh` for observation data or `river_fcsts.sh` and `FcstsToMysql.sh` for forecast data. The first scripts, `river_obs.sh` and `river_fcsts.sh`, rename and move the MARFC's SHEF encoded files from the MARFC FTP directory to the respective ODAAS archives directory. The second set of scripts, `ObsToMysql.sh` and `FcstsToMysql.sh`, are used to populate the rivers MySQL database.

`ObsToMysql.sh` performs the following tasks

1) copies the river observations SHEF file from the ODAAS archives directory to the local directory,
2) decodes the SHEF encoded file using NWS' SHEF parse routine (`run_sheftest`),
3) compares the comma delimited text output with existing records in the rivers database and writes out a new, comma delimited text file containing flags for duplicate values (`compare.pl`), and
4) imports the new, comma delimited text file to the rivers database using MySQL's mysqlimport tool.

This script calls two other scripts, `run_sheftest` and `compare.pl`. `Compare.pl` is a quality checking tool created using the Perl scripting language. It is designed to take the current, SHEF decoded, comma delimited text output and compare it against already existing records in the database. By appending a "v" to the end of the station ID, the script will flag records using the following criteria: the current record is already represented in the database table and the record's observation data value differs from the current record's value. The script will continue to flag with a "v" for a maximum of two v's. For more details about the function of `compare.pl` and its outputs, see Appendix B.

`FcstsToMysql.sh` performs the same tasks as `ObsToMysql.sh`; however, since all of the forecast records are necessary, the quality check provided in `compare.pl` is not used.

**Table 6. 1.** Name and Location of ODAAS' River Observations Files.

| ODAAS DIRECTORY | FILENAMES | DESCRIPTION |
|---|---|---|
| .../archives | /YYYYMM/ma/ YYYYMMDDHHdischargeobs.ma.shef | Archive of river observations |
| .../execlog | get_HHz.log | Log file of screen output during river_obs.sh run |
| | obsmysql_HHhr.log | Log file of screen output during run of ObsToMysql.sh |
| | mysqlinputerror.out | Error log file from the MySQL import tool |
| | obs_purge.log | Log file of purged river observations files from MARFC FTP directory |
| .../recent | observations.sql.YYYYMMDDHH | Dated copy of decoded SHEF file that was used as input to the MySQL database table |
| .../scripts | river_obs.sh | Renames river observations file and moves to ODAAS archives directory |
| | iget_riverobs.sh | Interactive script to rename and move river observations from FTP directory |
| | ObsToMysql.sh | Decodes the river observations file and adds it to the Mysql database observations table |
| | run_sheftest | SHEF parsing routine, utilizes parameter file, SHEFPARM |
| | compare.pl | Perl script which compares new obs records with records in the MySQL database table |

**Table 6. 2.** Name and Location of ODAAS' River Forecasts and 6-hourly Adjusted Observations Files.

| ODAAS DIRECTORY | FILENAMES | DESCRIPTION |
|---|---|---|
| .../archives | /YYYYMM/ma/ YYYYMMDDHHdischargefcsts.ma.shef | Archive of river forecast and 6-hourly adjusted obs |
| .../execlog | get_HHz.log | Log file of screen output during run of river_fcsts.sh |
| | fcstsmysql_HHhr.log | Log file of screen output during run of FcstsToMysql.sh |
| | mysqlinputerror.out | Error log file from the MySQL import tool |
| | fcsts_purge.log | Log file of purged river forecast files from MARFC FTP directory |
| .../recent | forecasts.sql.YYYYMMDDHH | Dated copy of decoded SHEF file that was used as input to the MySQL database table |
| .../scripts | river_fcsts.sh | Renames river forecasts file and moves to ODAAS archives directory |
| | iget_riverfcsts.sh | Interactive script to rename and move river forecasts from FTP directory |
| | FcstsToMysql.sh | Decodes the river forecast file and adds it to the Mysql database forecasts table |
| | run_sheftest | SHEF parsing routine, utilizes parameter file, SHEFPARM |

## 6.3. Scheduling

The scripts responsible for retrieving, parsing, and archiving river information from the MARFC are scheduled using the UNIX command '`crontab`'. The scripts, `river_obs.sh` and `river_fcsts.sh`, are designed to sleep for 15 minutes if the data is not available at the MARFC FTP directory. All scripts are identified by the UNIX '`crontab`' command file called `cronodaas` to be executed automatically. This file is located at the ODAAS home directory. In order for the ODAAS user to view the contents of `cronodaas` from any ODAAS directory, simply type the UNIX command '`crontab -l`'. The following tables depict the script launch times for observation and forecast information.

**Table 6. 3.** Location, names, launch times, and sleeping characteristics of the river observations scripts.

| ODAAS DIRECTORY *ALIAS: riverobs* | AUTO SCRIPT | TIMING CHARACTERISTICS | | | MANUAL SCRIPT |
|---|---|---|---|---|---|
| | | *Launch Time (GMT)* | *Sleep Period (sec)* | *Sleep Cutoff (GMT)* | |
| /rivers/obs/scripts | river_obs.sh | 0045, 0445, 0845, 1245, 1645, 2045 | 900 | 0200, 0600, 1000, 1400, 1800, 2002 | iget_riverobs.sh |
| /rivers/obs/scripts | ObsToMysql.sh | 0245, 0645, 1045, 1445, 1845, 2245 | n/a | n/a | n/a |

**Table 6. 4.** Location, names, launch times, and sleeping characteristics of the river forecasts scripts.

| ODAAS DIRECTORY *ALIAS: riverfcsts* | AUTO SCRIPT | TIMING CHARACTERISTICS | | | MANUAL SCRIPT |
|---|---|---|---|---|---|
| | | *Launch Time (GMT)* | *Sleep Period (sec)* | *Sleep Cutoff (GMT)* | |
| /rivers/fcsts/scripts | river_fcsts.sh | 1615 | 900 | 1800 | iget_riverfcsts.sh |
| /rivers/fcsts/scripts | FcstsToMysql.sh | 1815 | n/a | n/a | n/a |

In addition to the 'cronned' scripts listed above, a purge script was created to maintain diskspace in the MARFC's FTP directory. This purge script removes observations and forecasts files older than 7 days. The script, `purger.sh`, and log file, `purger.log`, are located under ODAAS' river scripts directory. `Purger.sh` is launched daily at 2300 GMT via the UNIX '`crontab`' command.

## 6.4. Log Files

There are several log files generated throughout the retrieving, parsing, and archiving process. The observations and forecasts' log files are located under ODAAS' river observations and forecasts `execlog` directories. There are four log files available under each `execlog` directory—

1) `get_HHz.log` ,where HH is the observation or forecast hour—screen output of movement from the MARFC FTP directory to the ODAAS archives directory,
2) `obs_purge.log` or `fcsts_purge.log`—screen output of the purge script to remove forecast files from the MARFC FTP directory,
3) `obstomysql_HHhr.log` or `fcstsmysql_HHhr.log`,where HH is the observation or forecast hour—screen output of the movement of observation or forecast files from the ODAAS archives directory to the appropriate MySQL database table, and
4) `mysqlinputerror.out`—MySQL error log file

# 7. OVERVIEW OF MYSQL

## 7.1. Introduction

MySQL is a high-performance, client/server relational database management system (RDBMS). The software was chosen by NOS because of its flexibility, speed, and low-to-no cost; for our purposes, MySQL is available for free. The system includes a SQL (Structured Query Language) server, tools and programs for client and administrative access, and customizing capabilities. The data collected in the database is organized into tables which contain rows (records) and columns (fields). Extensive online documentation is available at http://www.mysql.com/.

To use MySQL for the purposes of storing river information, the Perl DBI (Database Interface) was added to the standard system configuration. Perl is Open Source software, and it was chosen as a scripting language for interfacing with MySQL because of its ease of use, no cost, and good performance with this type of application. Details about what components you need and how to configure the DBI is available at http://www.mysql.com/downloads/api-dbi.html. For NOS' operations, four modules were installed with MySQL:

1) DBI-1.18 -- Perl DBI module,
2) Msql-Mysql-modules-1.2216 -- Perl DBD (Database Driver) module to access MySQL databases,
3) Data-Dumper-2.101 -- Perl Data-Dumper module, and
4) Data-ShowTable-3.3 -- Perl Data-ShowTable module.

Appendix C provides details about the installation and configuration process of setting up the MySQL `rivers` database with Perl DBI. The following sections describe the rivers database structure, how the river information is ingested into the database, and some general information for navigating through the database tables.

## 7.2. Database Structure

MySQL is installed on an SGI workstation (alias *mysql*). A single database was created for the rivers information called `rivers`. The data for this database is stored under the MySQL installation directory, (alias *mysql* and type "`cd /data/rivers`"). Within the `rivers` database, three tables were created: stations, observations, and forecasts.

All of the fields included in the SHEF decoded files provided by the MARFC were used in the `rivers` database tables. Different primary keys (non-null fields with unique data values) were selected for each table in order to control write access to the database. For the stations table, the station_id field was chosen as the primary key to ensure that unique stations are represented in the table. For the observations table, multiple column, primary keys were chosen – station_id, obs_date, obs_time, and source_code, so that records with the same station_id and time stamps are overwritten with each successive addition to the database table. For the forecasts table, more primary keys were chosen to ensure that all data ingested is written to the database with no successive records overwriting existing records in the database. These fields are station_id, fcsts_date, fcsts_time, create_date, create_time, and source_code. The field data types were based on the SHEF decoded

format and selected to control the size of the database tables. Depending upon the field values, the data types used were char, date, double, float, smallint, time, tinyint, or varchar. Null values were accepted as the default for all other fields that do not impact the final data quality. Table 7.1 details the fields used in each of the three tables of the rivers database.

**Table 7 1**. Descriptions of the fields used for the stations, observations, and forecasts tables.

| Fields | Descriptions |
|---|---|
| station_id | Station ID used by NOS/CSDL |
| USGS_id | USGS river gage station ID |
| NWS_id | NWS river gage station ID |
| location | Descriptive name/location of the river station |
| state | State location of the river station (two letter, postal abbreviation) |
| RFC | NWS' River Forecast Center responsible for the river station |
| WFO | NWS' Weather Forecast Office responsible for the river station |
| Fcst_horizon | Forecast horizon of the river station |
| obs_date | Date of the observed value—YYYY-MM-DD |
| obs_time | Time of the observed value—HH:MN:SS in GMT |
| fcst_date | Date of the forecast and 6-hrly adjusted observation value—YYYY-MM-DD |
| fcst_time | Time of the forecast and 6-hrly adjusted observation value— HH:MN:SS in GMT |
| create_date | Creation date of the SHEF file—YYYY-MM-DD |
| create_time | Creation time of the SHEF file— HH:MN:SS in GMT |
| P_element | SHEF Parameter Code: Physical Element—two characters; QR= discharge |
| type_code | SHEF Parameter Code: Type— one character; R= observed reading, F= forecast model |
| source_code | SHEF Parameter Code: Source— one character; X= best quality, A= adjusted, Z= nonspecific (default) |
| extremum_code | SHEF Parameter Code: Extremum—one character; Z= null character/filler |
| obs_value | Observed data value in thousands of cfs— a -9999 value denotes a bad/missing value |
| fcsts_value | Forecast and/or 6-hrly adjusted observation data value in thousands of cfs — a -9999 value denotes a bad/missing value |
| prob_code | SHEF Parameter Code: Probability—decimal value; -1.00= unspecified |
| duration_code | SHEF Parameter Code: Encoded Duration—one integer; 0= instantaneous |
| revision_code | SHEF Parameter Code: Revision—one integer; 0= not a revision, 1= revision |
| data_source | SHEF Parameter Code: Data Source—not supplied by the MARFC |
| timeseries_ind | SHEF Parameter Code: Time Series Indicator—one integer; 1= first element in a time series, 2= subsequent element |

The stations table contains station IDs, station names (locations), states, names of who monitors the stations, and forecast horizons for the nine river gauges NOS receives from the NWS' MARFC (See Table 7.2). One non-null field, station_id, was chosen as the primary key, where no duplicates can exist in the table. This field is identified by a "PRI" value in the Key column. The other fields are set-up to allow null values by default. These fields are identified with a "YES" value in the Null column. By allowing null values, we are allowing these fields to be blank in the database table. The default values for each field can be found in the Default column.

**Table 7 2.** Stations table format of MySQL `rivers` database. Primary keys are labeled PRI.

| Field | Type | Null | Key | Default |
|---|---|---|---|---|
| station_id | varchar(8) | NO | PRI | |
| USGS_id | varchar(8) | YES | | NULL |
| NWS_id | varchar(8) | YES | | NULL |
| location | varchar(50) | YES | | NULL |
| state | char(2) | YES | | NULL |
| RFC | varchar(6) | YES | | NULL |
| WFO | varchar(6) | YES | | NULL |
| Fcst_horizon | tinyint(3) | YES | | NULL |

The observations table records station ID, observation dates and times, file creation dates and times, data descriptor codes, and observation data values from the SHEF decoded, river observations file (See Table 7.3). Each field has a defined data type to minimize the number of bytes used for storage, and the multiple column, primary key is composed of four non-null, fields. The primary key fields are identified by a "PRI" in the Key column. The default values for each field are identified in the Default column.

**Table 7 3.** Observations table format of MySQL `rivers` database. Primary keys are labeled PRI.

| Field | Type | Null | Key | Default |
|---|---|---|---|---|
| station_id | char(8) | NO | PRI | |
| obs_date | date | NO | PRI | 0000-00-00 |
| obs_time | time | NO | PRI | 00:00:00 |
| create_date | date | NO | | 0000-00-00 |
| create_time | time | YES | | 00:00:00 |
| P_element | char(2) | YES | | NULL |
| type_code | char(1) | YES | | NULL |
| source_code | char(1) | NO | PRI | |
| extremum_code | char(1) | YES | | NULL |
| obs_value | float(10,3) | NO | | 0.000 |
| data_qual | char(1) | YES | | NULL |
| prob_code | double(6,2) | YES | | NULL |
| duration_code | smallint(5) | YES | | NULL |
| revision_code | tinyint(1) | YES | | NULL |
| data_source | char(8) | YES | | NULL |
| timeseries_ind | tinyint(1) | YES | | NULL |

The forecasts table records station IDs, forecast dates and times, file creation dates and times, data descriptor codes, and forecast or 6-hourly adjusted observation values from the SHEF decoded, river forecasts and adjusted observations file (See Table 7.4). Each field has a specified data type to minimize the size of the database, and the multiple column, non-null, primary key is composed of six fields. The primary key fields are identified by a "PRI" in the Key column. The default values for each field are identified in the Default column.

**Table 7 4.** Forecasts table format of MySQL `rivers` database. Primary keys are labeled PRI.

| Field | Type | Null | Key | Default |
|---|---|---|---|---|
| station_id | char(8) | NO | PRI | |
| fcsts_date | date | NO | PRI | 0000-00-00 |
| fcsts_time | time | NO | PRI | 00:00:00 |
| create_date | date | NO | PRI | 0000-00-00 |
| create_time | time | NO | PRI | 00:00:00 |
| P_element | char(2) | YES | | NULL |
| type_code | char(1) | YES | | NULL |
| source_code | char(1) | NO | PRI | |
| extremum_code | char(1) | YES | | NULL |
| fcsts_value | float(10,3) | NO | | 0.000 |
| data_qual | char(1) | YES | | NULL |
| prob_code | double(6,2) | YES | | NULL |
| duration_code | smallint(5) | YES | | NULL |
| revision_code | tinyint(1) | YES | | NULL |
| data_source | char(8) | YES | | NULL |
| timeseries_ind | tinyint(1) | YES | | NULL |

## 7.3. Database Entry

Automated database entry is achieved through one of two scripts, `ObsToMysql.sh` or `FcstsToMysql.sh`. These scripts add the SHEF decoded records to the observations or forecasts tables of the MySQL rivers database, respectively.

The `ObsToMysql.sh` script is designed to temporarily copy a SHEF encoded, river observations file from ODAAS' `archives` directory to the `scripts` working directory, decode the file using NWS' SHEF parsing routine by launching `run_sheftest`, compare the text output with existing records in observations table of the `rivers` database by launching the quality control, Perl script `compare.pl` (Appendix B), import the Perl script's output text file, `observations.sql`, to the observations table of the rivers database using MySQL's `mysqlimport` utility, and save a copy of `observations.sql` with a date/time stamp to the `recent` directory. As a safeguard, the script also checks to see that the MySQL database process is still running; otherwise, the script will re-launch the daemon. More details follow regarding the importance of the MySQL daemon.

The `FcstsToMysql.sh` script has the same functionality as `ObsToMysql.sh` with the exception of the Perl script `compare.pl`. Because the forecasts and 6-hourly adjusted observations are received once per day, all of the records are stored in the database, regardless of redundancy and `compare.pl` does not have to be run. The `FcstsToMysql.sh` script is designed to temporarily copy a SHEF encoded, river forecasts file from ODAAS' `archives` directory to the `scripts` working directory, decode the file using the SHEF parsing routine by launching `run_sheftest`, copy the `run_sheftest` output to a file called `forecasts.sql`, import `forecasts.sql` to the forecasts table of the rivers database using MySQL's `mysqlimport` utility, and save a copy of `forecasts.sql` with a date/time stamp to the `recent` directory. Similar to `ObsToMysql.sh`, the script also checks to see that the MySQL daemon is running.

MySQL has an importer utility, `mysqlimport`, which acts as a command line interface to the `LOAD DATA` statement. The `LOAD DATA` statement reads data from a file and bulk loads it to a specified database table. From a shell interface, the `mysqlimport` utility generates a `LOAD DATA` statement and functions similarly. For the comma-delimited input text file, the syntax for mysqlimport is as follows,

```
bin/mysqlimport -u username --local -r --fields-terminated-by=, databasename inputfile.sql
```

The MySQL daemon is critical to utilizing MySQL. To start the process in the background, one should go to the MySQL home directory (alias *mysql*) and enter "`bin/safe_mysqld &`." To stop the daemon, one will need to use "`ps -u ODAAS_username`" to find the process and then use "`kill PID`," where PID is the process ID, to kill the process. There are two processes associated with MySQL: `safe_mysql` and `mysqld`.

## 7.4. Navigating Through MySQL

Currently, only the ODAAS user account has access to MySQL and its databases. To log on to MySQL, one needs to navigate to the MySQL home directory (alias *mysql*) and type "`bin/mysql`." Once logged in, one can list all databases by typing "`SHOW DATABASES;`" and one can get to the `rivers` database by typing "`USE rivers;`." Notice that nearly all commands in MySQL are followed by a semi-colon (";").

There are multiple methods one can use to navigate around a MySQL database; however, the following are some useful commands to get started.

1) To view the tables in the `rivers` database, type "`SHOW TABLES;`." The tables listed under the rivers database are observations, forecasts, and stations.
2) Descriptions of the tables can be viewed by entering "`DESCRIBE tablename;`."
3) The stations table records can be viewed by typing "`SELECT * FROM stations`."
4) Alternatively, if one would like to find out how many records are returned by a query, one should type "`EXPLAIN SELECT * FROM stations;`." Due to a very large recordset, this method is recommended to predefine your query before running a "`SELECT * FROM`" statement against the observations and forecasts tables.
5) To select a subset of the observations table, one might use the following syntax
"`SELECT * FROM observations WHERE obs_date="2002-12-22" AND station_id LIKE "TR%;`."
This statement will return all records from the observations table that have an observation date of December 22, 2002 and a station_id that begins with TR, which in this case will be TREN4 plus any additional TREN4 IDs with an appended "v" (flag for multiple observation data records).
6) One can end a MySQL session by entering "`QUIT`" at the command prompt.
7) To do a text dump of a table, one can enter
"`mysqldump db_name table_name > outputfile`" from the MySQL home directory and at the UNIX prompt.

Additional information about MySQL commands is available in numerous textbooks, such as MySQL by DuBois (2000), MySQL and mSQL by Yarger et al. (1999), and on the Internet at http://www.mysql.com.

## 8. MONITORING PROCEDURES

The FTP arrival status of the river observations and forecasts from the MARFC is monitored using ODAAS' performance reports, the "Automated 00Z ODAAS Report," "Daily Status Report," and "Weekend Status Report." The scripts that create these reports check for the existence of the observations and forecasts files in the ODAAS archives directory.

In addition to the status reports, a quality check is performed within the `river_fcsts.sh` script to check for discrepancies between the date/time stamp in the filename and the date/time stamp within the file itself. If there is a discrepancy, an alert e-mail is sent to the ODAAS operator. This addition to the script was prompted by a file received from the MARFC, which when discovered, could no longer be investigated for the cause of discrepancy.

## 9. SUMMARY AND THE FUTURE

In summary, the following flowchart depicts the process of receiving and archiving river information received from the MARFC.

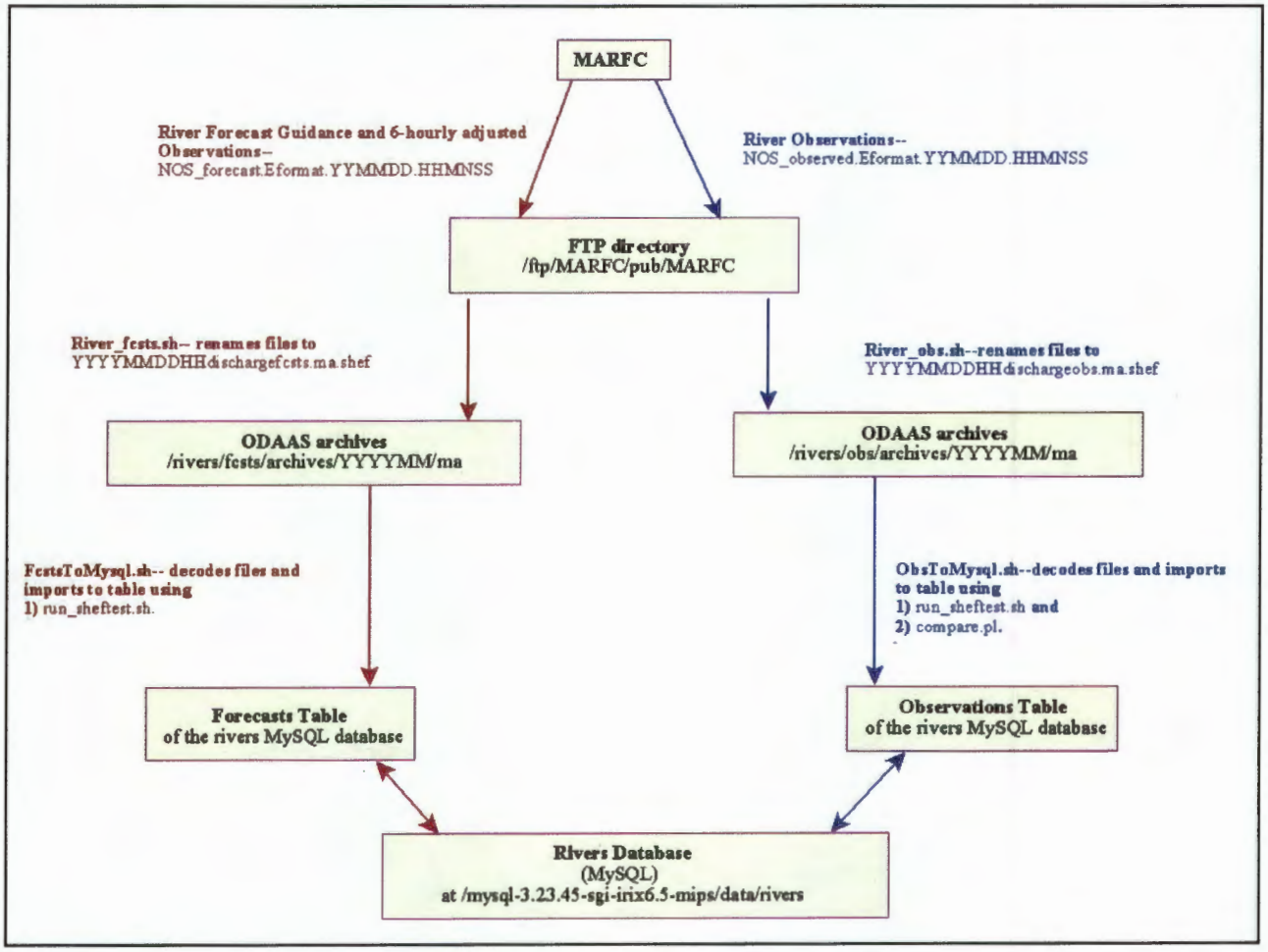


**Figure 9.1.** Flowchart of river information from the NWS' MARFC, to the ODAAS archives, and to the MySQL rivers database tables, forecasts and observations. YYYY is the year, MM is the month, DD is the day, HH is the hour, MN is the minute, and SS is the second.

The improvements, changes, or additions that will be applied to ODAAS' system for ingesting river information are as follows:

a) Improve the process flow in `compare.pl` by fine-tuning the nested "if" statements within the query that checks for record matches in the database,
b) Obtain river observations and NWS' river forecast guidance for the Patapsco, Mattaponi, and Pamunkey Rivers from the MARFC to support CBOFS and for the Hudson and Hackensack Rivers from the Northeastern River Forecast Center (NERFC) located at Taunton, MA to support NYOFS (See Figure 9.2), and

## 9. SUMMARY AND THE FUTURE

In summary, the following flowchart depicts the process of receiving and archiving river information received from the MARFC.

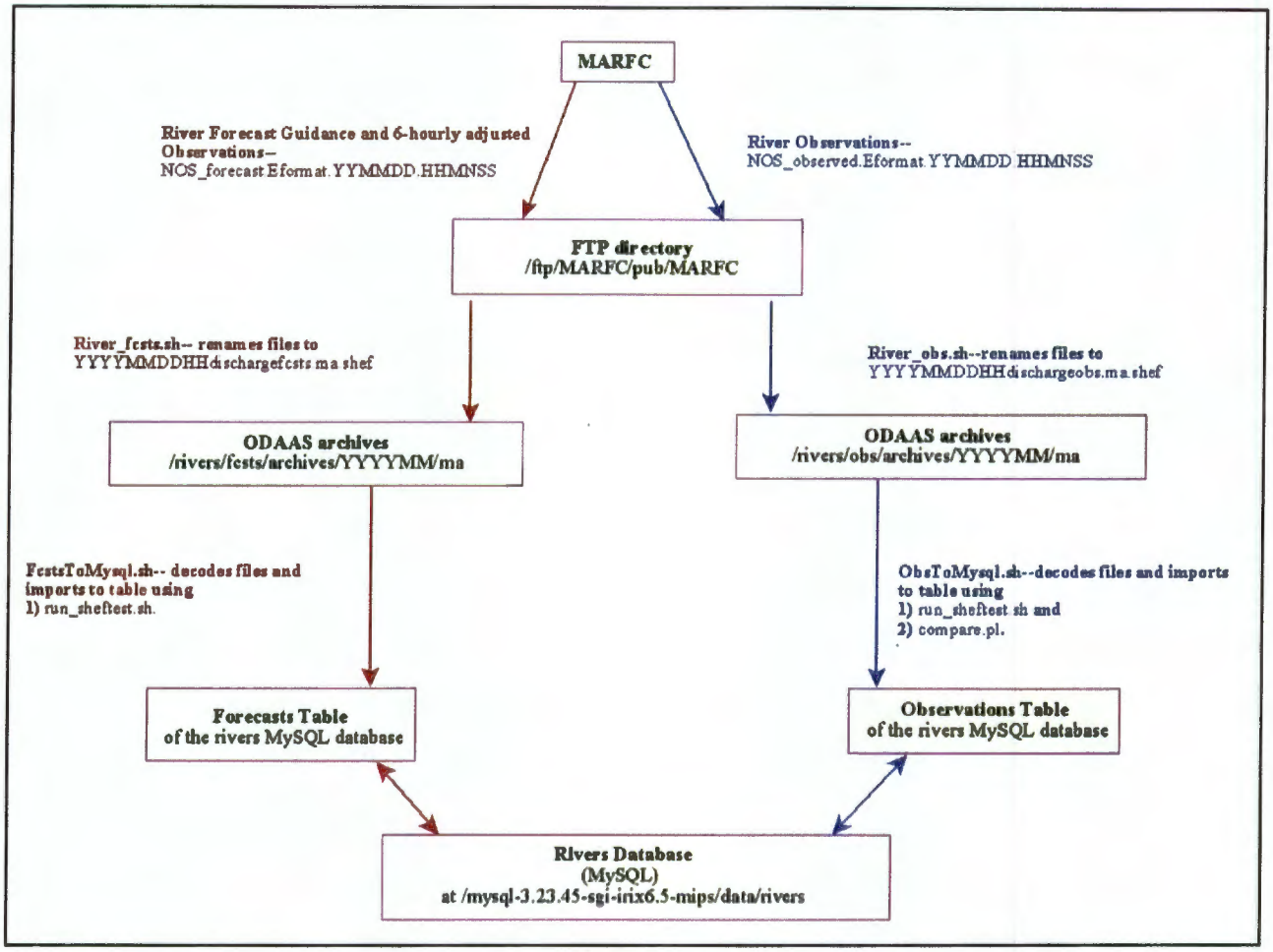


**Figure 9. 1.** Flowchart of river information from the NWS' MARFC, to the ODAAS archives, and to the MySQL rivers database tables, forecasts and observations. YYYY is the year, MM is the month, DD is the day, HH is the hour, MN is the minute, and SS is the second.

The improvements, changes, or additions that will be applied to ODAAS' system for ingesting river information are as follows:

a) Improve the process flow in `compare.pl` by fine-tuning the nested "if" statements within the query that checks for record matches in the database,
b) Obtain river observations and NWS' river forecast guidance for the Patapsco, Mattaponi, and Pamunkey Rivers from the MARFC to support CBOFS and for the Hudson and Hackensack Rivers from the Northeastern River Forecast Center (NERFC) located at Taunton, MA to support NYOFS (See Figure 9.2), and

c) In support of NOS' Galveston Bay Operational Forecast System, obtain river observations and NWS' river forecast guidance for the Trinity and San Jacinto Rivers from the Western Gulf River Forecast Center (WGRFC) located at Fort Wayne, TX.

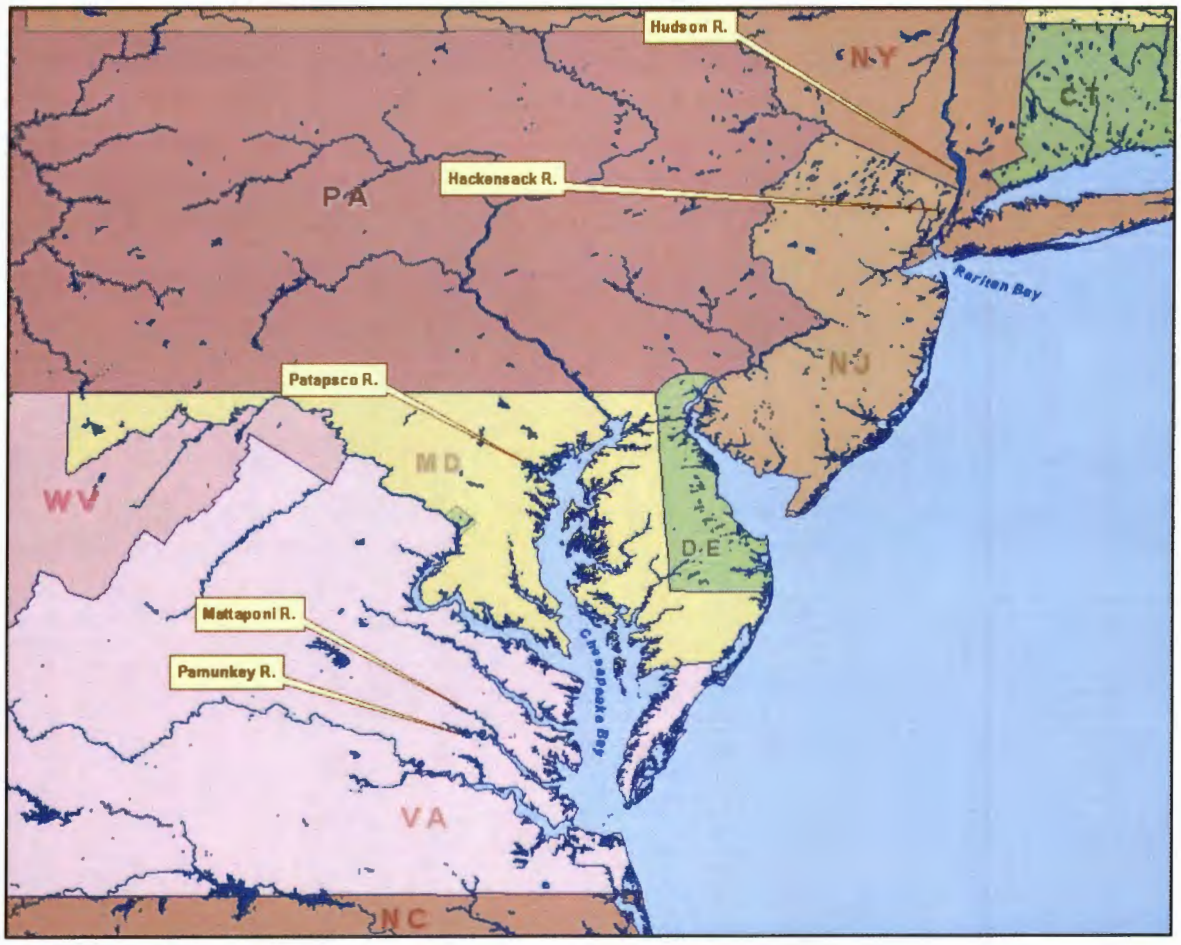


**Figure 9. 2.** The names and locations of outstanding rivers from which streamflow observations and forecast guidance are still needed by NOS' CBOFS and NYOFS.

The procedure for including additional river observations and forecast guidance from future RFCs is detailed in Appendix D.

## 10. ACKNOWLEDGMENTS

Joe Ostrowski, the Development and Operations Hydrologist at the NWS' MARFC, worked with CSDL in compiling the stations, sending them to CSDL in a timely manner, and generally helping out whenever questions arose or modifications to the data were needed. He was invaluable to the process of collecting river information for ODAAS. David Street and Virginia Galvin of NWS answered questions regarding the installation of the SHEF decoding software.

CSDL's Meredith Westington, John Kelley and Eugene Wei were responsible for ODAAS at CSDL and CO-OPS. The initial work of obtaining river observations and forecast guidance was done by Allison Thomson, a former Environmental Careers Organization (ECO) intern with CSDL. Tom Gross and Zhen Li introduced MySQL to MMAP for storing river information.

## REFERENCES

DuBois, P. 2000. **MySQL**. Indianapolis: New Riders Publishing Company. 756 pp.

Kelley, J., M. Westington, E. Wei, S. Maxwell, A. Thomson. 2001. "Description of the Operational Data Acquisition and Archive System (ODAAS) to Support the NOS Chesapeake Bay Operational Forecast System (CBOFS)." **NOAA Technical Report** NOS CS 10. Coast Survey Development Laboratory, Silver Spring, MD. 45 pp.

Office of Hydrology 1998. "Standard Hydrometeorological Exchange Format, Version 1.3.1." **Weather Service Hydrology Handbook No. 1**. National Weather Service, Silver Spring, MD. 63 pp.

Thomson, A. M. 2000. "Acquisition Status of Real-Time River Observations and River Forecast Guidance for Use in NOS Estuarine Nowcast/Forecast System Development." **NOAA Technical Report** NOS CS 8. Coast Survey Development Laboratory, Silver Spring, MD. 27 pp.

Yarger, J., G. Reese, and T. King 1999. **MySQL and mSQL**. Sebastopol, CA: O'Reilly and Associates, Inc. 487 pp.

## APPENDIX A. CUSTOMIZING THE SHEF DECODER

The SHEF decoding software was created in Fortran. The software package when unzipped contains a series of directories and program files. In order to better format the data for easier ingest to a MySQL database, the SHEF decoder software was modified. The program `shpout.f` was adjusted for this purpose. The program can be found under ODAAS' SHEF parser installation directory, `/tools/decoders/shefpars/shef_sheftest` directory (alias *`shefpars`* and type "`cd shef_sheftest`"). Modifications include the rearrangement of output parameters and their significant digits and the addition of comma delimitation. Figures A.1 and A.2 contain highlighted text of the changes made to `shpout.f`.

```
C ==================================================================
C pgm: SHPOUT .. Display "shefout" file
C
C use:    CALL SHPOUT(LUOUT,LUDIS)
C
C  in: LUOUT ........ logical unit number of "shefout" file -INT
C  in: LUDIS ........ logical unit number of display - INT
C ==================================================================
      SUBROUTINE SHPOUT(LUOUT,LUDIS)

      CHARACTER*8      ID,JID,PC
      CHARACTER*1      IQUAL
      CHARACTER*3      D1,D2,D3,D4,D5,D6
      INTEGER          LUOUT,LUDIS,IEND
      INTEGER          LDT(6),JDT(6),IDUR,IREV,ITIME
      DOUBLE PRECISION  VALU
      REAL             CODP

  200 FORMAT(A8,I4,5I2,I5,5I2,1X,A2,1X,A3,
     $    F10.3,1X,A1,F6.2,I5,I2,1X,A8,I1)

       IF (LUOUT.GE.0 .AND. LUDIS.GE.0) THEN

        REWIND (LUOUT,IOSTAT=IEND)

  100    IF (IEND .NE. 0) GOTO 110
         READ(LUOUT,IOSTAT=IEND) ID,LDT,JDT,PC(1:1),D1,PC(2:2),D2,
     $                IDUR,PC(4:4),D3,PC(5:5),D4,PC(6:6),D5,
     $                CODP,VALU,IQUAL,D6,IREV,JID,ITIME
         IF (IEND .EQ. 0) THEN
          WRITE(LUDIS,200,IOSTAT=IEND) ID,LDT,JDT,PC(1:2),
     $          PC(4:6),VALU,IQUAL,CODP,IDUR,IREV,JID,ITIME
         ENDIF
         GOTO 100
  110    CONTINUE

       ENDIF

      RETURN
      END
```

**Figure A.1**. The original `shpout.f` program packaged with the default SHEF software.

```
200 FORMAT(A8,',',I4,2I2.2,',',3I2.2,',',I4,2I2.2,',',3I2.2,',',A2
  $     ',',A1,',',A1,',',A1,',',F10.3,',',A1,',',F6.2,',',I5
  $     ',',I2,',',A8,',',I1)

    IF (LUOUT.GE.0 .AND. LUDIS.GE.0) THEN

     REWIND (LUOUT,IOSTAT=IEND)

100   IF (IEND .NE. 0) GOTO 110
      READ(LUOUT,IOSTAT=IEND) ID,LDT,JDT,PC(1:1),D1,PC(2:2),D2,
  $                IDUR,PC(4:4),D3,PC(5:5),D4,PC(6:6),D5,
  $                CODP,VALU,IQUAL,D6,IREV,JID,ITIME
      IF (IEND .EQ. 0) THEN
       WRITE(LUDIS,200,IOSTAT=IEND) ID,LDT,JDT,PC(1:2),
  $    PC(4:4),PC(5:5),PC(6:6),VALU,IQUAL,CODP,IDUR,IREV,JID,ITIME
```

**Figure A.2**. Changed portion of the `shpout.f` program.

In addition to modifying the Fortran program, `shpout.f`, changes were also made to the UNIX Korn Shell script `run_sheftest`. `Run_sheftest` was relocated from ODAAS' SHEF parser installation directory, `/tools/decoders/shefpars/shef_scripts` (alias *shefpars* and type "`cd shef_scripts`") to two places: the observations `scripts` directory and the forecasts `scripts` directory. The highlighted text in Figure A.3 shows the types of changes made to `run_sheftest` for use with ODAAS scripts `ObsToMysql.sh` and `FcstsToMysql.sh`.

```
#!/bin/ksh
# ==========================================================================
# pgm: run_sheftest .. script to make a test run using sheftest
#
# use:    run_sheftest
# ==========================================================================

In=$1

D_cur=/dir1/adaser/tools/decoders/shefpars/shef_sheftest
D_wrk=/dir1/adaser/rivers/obs/scripts
D_tes=${D_cur%/*}/shef_sheftest
D_dat=${D_cur%/*}/shef_data

[[ -d $D_wrk ]] || mkdir $D_wrk

[[ -d $D_wrk ]] || { print "ABORT: cannot find dir \"$D_wrk\"!"; exit 1; }
[[ -d $D_tes ]] || { print "ABORT: cannot find dir \"$D_tes\"!"; exit 1; }
[[ -d $D_dat ]] || { print "ABORT: cannot find dir \"$D_dat\"!"; exit 1; }

P_shf=$D_tes/sheftest

[[ -x $P_shf ]] || { print "ABORT: no sheftest executable!"; exit 1; }

F_par=$D_dat/SHEFPARM
F_sin=$D_wrk/$In

T_par=$D_wrk/SHEFPARM
T_sin=$D_wrk/$1

[[ -f $T_par ]] || cp $F_par $T_par || exit 2
[[ -f $T_sin ]] || cp $F_sin $T_sin || exit 3

cd $D_wrk || exit 4

$P_shf << EoF
$In
EoF

print ""
print "  For input file:          $D_wrk/$In"
print ""
print "  Binary shefout file:     $D_wrk/shefout"
print "  Text display of data:    $D_wrk/display"
print "  Text err and msg listing: $D_wrk/error"
print ""
```

**Figure A.3**. The `run_sheftest` script was modified to reflect its use and directory location. Note that the changes are highlighted, and this file is the `run_sheftest` routine for the river observations.

## APPENDIX B. DETAILS OF COMPARE.PL

The creation of a quality control script was necessary to ensure that gross errors (-9999.000 values) and duplicate data records were eliminated before being imported into the MySQL rivers database. Because river observations are received over the previous 24 hours every 4 hours, there are, at most, seven possible data values for the river observations. In an ideal world, the river gage would be accurate and would report the same value at each time interval; however, there are several instances where a good observation is received at one hour and then flagged as bad 4 hours later. The observation that differs from an existing observation is flagged and entered into the database. In cases of discrepancy, the MARFC has advised CSDL users to compare the observations with the 6-hourly adjusted observations.

Perl was chosen as the scripting language for this application because of its ease of use and low cost. Perl is Open Source software and is available for download at several websites, including http://www.perl.com/pub/a/language/info/software.html and http://www.cpan.org/.

In order to automate the process of quality checking the incoming data, a script, `compare.pl`, was created using the Perl scripting language. The script is located under ODAAS' scripts directory for river observations. `Compare.pl` is designed to

1) read a SHEF decoded observations data file,
2) compare each record against existing records in the MySQL database table observations,
3) write out a new text file, `observations.sql`, which is a reprint of the original observations data file with some modifications to the station_id if there is an already existing record with a different observation value in the database table, and
4) import the observations data to the existing observations database table.

Figure B.1 shows the `compare.pl` script.

```
# connect to DBI and to rivers database
use DBI;
use strict;

my $db = DBI->connect('DBI:mysql:rivers',undef,undef);


#open output file from run_sheftest, file is called display
open (RVFILE, $ARGV[0]) or die "Couldn't open $ARGV[0]";
#open a file to use to write to database with mysqlimport tool, file is called observations.sql
open (OBSFILE,">> observations.sql") || die "Cannot open observations.sql file";

#while reading lines in the display file
while (my $line = <RVFILE>) {
    #for each line in the display file
    NEW:  foreach ($line) {
        #create an array of fields created from parsing each line by a comma
        my @fields = split(/,/,$line);
```

**Figure B.1**. The `compare.pl` script.

```
#create a small subset of the station_id in order to search for wildcard chars later
my $smfield1 = substr($fields[0],0,4); #first 4 chars of station_id

#create a query of the observations table, where this line matches the table values
my $query = "SELECT * FROM observations WHERE station_id LIKE '$smfield1%' AND
                                    obs_date='$fields[1]' AND obs_time='$fields[2]'";

#prepare the query in a variable called myobs_output
my $myobs_output = $db->prepare($query);

#execute query for each line
$myobs_output->execute;

#create an array of the matched database row
my @ary = $myobs_output->fetchrow_array;
#queryvalue gives the number of database matches
my $queryvalue = $db->do($query);


#if the array exists, meaning there is a query match in the database
if ( @ary ) {
    #then while we are getting the rows, check the queryvalue to determine how many match
    while (my @ary = $myobs_output->fetchrow_array) {
        #if more than one row matches and then check the fields
        if ( $queryvalue > 1)  {
            #if the station_id contains a v and the datafields do not match
            if (($ary[0] =~ /v/) && ($fields[9] != $ary[9])) {
                $fields[0] =~ s/\s/vv/;
                print OBSFILE join
                            (",",$fields[0],$fields[1],$fields[2],$fields[3],$fields[4],$fields[5],$fiel
                            ds[6],$fields[7],$fields[8],$fields[9],$fields[10],$fields[11],$fields[12],
                            $fields[13],$fields[14],$fields[15]), "";
            }
            #if the station_id contains a v and the fields match, overwrite v rec
            if (($ary[0] =~ /v/) && ($fields[9] == $ary[9])) {
                $fields[0] =~ s/\s/v/;
                print OBSFILE join
                            (",",$fields[0],$fields[1],$fields[2],$fields[3],$fields[4],$fields[5],$fiel
                            ds[6],$fields[7],$fields[8],$fields[9],$fields[10],$fields[11],$fields[12],
                            $fields[13],$fields[14],$fields[15]), "";
            }
        #go to next new display input file record
        next NEW;
        }
    }

    #if less than two rows match the query
    if ( $queryvalue < 2 ) {
        #if the database obs value is -9999, overwrite with new rec
        if ( $ary[9] eq "-9999.000" ) {
            print OBSFILE join (",", $line), "";
        }
        #if the obs data fields match then overwrite with latest rec
        elsif ( $ary[9] == $fields[9] ) {
```

**Figure B.1**. (continued)

```
                    print OBSFILE join (",", $line), "";
                }
                #else if there is a query match, but the obs values don't match, add a v
                else {
                    $fields[0] =~ s/\s/v/;
                    print OBSFILE join
                                (",",$fields[0],$fields[1],$fields[2],$fields[3],$fields[4],$fields[5],$fiel
                                ds[6],$fields[7],$fields[8],$fields[9],$fields[10],$fields[11],$fields[12],
                                $fields[13],$fields[14],$fields[15]), "";
                }
            }

        }
        #if there is no query match to the database, write out the new record
        if (! @ary) {
            print OBSFILE join (",", $line), "";
        }
        #reset queryvalue to zero
        $queryvalue=0;
    }
}
```

**Figure B.1.** (continued)

## APPENDIX C. INSTALLING AND CONFIGURING MYSQL WITH PERL DBI

### INSTALLATION NOTES

### 1. MYSQL:

1) Obtain the installation tar.gz file from http://www.mysql.com
2) Copy tar.gz file to space on disk.
3) Unzip the file using "`gunzip filename.tar.gz`"
4) Untar using "`tar -xvf filename.tar`"—note: this will create and place files beneath a directory called *filename*
5) Run the following command from the top level directory – "`scripts/mysql_install_db`"

   The following is output from running the command above:

   Preparing db table
   Preparing host table
   Preparing user table
   Preparing func table
   Preparing tables_priv table
   Preparing columns_priv table
   Installing all prepared tables
   011127 18:18:35 ./bin/mysqld: Shutdown Complete

   To start mysqld at boot time you have to copy support-files/mysql.server
   to the right place for your system

   PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
   This is done with:
   ./bin/mysqladmin -u root -p password 'new-password'
   ./bin/mysqladmin -u root -h *SGI_domain_name* -p password 'new-password'
   See the manual for more instructions.

   You can start the MySQL daemon with:
   cd . ; ./bin/safe_mysqld &

   You can test the MySQL daemon with the benchmarks in the 'sql-bench' directory:
   cd sql-bench ; run-all-tests

   Please report any problems with the ./bin/mysqlbug script!

   The latest information about MySQL is available on the web at
   http://www.mysql.com
   Support MySQL by buying support/licenses at https://order.mysql.com

6) Start the mysql daemon by typing "`bin/safe_mysqld &`" at the top level directory
7) Type the following command to enter mysql database "`bin/mysql -u root`"
8) To set the root password type:
   "`USE mysql;`"
   "`SHOW TABLES;`"

```
"DESCRIBE user;"
"SELECT user, password FROM user;"
"UPDATE user SET password=PASSWORD('passwd') WHERE user="root";"
"FLUSH PRIVILEGES;"
```

## 2. SETTING UP THE RIVERS DATABASE:

1) Set-up the rivers database by typing `"create database rivers;"` at the MySQL prompt
2) Grant privileges to the rivers database by typing `"GRANT ALL ON rivers.* TO ODAAS_username@localhost;"`
3) Logout as root by typing `"quit"` and re-login under the ODAAS username by typing `"bin/mysql"`
4) Create new tables or can add old database tables to new rivers database by copying the files in the "data" directory over.

## 3. DBI:

The following Perl/DBI modules are available at http://www.mysql.com/downloads/api-dbi.html.

- DBI/DBI-1.13.tar.gz
- Data/Data-ShowTable-3.3.tar.gz
- Data/Data-Dumper-2.101.tar.gz
- DBD/Msql-Mysql-modules-1.2217.tar.gz

1) Save the downloaded modules listed above to the `mysql` directory.
2) Enter the following commands:

```
gunzip DBI-1.13.tar.gz
tar xvf DBI-1.13.tar
cd DBI-1.13
perl Makefile.PL
make
make test
make install

cd ..
gunzip Data-ShowTable-3.3.tar.gz
tar xvf Data-ShowTable-3.3.tar
cd Data-ShowTable-3.3
perl Makefile.PL
make
make install    # Don't try make test, the test suite is broken

cd ..
gunzip Data-Dumper-2.101.tar.gz
tar xvf Data-Dumper-2.101.tar
cd Data-Dumper-2.101
perl Makefile.PL
make
```

```
make test
make install

cd ..
gunzip Msql-Mysql-modules-1.2217.tar.gz
tar xvf Msql-Mysql-modules-1.2217.tar
cd Msql-Mysql-modules-1.2217
perl Makefile.PL (***see below for output from this command)
make
make test
make install
```

The following is a screen capture of the makefile.pl process when installing Msql-Mysql-modules-1.2217. The user entries are denoted in bold, Courier font.

*SGI_name 17#* **perl Makefile.PL**
Which drivers do you want to install?
1) MySQL only
2) mSQL only (either of mSQL 1 or mSQL 2)
3) MySQL and mSQL (either of mSQL 1 or mSQL 2)

4) mSQL 1 and mSQL 2
5) MySQL, mSQL 1 and mSQL 2
Enter the appropriate number: [3] **1**

Do you want to install the MysqlPerl emulation? You might keep your old
Mysql module (to be distinguished from DBD::mysql!) if you are concerned
about compatibility to existing applications! [n] **n**

Where is your MySQL installed? Please tell me the directory that
contains the subdir 'include'. [/usr/local] /*SGI_directory*/mysql-3.23.45-sgi-irix6.5-mips
Which database should I use for testing the MySQL drivers? [test] **rivers** (*** **Note:** Although this text is meant to be useful, the following information may be incorrect. In hindsight, the correct entry at this stage should have been **test**. The following information is based on my entry as **rivers**; use with caution.)

On which host is database rivers running (hostname, ip address
or host:port) [localhost] ***ODAAS_username***

User name for connecting to database rivers? [undef] ***ODAAS_username***
Password for connecting to database rivers? [undef] ***password***
Creating files for MySQL...
Checking if your kit is complete...
Looks good
Note (probably harmless): No library found for -lgz
Using DBI 1.18 installed in /usr/share/lib/perl5/site_perl/irix-n32/auto/DBI
Writing Makefile for DBD::mysql
Writing Makefile for Msql-Mysql-modules

The following are additional notes taken from Internet Perl/DBI resources:

'During "perl Makefile.PL" you will be prompted some questions. In particular you have to choose the installed drivers (MySQL, mSQL2 and/or mSQL1). The MySQL driver will be called

DBD::mysql, and a single mSQL driver will be called DBD::mSQL. If you want to support both mSQL1 and mSQL2, the former will be DBD::mSQL1.

Other questions are the directories with header files and libraries. For example, if your file mysql.h is in /usr/include/mysql/mysql.h, then enter the header directory /usr, likewise for /usr/lib/mysql/libmysqlclient.a or /usr/lib/libmysqlclient.so.'

## APPENDIX D. STEPS FOR ADDING NEW STATIONS TO THE RIVERS DATABASE

In the future, CSDL intends to include river observations and NWS forecast guidance for other estuaries along the East and Gulf Coasts. The following text describes the process necessary for adding stations to the current river information ingest system.

The first task will be to set-up a new FTP account for the new RFC to deposit files containing river observations, forecast guidance, and 6-hrly adjusted observations. The home directory of the new account should be a sub-directory under `/ftp`.

The second task will be to set-up the appropriate archives directory structure under ODAAS' river observations and forecasts directories. Under the `archives` directory, there is a month directory called `YYYYMM`, where `YYYY` is the year and `MM` is the month. Currently there is a sub-directory called `ma` under this directory which contains river information from the MARFC. A similar, meaningful name should be given for new sub-directories.

The next task will be to identify the file structure of the new incoming information. When new stations arrive from other RFC's, it is important that the SHEF encoded file formats are exactly the same as those received from the MARFC. The names of the files are not important, but the way the internal information is ordered is critical. Section 4 of this technical report describes the structure of the SHEF encoded files in detail. One way to check if the file formats are the same is to decode them with the `run_sheftest` program created by NWS. If the files parse like those depicted in Section 5, then the files will correspond later to the fields defined in the MySQL database.

Assuming that the file formats are identical, the next step is to include the new file names in the ODAAS archive scripts, `river_obs.sh` and `river_fcsts.sh`. These scripts will take files from the RFC's home FTP directory, rename them, and copy them to the ODAAS archives directory specified in the second step listed above. The modifications required include the addition of extra lines of script similar to what is already included. Adjustments will need to be made to file names and paths that reflect the new river information.

In order to prepare for the transfer of information to the rivers MySQL database, one should modify the stations table. A record for each new river station should be added to the table with information such as the station ID, USGS ID, NWS ID, location/name, state, RFC, WFO, and forecast horizon. There are numerous ways to add a new record to the table; however, the following code was typed at a MySQL command prompt to add river stations, TREN4 and PADP1, to the stations table:

```
"INSERT INTO stations VALUES ('TREN4','1463500','TREN4','Delaware River at
Trenton','NJ','MARFC','KPHI','24');('PADP1','1474500','PADP1','Schuykill River at
Philadelphia','PA','MARFC','KPHI','24');"
```

Once the files are archiving successfully via the scripts `river_obs.sh` and `river_fcsts.sh`, the next step is to incorporate the archived files into the scripts `ObsToMysql.sh` and `FcstsToMysql.sh`. One way to add future river files to these scripts is to open each script in a text editor, select all of the scripting code, copy it, and append it to the original script file. The appended lines of script should be modified to reflect the file names and paths of the new river information. Assuming that the file structures are the same as those received from the MARFC, the river

information should import directly to the already existing observations and forecasts tables of the `rivers` database. The new river information will be appended to the existing tables.