

Documentation for Transfer of Verify Database Data revised October 14,2004

1.0 General Information

1.1 Application Description

The transfer of verify database data consists of 1) a procedure for the meta-data and reference tables and 2) two applications with companion scripts for SHEF encoding the observed and forecast data. These applications are:

```
vfyobs2shef (esql/C)
run_vfyobs2shef (bash script)
vfyfcst2shef (esql/C)
run_vfyfcst2shef (bash script)
```

The procedure covers the unloading and loading of data from three tables of the verify database, vlocation, vrivergaugeloc and addadjust. The SHEF encoder applications read the data from the verify database observed and forecast data monthly tables. Since the monthly tables in existence at each RFC could vary, the applications read the sysmaster database systabnames table to find out the table names at each RFC. All applications take advantage of the apps_defaults system.

1.2 Design Considerations

The vlocation and vrivergaugeloc table definitions in the RFC Archive database are identical to the existing definitions in the verify database. The verify db addadjust table has a new name and the table structure has some minor changes. See Attachment A thru C see for table definitions.

1.3 Enhancements/Bug Fixes/Changes

Build OB4

With Build OB4 tables vlocation and vrivergaugeloc no longer exist. Section 3.0 of this document has been updated accordingly.

2.0 Configuration Information

Two apps_defaults tokens are used by the various programs and scripts, these are:

vsys_vdb name of the verify database on ds1
server_name Informix server name on ds1

The tokens vsys_vdb and server_name are used by both applications, vfyobs2shef and vfyfcst2shef.

3.0 Procedure for Transferring Data From Meta-data and Reference Tables

vlocation table

This table is not used by the OB4 version of the verify application. The table is eliminated in Build OB4.

vrivergaugeloc table

This table is not used by the OB4 version of the verify application. The table is eliminated in Build OB4.

vaddadjust table

This table may or may not have any rows in it. If rows exist, generally it is just a small number of rows. Use dbaccess and unload the data from the addadjust table of the verify database (vdb1_1xxx, where xxx is the RFC 3-character identifier). Suggested command is:

```
unload to "vadd.unl"  
select * from addadjust
```

Before loading this dataset, the user must do some minor editing of the file. The data is basically in the same order but column "pe" is split into two columns, insert a column before the numeric duration for the character duration code and split the "ts" column into two columns. See Attachment C for details. In the case of the duration code, in the current verify database only the numeric value of the SHEF duration code is stored, but in the archive database both the numeric value and character code for the SHEF duration code are stored. For example, 0 ("zero") is the numeric duration code and I is the character SHEF duration code, 2001 is a D, 5004 is a P and so on.

Example *BEFORE* file:

```
AJMM7|HG|0|RG|Z|10.0|  
JFMM7|HG|0|RG|Z|20.0|
```

Example *AFTER* file:

```
AJMM7|H|G|I|O|R|G|Z|10.0|  
JFMM7|H|G|I|O|R|G|Z|20.0|
```

Transfer this file to the archive box and load the data into the RFC Archive database (adb5_22xxx, where xxx is the RFC 3-character identifier) with dbaccess. Suggested command is:

```
load from "vadd.unl"  
insert into vaddadjust
```

4.0 Running vfyobs2shef and vfyfcst2shef Programs

Creating the shef messages for the RFC Archive DB's raw shefdecoder is as easy as one, two, three, etc. The steps are as follows:

- login in as oper on the archive system
- cd to /arc/bin/vfyprogs

- execute the run_vfyobs2shef script
- optional - review the output files (obshtccyymm.out)
- copy or move the ".out" files to the shef_decoder_raw input directory (/rfc_arc_data/q/raw)
Older data may not post if the data is older than the time window that your office has defined for the raw shefdecoder ... it is suggested that the token values be reviewed and any needed changes are made before moving these files.

- execute the run_vfyfcst2shef script
- optional - review the output files (fcsthtccyymm.out)
- copy or move the ".out" files to the shef_decoder_raw input directory (/rfc_arc_data/q/raw)
Older data may not post if the data is older than the time window that your office has defined for the raw shefdecoder ... it is suggested that the token values be reviewed and any needed changes are made before moving these files.

Note: 1) These ".out" files can be quite large and depending on the number of years of data stored in the verify database there can be quite a few of them. 2) These applications can take several minutes to run.

4.1 Example vfyobs2shef output file

```
ZCZC WOHVERIFY DEF  
TTAA00 KWOH 292207
```

.A ABCL1 20020301 Z DH000000/HGIRGZ 7.84
.A ABCL1 20020301 Z DH060000/HGIRGZ 7.84
.A ABCL1 20020301 Z DH120000/HGIRGZ 7.84
.A ABCL1 20020301 Z DH180000/HGIRGZ 7.87
.A ABCL1 20020302 Z DH000000/HGIRGZ 8.09
.A ABCL1 20020302 Z DH060000/HGIRGZ 10.76
.A ABCL1 20020302 Z DH120000/HGIRGZ 11.99
.A ABCL1 20020302 Z DH180000/HGIRGZ 12.02
.A ABCL1 20020303 Z DH000000/HGIRGZ 11.89
.A ABCL1 20020303 Z DH060000/HGIRGZ 10.07
.A ABCL1 20020303 Z DH120000/HGIRGZ 9.25
.A ABCL1 20020303 Z DH180000/HGIRGZ 8.90
.A ABCL1 20020304 Z DH000000/HGIRGZ 8.68
.A ABCL1 20020304 Z DH060000/HGIRGZ 8.52
.A ABCL1 20020304 Z DH120000/HGIRGZ 8.41
.A ABCL1 20020304 Z DH180000/HGIRGZ 8.32
.A ABCL1 20020305 Z DH000000/HGIRGZ 8.27

NNNN

4.2 Example vfyfcst2shef output file

ZCZC WOHVERIFY DEF
TTAA00 KWOH 292222

.A ACML1 20020301 Z DC200202261432/DH000000/HGIFZZ 26.60
.A ACML1 20020301 Z DC200202261432/DH060000/HGIFZZ 26.50
.A ACML1 20020301 Z DC200202261432/DH120000/HGIFZZ 26.40
.A ACML1 20020301 Z DC200202271521/DH000000/HGIFZZ 26.00
.A ACML1 20020301 Z DC200202271521/DH060000/HGIFZZ 25.90
.A ACML1 20020301 Z DC200202271521/DH120000/HGIFZZ 25.80
.A ACML1 20020301 Z DC200202271521/DH180000/HGIFZZ 25.70
.A ACML1 20020302 Z DC200202271521/DH000000/HGIFZZ 25.60
.A ACML1 20020302 Z DC200202271521/DH060000/HGIFZZ 25.50
.A ACML1 20020302 Z DC200202271521/DH120000/HGIFZZ 25.50
.A ACML1 20020301 Z DC200202281555/DH000000/HGIFZZ 24.70
.A ACML1 20020301 Z DC200202281555/DH060000/HGIFZZ 24.60
.A ACML1 20020301 Z DC200202281555/DH120000/HGIFZZ 24.50
.A ACML1 20020301 Z DC200202281555/DH180000/HGIFZZ 24.40
.A ACML1 20020302 Z DC200202281555/DH000000/HGIFZZ 24.30
.A ACML1 20020302 Z DC200202281555/DH060000/HGIFZZ 24.20
.A ACML1 20020302 Z DC200202281555/DH120000/HGIFZZ 24.20

NNNN

5.0 Troubleshooting Information

The two applications, vfyobs2shef and vfyfcst2shef have a simple debug option. To turn debug option on, the user must edit the run_vfyobs2shef and/or run_vfyfcst2shef scripts and add an argument after the program name.... vfyobs2shef Y > debug1.out and/or vfyfcst2shef Y > debug2.out. If the application runs successfully the output written to screen can be quite lengthy, so it is recommended the user redirect the output into a file as shown above. The important part of this debug information is generally the lines related to database name and sql code values when the programs are not finding data. Excerpt from successful runs with debug option turned on are shown in sections 5.1 and 5.2 for each program.

5.1 Sample debug output for vfyobs2shef

```
sysmaster@ONLINE
sqlca.sqlcode open database 0
select tabname from systabnames where dbsname='vdb1_lorn' and
tabname matches '*obsht??*' order by 1
select completed error= 0
open tblcursor completed error= 0
obsht200203_1 ***
obsht200204_1 ***
obsht200205_1 ***
fetch tblcursor error code = 100
vdb1_lorn@ONLINE
sqlca.sqlcode open database 0
tdyyr=2002 iyear=2002 tdymo=9 imon=3
select * from obsht200203_1 order by 1,6;
select completed error= 0
open cursor completed error= 0
error code = 100
tdyyr=2002 iyear=2002 tdymo=9 imon=4
select * from obsht200204_1 order by 1,6;
select completed error= 0
open cursor completed error= 0
error code = 100
tdyyr=2002 iyear=2002 tdymo=9 imon=5
select * from obsht200205_1 order by 1,6;
select completed error= 0
open cursor completed error= 0
error code = 100

vfyobs2shef finished
```

5.2 Sample debug output for vfyfcst2shef

```
sysmaster@ONLINE
```

```
sqlca.sqlcode open database 0
select tabname from systabnames where dbsname='vdb1_lorn' and
tabname matches '*fcstht??*' order by 1
select completed error= 0
open tblcursor completed error= 0
fcstht200203_1 ***
fcstht200204_1 ***
fcstht200205_1 ***
fetch tblcursor error code = 100
vdb1_lorn@ONLINE
sqlca.sqlcode open database 0
tdyvr=2002 iyvr=2002 tdymo=9 imon=3
select * from fcstht200203_1 order by 1,7,8;
select completed error= 0
open cursor completed error= 0
error code = 100
tdyvr=2002 iyvr=2002 tdymo=9 imon=4
select * from fcstht200204_1 order by 1,7,8;
select completed error= 0
open cursor completed error= 0
error code = 100
tdyvr=2002 iyvr=2002 tdymo=9 imon=5
select * from fcstht200205_1 order by 1,7,8;
select completed error= 0
open cursor completed error= 0
error code = 100

vfyfcst2shef finished
```

6.0 Installation Instructions

“Under Construction”

7.0 Maintenance Information

Originating Programmer/Office: Meyer, A. Juliann
Missouri Basin River Forecast Center
Pleasant Hill MO

Maintenance programmer/Office: Meyer, A. Juliann
Missouri Basin River Forecast Center
Pleasant Hill MO

8.0 References

data dictionary for the archive database
data dictionary for the verify database

Attachment A

archive database vlocation table column datatype	Description
lid char(8)	location identifier
county char(20)	county name
elev float	elevation (ft msl)
hsa char(3)	Hydrologic Service Area

lrevise	date	date data revised.
name	char(25)	gage description
rb	char(30)	river basin
rfc	char(5)	River Forecast Center
state	char(2)	state Post Office code
wfo	char(3)	Weather Forecast Office
region	char(20)	NWS region headquarters

primary key lid	
-----------------	--

Attachment B

archive database vrivergaugeloc table column datatype	Description
lid char(8)	location identifier
bankfull_stg float	bankfull stage (ft)
warn_stg float	forecast issuance stage (ft)
action_stg float	action stage (ft)
fld_stg float	flood stage (ft)
mod_fld_stg float	moderate flood category stage (ft)
maj_fld_stg float	major flood category stage (ft)
rec_fld_stg float	stage of record (ft)
bankfull_flow float	bankfull stage flow (cfs)
warn_flow float	forecast issuance stage flood flow (cfs)
action_flow float	action stage flood flow (cfs)
fld_flow float	flood stage flood flow (cfs)
mod_fld_flow float	moderate flood flow (cfs)
maj_fld_flow float	major flood flow (cfs)
rec_fld_flow float	record flood flow (cfs)
flow_size char(6)	fast, medium or slow
sensor_1 char(2)	SHEF Type/Source code 1 st choice

sensor_2 char(2)	SHEF Type/Source code 2 nd choice
sensor_3 char(2)	SHEF Type/Source code 3 rd choice
pe_1 char(2)	SHEF PE code 1 st choice
pe_2 char(2)	SHEF PE code 2 nd choice
pe_3 char(2)	SHEF PE code 3 rd choice
pe_4 char(2)	SHEF PE code 4 th choice
primary key lid	

Attachment C

verify database addadjust table column datatype	archive database vaddadjust table column datatype	Description
lid char(8)	lid char(8)	location identifier
pe char(2)	pe1 char(1) pe2 char(1)	SHEF Physical Element codes
	dur char(1)	SHEF Duration code
dur smallint	idur smallint	numeric value of SHEF Duration code
ts char(2)	t s char(1) char(1)	SHEF Type-Source codes

extremum char(1)	e char(1)	SHEF Extremum code
adjustment float	adjustment float	value to add/subtract to stages
primary key lid, pe, dur, ts, extremum	primary key lid, pe1, pe2, dur, idur, t, s, e	