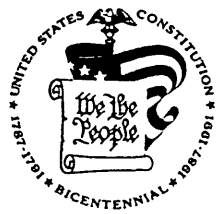
PROGRAM-MAINTENANCE MANUAL FOR THE CONVERSION AND VALIDATION OF UNIT-VALUES SITE-INDEX DATA FOR THE MASTER WATER DATA INDEX OF THE NATIONAL WATER INFORMATION SYSTEM OF THE U.S. GEOLOGICAL SURVEY

By Scott H. Beddingfield

U.S. GEOLOGICAL SURVEY

Open-File Report 87-556



Baton Rouge, Louisiana

1988

DEPARTMENT OF THE INTERIOR

DONALD PAUL HODEL, Secretary

U.S. GEOLOGICAL SURVEY

Dallas L. Peck, Director

For additional information write to:

District Chief
U.S. Geological Survey
P.O. Box 66492
Baton Rouge, LA 70896

Telephone: (504) 389-0281

U.S. Geological Survey
Branch of Computer
Technology, MS-440
12201 Sunrise Valley Dr.
Reston, VA 22092

Copies of this report can be purchased from:

U.S. Geological Survey
Books and Open-File Reports
Federal Center, Bldg. 810
Box 25425
Denver, CO 80225

CONTENTS

ILLUSTRATIONS

PROGRAM-MAINTENANCE MANUAL FOR THE CONVERSION AND VALIDATION OF UNIT-VALUES SITE-INDEX DATA FOR THE MASTER WATER DATA INDEX OF THE NATIONAL WATER INFORMATION SYSTEM OF THE U.S. GEOLOGICAL SURVEY

By Scott H. Beddingfield

ABSTRACT

Program-maintenance manuals are used to aid maintenance programmers when they must correct programing flaws or modify existing programs to adapt to changing needs or conditions. This program-maintenance manual provides a detailed description of the logic used in the program for the conversion and validation of unit-values site-index data for the Master Water Data Index (MWDI) of the National Water Information System (NWIS) developed by the U.S. Geological Survey. Included in this document are narratives that describe the processing logic, descriptions of input and output formats for the various files used in the program, a description of the necessary hardware to execute the program, and instructions on procedures for modifying the program if changes are necessary. In the appendices of this document are detailed input- and output-data formats for all of the files used by the program, a list of the data structures and variables used in the program, examples of the output and listings generated by the program, pseudo code describing the logic of the program, a visual table of contents (VTOC) organization chart, and data-flow diagrams, or cross-reference listing, and the source code.

INTRODUCTION

Background

The Master Water Data Index (MWDI) of the National Water Information System (NWIS) of the U.S. Geological Survey serves as a mechanism for documenting the existence and location of site-specific water data stored in the NWIS. The MWDI also contains information about water data and meteorological data available from other organizations throughout the scientific community. The existing MWDI data base of the U.S. Geological Survey's National Water Data Exchange (NAWDEX) is currently used for this purpose, but no longer fulfills all of the needs of the scientific community. The MWDI of the NWIS is a version of the MWDI of the NAWDEX that also contains data from the National Water Data Storage and Retrieval System (WATSTORE) Station Header File and Ground-Water Site Inventory (GWSI) that has been modified to satisfy those unmet needs. To populate the unit-values site-index data base of the MWDI of the NWIS with data from the MWDI of the NAWDEX and the WATSTORE files, a computer program was written.

The development of the program was sponsored by the Branch of Computer Technology, Office of the Assistant Chief Hydrologist for Scientific Information Management, Water Resources Division, U.S. Geological Survey and was developed under the direction of the NWIS Project Office within the Branch of Computer Technology.

Purpose and Scope

This document is for use by personnel responsible for monitoring the program during its life cycle. Contained in this document is information necessary for modifying and testing the program. This information includes (1) a general program description, (2) a description of the hardware environment necessary for the program to operate, (3) maintenance and verification procedures, (4) input- and output-data formats, (5) listings of the data structures and variables used in the program, (6) output-listing examples, (7) pseudo code, (8) a visual table of contents (VTOC) organization chart, and data-flow diagrams, (9) a cross-reference listing to link the VTOC organization chart, the subroutines in the source code, and the program design specifications as listed in Program Specifications for the Conversion and Validation of Unit-Values Site-Index Data for the MWDI of the NWIS (M.D. Edwards, U.S. Geological Survey, written commun., 1985), and (10) the source code.

SECTION 1. GENERAL INFORMATION

1.1 Summary

The program described in this document is used for converting unit-values site-index data and other surface-water data elements in the existing MWDI, WATSTORE Station Header File, and GWSI to formats suitable for establishment and loading of the MWDI of the NWIS; for the computation and validation of indexing information pertaining to unit values and other related surface-water data stored in the MWDI using data stored in the unit-values file of WATSTORE; for the computation of the frequencies of collection of unit-value data; and for the generation of transactions that document the existence of groundwater, quality-of-water, and meteorological data stored in the WATSTORE unit-values file for subsequent processing by other software programs. The specifications for the site-index data of the NWIS and those used to develop this program may be obtained by contacting the NWIS Project Office within the Branch of Computer Technology of the U.S. Geological Survey Headquarters Office in Reston, Virginia.

1.2 Hardware Environment

The unit-values conversion and validation program was developed for execution on the AMDAHL V7[a] computer located at the U.S. Geological Survey

[a] The use of product or trade names in this report is for identification purposes only, and does not constitute endorsement by the U.S. Geological Survey.

National Center in Reston, Virginia. The program is written in PL/1, level F, and was developed on the PRIME 750 minicomputer at the Louisiana District office of the U.S. Geological Survey and transferred to the AMDAHL computer for testing and execution.

SECTION 2. PROGRAM DESCRIPTION

This program is identified as program number IXCVUV.DB. This section discusses the details of the operation of the program.

2.1 Problem and Solution Method

Large volumes of data stored in WATSTORE have been previously indexed in the MWDI maintained by the NAWDEX. Before these data are transferred to the NWIS, it is necessary to validate that the data contained in WATSTORE have been accurately indexed, to validate that all sites for which data are stored in WATSTORE have been indexed, convert the existing data into formats and values that are compatible for entry into the new data base, and compute or derive values for many new indexing variables that have been added during the design of the NWIS. These functions are accomplished by comparing and validating the existing indexing information with the data stored in WATSTORE, translating and reformatting values of existing indexing variables to values and formats defined by the NWIS system specifications, and computing or deriving values of newly defined indexing variables from the data stored in WATSTORE.

2.2 Input

The program requires three files as input:

a. A file of MWDI data that is stored off-line on magnetic tape, and produced from program IXCVHD.DB (program for the conversion and validation of site-descriptor data for the MWDI). This file is named OLDMWDI in the program source code. The content and format of data records contained in this file are presented in table A-1 of appendix A.

b. WATSTORE unit values are retrieved with WATSTORE program H572, and stored off-line on magnetic tape. This file is named WATSTOR in the program source code. The content and format of data records contained in this file are presented in table A-2 of appendix A.

c. A direct-access file that is on magnetic disk containing the next available unique site identifier for each state generated as output from program IXCVHD.DB. This file is named NXTAVAL in the program source code. The content and format of data records contained in this file are presented in table A-3 of appendix A.

2.3 Processing

This section describes the processing features important to the maintenance programmer.

2.3.1 Processing Logic

The program is designed to take two input data sets, one consisting of MWDI records generated by program IXCVHD.DB, and the other consisting of records retrieved from the WATSTORE unit values data base using program H572, and generate a third data set, consisting of updated MWDI records generated by the program. The program also keeps track of several counters and statistics about itself, and provides listings of results produced by the program for validation purposes. All three data sets are sorted in ascending order by state, agency code, station number, year, and parameter code.

If the MWDI station number is smaller than the WATSTORE station number, or if there are no more WATSTORE records to be processed (end of file), the program takes parts of the MWDI input record and moves them to the MWDI output record, assigns surface water data base identifiers, and increments some counters. The program then writes out the MWDI output record, and reads in a new MWDI input record.

If the MWDI station number equals the WATSTORE station number, the program moves some items from the MWDI input record to the MWDI output record, and adds new elements. The program then enters a loop where the WATSTORE record is used to update the MWDI output record; a new WATSTORE record is read, and if its station number is equal to the MWDI station number, the new WATSTORE record is also used to update the MWDI output record. This process is repeated until the WATSTORE station number is not equal to the MWDI station number, or the WATSTORE data file is emptied. At that time, surface-water data base identifiers are set, and counters are incremented. The MWDI output record is then written, and a new MWDI input record is read.

During the looping mentioned above, if the station has surface-water records, a surface-water frequency history record is built. This record is updated during each pass through the loop as long as the state, agency code, station number, and year remain the same, and the parameter codes for each WATSTORE record processed are all surface-water codes. If any of these conditions are not met, the record is output to the surface-water frequency history file. In other words, a surface-water frequency history record is generated for each unique state/agency/station number/year combination.

The other data transactions file is also populated during this looping. If the particular station has quality of water, ground water, or meteorological records in WATSTORE, an other data transactions record is built. This record is updated with each pass through the loop as long as the state, agency code, station number, year, and parameter code remain the same. When any of these conditions are not met, the record is output to the file. In other words, an other data transaction record is generated for nonsurface-water records for each unique state/agency/station number/year/parameter code combination.

If the MWDI station number is larger than the WATSTORE station number, the program must build a new record for the WATSTORE data. The program gets the next available unique site identifier from an on-line disk file generated as output from program IXCVHD.DB. Then, the program enters a loop where the WATSTORE record is used to update the MWDI output record; a new WATSTORE record is read, and if its station number is equal to the last WATSTORE record station number, it is also used to update the MWDI output record. This process is repeated until the old WATSTORE record and the new WATSTORE station number do not match, or the WATSTORE data file is emptied. At that time, surface-water data base identifiers are set, and counters are incremented. Then an MWDI output record is written, and a new MWDI input record is read.

During the above mentioned loop, the surface-water frequency history file and the other data transaction file are populated according to the criteria outlined when the MWDI station number and WATSTORE station number are equal.

When both the MWDI input file and the WATSTORE file are empty, the program writes out all of the counters it has been keeping, and the program terminates.

2.3.2 Variables

The variables used in this program are listed in appendix C.

2.3.3 Error Handling Provisions

There are no error handling provisions built into the program. Any problems encountered should be handled using standard debugging techniques.

2.3.4 Restrictions and Limitations

This program is designed to be executed only after the successful execution of program IXCVHD.DB because program IXCVHD.DB provides necessary input data to this program.

2.4 Output

The output of this program are four files and a listing of counters, statistics, and data computed and generated during the program execution. The four files are the MWDI output data, the surface-water frequency history data, the ground-water, quality-of-water, and meteorological transactions data, and the next available unique identifier file. In the program, these files are named NEWMWDI, SWFRQHT, OTRTRNS, and NXTAVAL, respectively. Detailed tables of the content and structure of these data sets are in appendix B, except for the next available unique site identifier file, which is in table A-3 in appendix A.

The listings generated by this program contain surface-water data element counts for input and output, and statistics of the program run such as total number of sites in and out of the program, how many sites were added to the MWDI data base, and totals of ground-water and quality-water transactions. The listings also include samples of validated surface-water data (first 100 records where the agency is not equal to USGS, and the first 200 records where the agency is equal to USGS), and ground-water, quality-of-water, and meteorological data (first 200 records). Examples of all of the outputs and their formats are in appendix D.

2.5 Interfaces

This program is one of a series of programs which uses data generated by the previous program in the series, and generates data for the next program in the series. It is important that the data the program uses for input are in the format that the program can use, and that the data the program generates are in the format that the next program can use, according to the coordinator for Subsystem Specifications for the Conversion and Validation of Site-Index Data of the NWIS, and Program Specifications for the Conversion and Validation of Unit-Values Site-Index Data for the MWDI of the NWIS (M.D. Edwards, U.S. Geological Survey, written commun., 1985). These formats are shown in appendices A and B.

2.6 Run Description

The procedures to run this program are detailed in the document titled, "User's Manual for the Conversion and Validation of Unit-Values Site-Index Data for the Master Water Data Index of the National Water Information System," and figures 1 and 2 show the job control language (JCL) needed to compile and load, and to execute the program.

```
//XXXXXXXX JOB (NNNNNNNN,NWIS,,20),'NAME',CLASS=C
//PROCLIB  DD DSN=NWDX.PROCLIB,DISP=SHR
//STEP1 EXEC PLIXCL,SOURCE='AG40WAH.SOURCE',PROG=IXCVUV,
//       LOADLIB='AG40WAH.BINARY',
//        PLIPARM='AG,A(F),X(F),MARGINS(10,100,9),MI('''' '''')'
/*
//
```

Figure 1.--Job control language (JCL) used to compile and load the program IXCVUV.DB.

```
//xxxxxxxx JOB (nnnnnnnnn,NWIS,5,30),'name',CLASS=C
/*SETUP    224415/H
/*SETUP    110651/H
/*SETUP    222383/HR
/*SETUP    222384/HR
/*SETUP    222385/HR
//STEP1  EXEC PGM=IXCVUV,REGION=950K,TIME=(5,0)
//STEPLIB  DD DSN=AG40WAH.BINARY,DISP=SHR
//         DD DSN=SYS1.PLIX.TRANSLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//OLDMWDI  DD DSN=NWIS.IXCVUV.MWDI.INPUT,DISP=(OLD,KEEP),
//         UNIT=TAPE62,VOL=SER=224415,
//         DCB=(RECFM=FB,LRECL=1394,BLKSIZE=18122),LABEL=(1,SL)
//WATSTOR  DD DSN=NWIS.IXCVUV.UNIT.VALUES,DISP=(OLD,KEEP),
//         UNIT=TAPE62,VOL=SER=110651
//NXTAVAL  DD DSN=NWIS.IXCVHD.UIDHV,DCB=DSORG=IS,DISP=SHR,
//         UNIT=3350,VOL=SER=WRD513
//NEWMWDI  DD DSN=NWIS.IXCVUV.MWDI.OUTPUT,DISP=(OLD,KEEP),
//         UNIT=TAPE62,VOL=SER=222383,
//         DCB=(RECFM=FB,LRECL=1370,BLKSIZE=19180),
//         LABEL=(1,SL,RETPD=365)
//SWFRQHT  DD DSN=NWIS.IXCVUV.FREQ.HIST,DISP=(OLD,KEEP),
//         UNIT=TAPE62,VOL=SER=222384,
//         DCB=(RECFM=FB,LRECL=44,BLKSIZE=880),LABEL=(1,SL,RETPD=365)
//OTRTRNS  DD DSN=NWIS.IXCVUV.TRANS.DATA,DISP=(OLD,KEEP),
//         UNIT=TAPE62,VOL=SER=222385,
//         DCB=(RECFM=FB,LRECL=40,BLKSIZE=800),LABEL=(1,SL,RETPD=365)
/*
//
```

Figure 2.--Job control language (JCL) used to execute the program IXCVUV.DB.

SECTION 3. OPERATING ENVIRONMENT

3.1 Hardware

The following equipment and resources are used for executing the program.

3.1.1 Computer Hardware

The program will be executed on the AMDAHL V7 computer of the USGS located at its National Center in Reston, Virginia.

3.1.2 Data Storage

All data used as input to the program or produced as output from the program will be stored in sequential files on magnetic tape, except for the next available unique site identifier file, which is kept on-line in a direct access file.

3.1.3 Input/Output Devices

The program will require the following input/output devices:

- A terminal suitable for executing the software.
- Five magnetic tape drives: two input drives and three output drives.
- One magnetic disk drive for storing work files.
- One printer for producing printed output.

3.2 Support Software

The Unit Values Retrieval Program (program H572) is used to retrieve data from the WATSTORE Unit Values ISAM current file, and sequential backfile. Instructions for the use of program H572 are in Bower and Dempster (1978). The retrieved data will be stored on magnetic tape for subsequent processing.

SECTION 4. MAINTENANCE PROCEDURES

4.1 Verification Procedures

The following verification procedures should be used to ensure that the integrity of the data being produced is protected.

4.1.1 Verification of Production Runs

The performance of the program is monitored by checking the listings generated by the program. These listings contain counts of various input and output variables, a sample of the validated surface-water data, and a sample of the ground-water, quality-of-water, and meteorological data. Examples of the contents of each of these listings are in appendix D.

A complete and successful run is indicated by the presence of the data-element counts for a complete run. (An example of this listing is in example 2, appendix D.) This listing will immediately follow the listings of the data-element counts for each state.

4.1.2 Verification of Modifications to Program

Following any modification to the program, it must be verified to show that the change has done only what it was supposed to do. To ensure this, the program must be run through a low volume test and a high volume test, using verified data sets provided by the NWIS office, and the results must pass a review process. Only after the program has successfully completed this process can it be used for production work.

4.2 Error Correction Procedures

Any error conditions that arise are to be handled in the same manner outlined in 4.1.2: The error will be diagnosed, a fix will be applied, and the program will be rerun from the beginning.

4.3 Special Maintenance Procedures

Because this is a program designed to be used once, it neither has nor requires any special maintenance procedures.

4.4 Listings and Flowcharts

Appendix E contains pseudo code, appendix F contains a VTOC organization chart, and appendix G contains a cross-reference listing. These data can be used to link the source code with the program specifications. The source code is listed in appendix H, and the program specifications can be obtained by contacting the NWIS office of the Water Resources Division of the U.S. Geological Survey in Reston, Virginia.

SECTION 5. REFERENCES

Bower, D.E., and Dempster, G.R., Jr., 1978, Chapter III. Instructions for unit values file, Section B. Retrieval of data from unit values file (program H572), in WATSTORE User's Guide: U.S. Geological Survey Open-File Report 77-729, v. 5, chap. III, sec. B, 140 p.

U.S. Department of Commerce, 1976, Guidelines for documentation of computer programs and automated data systems: U.S. Department of Commerce, National Bureau of Standards, FIPS PUB 38, 55 p.

APPENDIX A. INPUT DATA FORMATS

Table A-1.--Format of Master Water Data Index (MWDI) input data records

Data Element Name	Format[1]	Byte Positions
Site Descriptor Data	---	---
Unique Site Identifier	Fixed Binary (31)	1 - 4
Agency Code	CHAR (5)	5 - 9
Agency Station Number	CHAR (15)	10 - 24
Local Well Number	CHAR (24)	25 - 48
Station Name	CHAR (48)	49 - 96
Latitude	Fixed Binary (31)	97 - 100
Longitude	Fixed Binary (31)	101 - 104
Lat-Long Accuracy	CHAR (1)	105
Site Type	CHAR (2)	106 - 107
Country Code	CHAR (2)	108 - 109
Reserved Space	CHAR (1)	110
State Code	Fixed Binary (15)	111 - 112
County Code	Fixed Binary (15)	113 - 114
Reserved Space	CHAR (2)	115 - 116
State-County Code	Fixed Binary (31)	117 - 120
District Code	CHAR (2)	121 - 122
Reserved Space	CHAR (2)	123 - 124
Hydrologic Unit Code	Fixed Binary (31)	125 - 128
Office Code	Fixed Binary (31)	129 - 132
Total Drainage Area	Float Dec (6)	133 - 136
Contributing Drainage Area	Float Dec (6)	137 - 140
Altitude (Datum)	Float Dec (6)	141 - 144
Altitude Method of Meas.	CHAR (1)	145
Reserved Space	CHAR (3)	146 - 148
Altitude Accuracy	Float Dec (6)	149 - 152
River Reach Number	Fixed Binary (15)	153 - 154
River Reach Segment Flag	Fixed Binary (15)	155 - 156
Depth of Well	Float Dec (6)	157 - 160
Source of Well Depth Data	CHAR (1)	161
Reserved Space	CHAR (3)	162 - 164
Depth of Hole	Float Dec (6)	165 - 168
Principal Aquifer Code	CHAR (8)	169 - 176
Aquifer Type Code	CHAR (1)	177
Consolidated Aquifer Code	CHAR (1)	178
Basin Descriptor	(3) CHAR (1)	179 - 181
Use of Water	(3) CHAR (1)	182 - 184
Use of Site	(3) CHAR (1)	185 - 187
Other Data Available	(6) CHAR (1)	188 - 193
SW Active Code	CHAR (1)	194
GW Active Code	CHAR (1)	195
QW Active Code	CHAR (1)	196
BIO Active Code	CHAR (1)	197

[1] Data formats are given in PL/1 notation.

Table A-1.--Format of Master Water Data Index (MWDI) input data records--Continued

Data Element Name	Format[1]	Byte Positions
Site Descriptor Data - Continued	---	---
PHY Active Code	CHAR (1)	198
SED Active Code	CHAR (1)	199
CHM Active Code	CHAR (1)	200
MET Active Code	CHAR (1)	201
Reserved Space	CHAR (3)	202 - 204
Date of Last Update	Fixed Binary (31)	205 - 208
Node Location Information	---	---
DIS Node Identifier	CHAR (6)	209 - 214
SW Data Code	CHAR (1)	215
GW Data Code	CHAR (1)	216
QW Data Code	CHAR (1)	217
MET Data Code	CHAR (1)	218
Other Data Code	CHAR (1)	219
Reserved Space	CHAR (1)	220
Node Date of Last Update	Fixed Binary (31)	221 - 224
Surface Water Data	---	---
SW Begin Year	PIC '999R'	225 - 228
SW End Year	PIC '999R'	229 - 232
SW Interrupted Code	CHAR (1)	233
Complete Stage	CHAR (1)	234
Peak Stage	CHAR (1)	235
Low Stage	CHAR (1)	236
Stage Storage Media	CHAR (1)	237
Complete Flow	CHAR (1)	238
Peak Flow	CHAR (1)	239
Low Flow	CHAR (1)	240
Miscellaneous Flow Meas.	CHAR (1)	241
Flow Storage Media	CHAR (1)	242
Volume	CHAR (1)	243
Volume Change	CHAR (1)	244
Volume Storage Media	CHAR (1)	245
Unit Flow	CHAR (1)	246
Unit Stage	CHAR (1)	247
Unit Volume	CHAR (1)	248
SW Other Data Available	CHAR (12)	249 - 260
SW Telemetry Code	PIC 'R'	261
SW Date of Last Update	CHAR (4)	262 - 265
SW Purpose Code	CHAR (9)	266 - 274
SW Recorder Type Code	CHAR (1)	275
SW Recorder Frequency Code	CHAR (1)	276

Table A-1.--Format of Master Water Data Index (MWDI) input data records--Continued

Data Element Name	Format[1]	Byte Positions
SW Modifiers (10)	---	277 - 356
SW Pointer	CHAR (7)	
SW Mod File	CHAR (1)	
Ground Water Data	---	---
GW Begin Year	PIC '999R'	357 - 360
GW End Year	PIC '999R'	361 - 364
GW Interrupted Code	CHAR (1)	365
Water-Level Frequency	CHAR (1)	366
Water-Level Storage Media	CHAR (1)	367
Discharge Frequency	CHAR (1)	368
Discharge Storage Media	CHAR (1)	369
Subsidence Frequency	CHAR (1)	370
Subsidence Storage Media	CHAR (1)	371
GW Other Data Available	CHAR (12)	372 - 383
Major Variations Code	CHAR (4)	384 - 387
GW Telemetry Code	PIC 'R'	388
GW Date of Last Update	CHAR (4)	389 - 392
GW Purpose Code	CHAR (9)	393 - 401
GW Recorder Type Code	CHAR (1)	402
GW Recorder Frequency Code	CHAR (1)	403
GW Modifiers (10)	---	404 - 483
GW Pointer	CHAR (7)	---
GW Mod File	CHAR (1)	---
Quality of Water Data	---	---
QW Begin Year	PIC '999R'	484 - 487
QW End Year	PIC '999R'	488 - 491
QW Interrupted Code	CHAR (1)	492
QW Telemetry Code	CHAR (1)	493
QW Date of Last Update	CHAR (4)	494 - 497
QW Purpose Code	CHAR (9)	498 - 506
QW Recorder Type Code	CHAR (1)	507
QW Recorder Frequency Code	CHAR (1)	508
STORET Pointer	CHAR (7)	509 - 515
QW Modifiers (10)	---	516 - 595
QW Pointer	CHAR (7)	---
QW Mod File	CHAR (1)	---

Table A-1.--Format of Master Water Data Index (MWDI) input data records--Continued

Data Element Name	Format[1]	Byte Positions
Biological Data	---	---
Enteric Bacteria	CHAR (1)	596
Native Bacteria	CHAR (1)	597
Phytoplankton	CHAR (1)	598
Zooplankton	CHAR (1)	599
Periphyton	CHAR (1)	600
Macrophyton	CHAR (1)	601
Microinvertebrates	CHAR (1)	602
Macroinvertebrates	CHAR (1)	603
Vertebrates	CHAR (1)	604
Fungi	CHAR (1)	605
Viruses	CHAR (1)	606
BIO Begin Year	CHAR (4)	607 - 610
BIO End Year	CHAR (4)	611 - 614
BIO Date of Last Update	CHAR (4)	615 - 618
BIO Storage Media	CHAR (1)	619
Primary Productivity	CHAR (1)	620
Secondary Productivity	CHAR (1)	621
Chemosynthetic Activity	CHAR (1)	622
Biostimulatory Test	CHAR (1)	623
Toxicity Test	CHAR (1)	624
Other Bio-Assay Test	CHAR (1)	625
Chemical Tissue Analysis	CHAR (1)	626
Histopathic Analysis	CHAR (1)	627
Other Tissue Analyses	CHAR (1)	628
BIO Modifiers (10)	---	629 - 708
BIO Pointer	CHAR (7)	---
BIO Mod File	CHAR (1)	---
QW Physical Data	---	---
Temperature	CHAR (1)	709
Specific Conductance	CHAR (1)	710
Turbidity	CHAR (1)	711
Color	CHAR (1)	712
Odor	CHAR (1)	713
pH	CHAR (1)	714
Suspended Solids	CHAR (1)	715
PHY Begin Year	CHAR (4)	716 - 719
PHY End Year	CHAR (4)	720 - 723
PHY Date of Last Update	CHAR (4)	724 - 727
PHY Storage Media	CHAR (1)	728

Table A-1.--Format of Master Water Data Index (MWDI) input data records--Continued

Data Element Name	Format[1]	Byte Positions
PHY Modifiers (10)	---	729 - 808
PHY Pointer	CHAR (7)	---
PHY Mod File	CHAR (1)	---
QW Sediment Data	---	---
Bed Load	CHAR (1)	809
Suspended Concentration	CHAR (1)	810
Total Concentration	CHAR (1)	811
Suspended Particle Size	CHAR (1)	812
Bed Particle Size	CHAR (1)	813
Suspended Sediment Discharge	CHAR (1)	814
Total Sediment Discharge	CHAR (1)	815
SED Begin Year	CHAR (4)	816 - 819
SED End Year	CHAR (4)	820 - 823
SED Date of Last Update	CHAR (4)	824 - 827
SED Storage Media	CHAR (1)	828
SED Modifiers (10)	---	829 - 908
SED Pointer	CHAR (7)	---
SED Mod File	CHAR (1)	---
QW Chemical Data	---	---
Dissolved Solids	CHAR (1)	909
Major Ions	CHAR (1)	910
Hardness	CHAR (1)	911
Silica	CHAR (1)	912
Phosphorus	CHAR (1)	913
Phosphorus Species	CHAR (1)	914
Nitrogen	CHAR (1)	915
Nitrogen Species	CHAR (1)	916
Detergents	CHAR (1)	917
Other Minor Constituents	CHAR (1)	918
Radioactivity	CHAR (1)	919
Radiochemical Species	CHAR (1)	920
Carbon	CHAR (1)	921
Organic Groups	CHAR (1)	922
Pesticide Species	CHAR (1)	923
Other Organic Species	CHAR (1)	924
Biochemical Oxygen Demand	CHAR (1)	925
Chemical Oxygen Demand	CHAR (1)	926
Dissolved Oxygen	CHAR (1)	927
Other Dissolved Gases	CHAR (1)	928

Table A-1.--Format of Master Water Data Index (MWDI) input data records--Continued

Data Element Name	Format[1]	Byte Positions
QW Chemical Data - Continued	---	---
CHM Begin Year	CHAR (4)	929 - 932
CHM End Year	CHAR (4)	933 - 936
CHM Date of Last Update	CHAR (4)	937 - 940
CHM Storage Media	CHAR (1)	941
CHM Modifiers (10)	---	942 - 1021
CHM Pointer	CHAR (7)	---
CHM Mod File	CHAR (1)	---
Projects (10)	---	1022 - 1071
WRD Project Number	CHAR (5)	---
Networks (10)	---	1072 - 1111
Network Code	CHAR (4)	---
Other Source (10)	---	1112 - 1161
Other Source Agency	CHAR (5)	---
Source Information (10)	---	1162 - 1281
Source File Identifier	CHAR (4)	---
Source File Agency	CHAR (8)	---
Meteorological Data	---	---
MET Begin Year	CHAR (4)	1282 - 1285
MET End Year	CHAR (4)	1286 - 1289
MET Interrupted Code	CHAR (1)	1290
Rainfall	CHAR (1)	1291
Unit Rainfall	CHAR (1)	1292
Air Temperature	CHAR (1)	1293
Wind Velocity	CHAR (1)	1294
MET Other Data Available	CHAR (12)	1295 - 1306
MET Telemetry Code	CHAR (1)	1307
MET Date of Last Update	CHAR (4)	1308 - 1311
MET Storage Media	CHAR (1)	1312
MET Recorder Type Code	CHAR (1)	1313
MET Recorder Frequency Code	CHAR (1)	1314
MET Modifiers (10)	---	1315 - 1394
MET Pointer	CHAR (7)	---
MET Mod File	CHAR (1)	---

Table A-2.--Format of National Water Data Storage and Retrieval System (WATSTORE) unit values file input records

Data Element Name	Format[1]	Byte Positions
Reserved Space	CHAR (1)	1
Record Format	CHAR (1)	2
State Code	CHAR (2)	3 - 4
Agency Code	CHAR (5)	5 - 9
Station Identifier	CHAR (15)	10 - 24
Cross Section Locator	Float Dec (6)	25 - 28
Depth Locator	Float Dec (6)	29 - 32
Parameter Code	Fixed Binary (31)	33 - 36
Year Number	Fixed Binary (15)	37 - 38
Month Number	Fixed Binary (15)	39 - 40
Day Number	Fixed Binary (15)	41 - 42
Statistic Code	Fixed Binary (15)	43 - 44
Readings Per Day	Fixed Binary (15)	45 - 46
First Reading Position	Fixed Binary (15)	47 - 48
Number of Readings	Fixed Binary (15)	49 - 50
Retrieval Sequence Number	Fixed Binary (15)	51 - 52
Reserved Space	CHAR (9)	53 - 61
Site Code	CHAR (2)	62 - 63
District Code	CHAR (2)	64 - 65
County Code	CHAR (3)	66 - 68
Create Date	Fixed Binary (31)	69 - 72
Reserved Space	CHAR (4)	73 - 76
Missing Value Indicator	Float Dec (6)	77 - 80
Unit Values	Float Dec (6)	81 - **

[1] Data formats are given in PL/1 notation.

** - The unit values are stored in a variable-length array. The length of the array is determined by the value stored in byte positions 49 - 50 (number of readings). For example, if the number of readings equals 24, the unit values will be stored in byte positions 81 - 176. The maximum record length is 11,600 bytes (2880 readings per day). See the WATSTORE, 1978, User's Guide, v. 5, chap. III, sec. III.3, p. 31 for a further description of data storage in the record.

NOTE: If all unit values stored in the record are equal to the missing value indicator (byte positions 77 - 80) or the value 999998, the record will be rejected.

Table A-3.--Format of unique site identifier input/output records

Data Element Name	Data Format[1]	Byte Positions
State Code	Fixed Binary (31)	1 - 4
Next Available Unique Site Id	Fixed Binary (31)	5 - 8

[1] Data formats are given in PL/1 notation.

APPENDIX B. OUTPUT DATA FORMATS

Table B-1.--Format of Master Water Data Index (MWDI) output data records

Data Element Name	Data Format[1]	Byte Positions
Site Descriptor Data	---	---
Unique Site Identifier	Fixed Binary (31)	1 - 4
Agency Code	CHAR (5)	5 - 9
Agency Station Number	CHAR (15)	10 - 24
Local Well Number	CHAR (24)	25 - 48
Station Name	CHAR (48)	49 - 96
Latitude	Fixed Binary (31)	97 - 100
Longitude	Fixed Binary (31)	101 - 104
Lat-Long Accuracy	CHAR (1)	105
Site Type	CHAR (2)	106 - 107
Country Code	CHAR (2)	108 - 109
Reserved Space	CHAR (1)	110
State Code	Fixed Binary (15)	111 - 112
County Code	Fixed Binary (15)	113 - 114
Reserved Space	CHAR (2)	115 - 116
State-County Code	Fixed Binary (31)	117 - 120
District Code	CHAR (2)	121 - 122
Reserved Space	CHAR (2)	123 - 124
Hydrologic Unit Code	Fixed Binary (31)	125 - 128
Office Code	Fixed Binary (31)	129 - 132
Total Drainage Area	Float Dec (6)	133 - 136
Contributing Drainage Area	Float Dec (6)	137 - 140
Altitude (Datum)	Float Dec (6)	141 - 144
Altitude Method of Meas.	CHAR (1)	145
Reserved Space	CHAR (3)	146 - 148
Altitude Accuracy	Float Dec (6)	149 - 152
River Reach Number	Fixed Binary (15)	153 - 154
River Reach Segment Flag	Fixed Binary (15)	155 - 156
Depth of Well	Float Dec (6)	157 - 160
Source of Well Depth Data	CHAR (1)	161
Reserved Space	CHAR (3)	162 - 164
Depth of Hole	Float Dec (6)	165 - 168
Principal Aquifer Code	CHAR (8)	169 - 176
Aquifer Type Code	CHAR (1)	177
Consolidated Aquifer Code	CHAR (1)	178
Basin Descriptor	(3) CHAR (1)	179 - 181
Use of Water	(3) CHAR (1)	182 - 184
Use of Site	(3) CHAR (1)	185 - 187
Other Data Available	(6) CHAR (1)	188 - 193
SW Active Code	CHAR (1)	194
GW Active Code	CHAR (1)	195
QW Active Code	CHAR (1)	196
BIO Active Code	CHAR (1)	197

[1] Data formats are given in PL/1 notation.

Table B-1.--Format of Master Water Data Index (MWDI) output data records--Continued

Data Element Name	Data Format[1]	Byte Positions
Site Descriptor Data - Continued	---	---
PHY Active Code	CHAR (1)	198
SED Active Code	CHAR (1)	199
CHM Active Code	CHAR (1)	200
MET Active Code	CHAR (1)	201
Reserved Space	CHAR (3)	202 - 204
Date of Last Update	Fixed Binary (31)	205 - 208
Node Location Information	---	---
DIS Node Identifier	CHAR (6)	209 - 214
SW Data Code	CHAR (1)	215
GW Data Code	CHAR (1)	216
QW Data Code	CHAR (1)	217
MET Data Code	CHAR (1)	218
Other Data Code	CHAR (1)	219
Reserved Space	CHAR (1)	220
Node Date of Last Update	Fixed Binary (31)	221 - 224
Surface Water Data	---	---
SW Begin Year	PIC '999R'	225 - 228
SW End Year	PIC '999R'	229 - 232
SW Interrupted Code	CHAR (1)	233
Complete Stage	CHAR (1)	234
Peak Stage	CHAR (1)	235
Low Stage	CHAR (1)	236
Stage Storage Media	CHAR (1)	237
Complete Flow	CHAR (1)	238
Peak Flow	CHAR (1)	239
Low Flow	CHAR (1)	240
Miscellaneous Flow Meas.	CHAR (1)	241
Flow Storage Media	CHAR (1)	242
Volume	CHAR (1)	243
Volume Change	CHAR (1)	244
Volume Storage Media	CHAR (1)	245
Stage Type	CHAR (2)	246 - 247
Stage Frequency Code	CHAR (1)	248
Flow Type	CHAR (2)	249 - 250
Flow Frequency Code	CHAR (1)	251
Volume Type	CHAR (2)	252 - 253
Volume Frequency Code	CHAR (1)	254
Velocity Type	CHAR (2)	255 - 256
Velocity Frequency Code	CHAR (1)	257

Table B-1.--Format of Master Water Data Index (MWDI) output data records--Continued

Data Element Name	Data Format[1]	Byte Positions
Surface Water Data - Continued	---	---
Basin Characteristics Data Availability Code	CHAR (1)	258
SW Data Storage Media	CHAR (1)	259
SW Telemetry Code	CHAR (1)	260
SW Recorder Type Code	CHAR (1)	261
SW Data Purpose Code	(4) CHAR (1)	262 - 265
Other Surface-Water Related Data	(12) CHAR (1)	266 - 277
SW Seasonal Code	CHAR (1)	278
Reserved Space	CHAR (2)	279 - 280
SW Date of Last Update	Fixed Binary (31)	281 - 284
SW Data Base Identifiers (4)	---	285 - 332
SW Data Base Identifier	CHAR (4)	---
SW Data Base Subset Ident.	CHAR (4)	---
SW Data Base Date of Last Update	Fixed Binary (31)	---
Ground Water Data	---	---
GW Begin Year	PIC '999R'	333 - 336
GW End Year	PIC '999R'	337 - 340
GW Interrupted Code	CHAR (1)	341
Water-Level Frequency	CHAR (1)	342
Water-Level Storage Media	CHAR (1)	343
Discharge Frequency	CHAR (1)	344
Discharge Storage Media	CHAR (1)	345
Subsidence Frequency	CHAR (1)	346
Subsidence Storage Media	CHAR (1)	347
GW Other Data Available	CHAR (12)	348 - 359
Major Variations Code	CHAR (4)	360 - 363
GW Telemetry Code	PIC 'R'	364
GW Date of Last Update	CHAR (4)	365 - 368
GW Purpose Code	CHAR (9)	369 - 377
GW Recorder Type Code	CHAR (1)	378
GW Recorder Frequency Code	CHAR (1)	379
GW Modifiers (10)	---	380 - 459
GW Pointer	CHAR (7)	---
GW Mod File	CHAR (1)	---

Table B-1.--Format of Master Water Data Index (MWDI) output data records--Continued

Data Element Name	Data Format[1]	Byte Positions
Quality of Water Data	---	---
QW Begin Year	PIC '999R'	460 - 463
QW End Year	PIC '999R'	464 - 467
QW Interrupted Code	CHAR (1)	468
QW Telemetry Code	CHAR (1)	469
QW Date of Last Update	CHAR (4)	470 - 473
QW Purpose Code	CHAR (9)	474 - 482
QW Recorder Type Code	CHAR (1)	483
QW Recorder Frequency Code	CHAR (1)	484
STORET Pointer	CHAR (7)	485 - 491
QW Modifiers (10)	---	492 - 571
QW Pointer	CHAR (7)	---
QW Mod File	CHAR (1)	---
Biological Data	---	---
Enteric Bacteria	CHAR (1)	572
Native Bacteria	CHAR (1)	573
Phytoplankton	CHAR (1)	574
Zooplankton	CHAR (1)	575
Periphyton	CHAR (1)	576
Macrophyton	CHAR (1)	577
Microinvertebrates	CHAR (1)	578
Macroinvertebrates	CHAR (1)	579
Vertebrates	CHAR (1)	580
Fungi	CHAR (1)	581
Viruses	CHAR (1)	582
BIO Begin Year	CHAR (4)	583 - 586
BIO End Year	CHAR (4)	587 - 590
BIO Date of Last Update	CHAR (4)	591 - 594
BIO Storage Media	CHAR (1)	595
Primary Productivity	CHAR (1)	596
Secondary Productivity	CHAR (1)	597
Chemosynthetic Activity	CHAR (1)	598
Biostimulatory Test	CHAR (1)	599
Toxicity Test	CHAR (1)	600
Other Bio-Assay Test	CHAR (1)	601
Chemical Tissue Analysis	CHAR (1)	602
Histopathic Analysis	CHAR (1)	603
Other Tissue Analyses	CHAR (1)	604
BIO Modifiers (10)	---	605 - 684
BIO Pointer	CHAR (7)	---
BIO Mod File	CHAR (1)	---

Table B-1.--Format of Master Water Data Index (MWDI) output data records--Continued

Data Element Name	Data Format[1]	Byte Positions
QW Physical Data	---	---
Temperature	CHAR (1)	685
Specific Conductance	CHAR (1)	686
Turbidity	CHAR (1)	687
Color	CHAR (1)	688
Odor	CHAR (1)	689
pH	CHAR (1)	690
Suspended Solids	CHAR (1)	691
PHY Begin Year	CHAR (4)	692 - 695
PHY End Year	CHAR (4)	696 - 699
PHY Date of Last Update	CHAR (4)	700 - 703
PHY Storage Media	CHAR (1)	704
PHY Modifiers (10)	---	705 - 784
PHY Pointer	CHAR (7)	---
PHY Mod File	CHAR (1)	---
QW Sediment Data	---	---
Bed Load	CHAR (1)	785
Suspended Concentration	CHAR (1)	786
Total Concentration	CHAR (1)	787
Suspended Particle Size	CHAR (1)	788
Bed Particle Size	CHAR (1)	789
Suspended Sediment Discharge	CHAR (1)	790
Total Sediment Discharge	CHAR (1)	791
SED Begin Year	CHAR (4)	792 - 795
SED End Year	CHAR (4)	794 - 799
SED Date of Last Update	CHAR (4)	800 - 803
SED Storage Media	CHAR (1)	804
SED Modifiers (10)	---	805 - 884
SED Pointer	CHAR (7)	---
SED Mod File	CHAR (1)	---
QW Chemical Data	---	---
Dissolved Solids	CHAR (1)	885
Major Ions	CHAR (1)	886
Hardness	CHAR (1)	887
Silica	CHAR (1)	888
Phosphorus	CHAR (1)	889
Phosphorus Species	CHAR (1)	890

Table B-1.--Format of Master Water Data Index (MWDI) output data records--Continued

Data Element Name	Data Format[1]	Byte Positions
QW Chemical Data - Continued	---	---
Nitrogen	CHAR (1)	891
Nitrogen Species	CHAR (1)	892
Detergents	CHAR (1)	893
Other Minor Constituents	CHAR (1)	894
Radioactivity	CHAR (1)	895
Radiochemical Species	CHAR (1)	896
Carbon	CHAR (1)	897
Organic Groups	CHAR (1)	898
Pesticide Species	CHAR (1)	899
Other Organic Species	CHAR (1)	900
Biochemical Oxygen Demand	CHAR (1)	901
Chemical Oxygen Demand	CHAR (1)	902
Dissolved Oxygen	CHAR (1)	903
Other Dissolved Gases	CHAR (1)	904
CHM Begin Year	CHAR (4)	905 - 908
CHM End Year	CHAR (4)	909 - 912
CHM Date of Last Update	CHAR (4)	913 - 916
CHM Storage Media	CHAR (1)	917
CHM Modifiers (10)	---	918 - 997
CHM Pointer	CHAR (7)	---
CHM Mod File	CHAR (1)	---
Projects (10)	---	998 - 1047
WRD Project Number	CHAR (5)	---
Networks (10)	---	1048 - 1087
Network Code	CHAR (4)	---
Other Source (10)	---	1088 - 1137
Other Source Agency	CHAR (5)	---
Source Information (10)	---	1138 - 1257
Source File Identifier	CHAR (4)	---
Source File Agency	CHAR (8)	---
Meteorological Data	---	---
MET Begin Year	CHAR (4)	1258 - 1261
MET End Year	CHAR (4)	1262 - 1265
MET Interrupted Code	CHAR (1)	1266
Rainfall	CHAR (1)	1267
Unit Rainfall	CHAR (1)	1268

Table B-1.--Format of Master Water Data Index (MWDI) output data records--Continued

Data Element Name	Data Format[1]	Byte Positions
Meteorological Data - Continued	---	---
Air Temperature	CHAR (1)	1269
Wind Velocity	CHAR (1)	1270
MET Other Data Available	CHAR (12)	1271 - 1282
MET Telemetry Code	CHAR (1)	1283
MET Date of Last Update	CHAR (4)	1284 - 1287
MET Storage Media	CHAR (1)	1288
MET Recorder Type Code	CHAR (1)	1289
MET Recorder Frequency Code	CHAR (1)	1290
MET Modifiers (10)	---	1291 - 1370
MET Pointer	CHAR (7)	---
MET Mod File	CHAR (1)	---

Table B-2.--Format of the surface-water frequency history output records

Data Element Name	Data Format[1]	Byte Positions
Unique Site Identifier	Fixed Binary (31)	1 - 4
Agency Code	CHAR (5)	5 - 9
Agency Station Number	CHAR (15)	10 - 24
SW Year of Measurement	Fixed Binary (31)	25 - 28
Stage Type History	CHAR (2)	29 - 30
Stage Frequency History	CHAR (1)	31
Flow Type History	CHAR (2)	32 - 33
Flow Frequency History	CHAR (1)	34
Volume Type History	CHAR (2)	35 - 36
Volume Frequency History	CHAR (1)	37
Velocity Type History	CHAR (2)	38 - 39
Velocity Frequency History	CHAR (1)	40
SW History Date of Last Update	Fixed Binary (31)	41 - 44

[1] Data formats are given in PL/1 notation.

Table B-3.--Format of ground-water, quality-of-water, and meteorological transaction output records

Data Element Name	Data Format[1]	Byte Positions
Unique Site Identifier	Fixed Binary (31)	1 - 4
Agency Code	CHAR (5)	5 - 9
Agency Station Number	CHAR (15)	10 - 24
Parameter Code	Fixed Binary (31)	25 - 28
Year of Measurement	Fixed Binary (15)	29 - 30
Frequency Code	CHAR (1)	31
Seasonal Code	CHAR (1)	32
Data Base Identifier	CHAR (4)	33 - 36
Data Base Subset Identifier	CHAR (4)	37 - 40

[1] Data formats are given in PL/1 notation.

APPENDIX C. DATA STRUCTURES AND PROGRAM VARIABLES

Structure 1.--Data structure variables for Master Water Data Index (MWDI) input file.

```
1 MWDI_INPUT,
  2 SITE_DESCRIPTOR_DATA,
    3 SITE_ID FIXED BINARY(31),
    3 AGENCY_CODE CHAR(5),
    3 AGENCY_STATION_NO CHAR(15),
    3 LOCAL_WELL_NO CHAR(24),
    3 STATION_NAME CHAR(48),
    3 LATITUDE FIXED BINARY(31),
    3 LONGITUDE FIXED BINARY(31),
    3 LAT_LONG_ACCURACY CHAR(1),
    3 SITE_TYPE CHAR(2),
    3 COUNTRY_CODE CHAR(2),
    3 RESERVED1 CHAR(1),
    3 STATE_CODE FIXED BINARY(15),
    3 COUNTY_CODE FIXED BINARY(15),
    3 RESERVED2 CHAR(2),
    3 STATE_COUNTY_CODE FIXED BINARY(31),
    3 DISTRICT_CODE CHAR(2),
    3 RESERVED3 CHAR(2),
    3 HYDRO_UNIT_CODE FIXED BINARY(31),
    3 OFFICE_CODE FIXED BINARY(31),
    3 TOTAL_DRAINAGE_AREA FLOAT DEC(6),
    3 CONTRIB_DRAINAGE_AREA FLOAT DEC(6),
    3 ALTITUDE_DATUM FLOAT DEC(6),
    3 ALTITUDE_METH_OF_MEAS CHAR(1),
    3 RESERVED4 CHAR(3),
    3 ALTITUDE_ACCURACY FLOAT DEC(6),
    3 RIVER_REACH_NO FIXED BINARY(15),
    3 RIVER_REACH_SEG_FLAG FIXED BINARY(15),
    3 DEPTH_OF_WELL FLOAT DEC(6),
    3 SOURCE_WELL_DEPTH_DATA CHAR(1),
    3 RESERVED5 CHAR(3),
    3 DEPTH_OF_HOLE FLOAT DEC(6),
    3 PRINCIPAL_AQUIFER_CODE CHAR(8),
    3 AQUIFER_TYPE_CODE CHAR(1),
    3 CONSOLIDATED_AQUIFER_CODE CHAR(1),
    3 BASIN_DESCRIPTOR(3) CHAR(1),
    3 USE_OF_WATER(3) CHAR(1),
    3 USE_OF_SITE(3) CHAR(1),
    3 OTHER_DATA_AVAILABLE(6) CHAR(1),
    3 SW_ACTIVE_CODE CHAR(1),
    3 GW_ACTIVE_CODE CHAR(1),
    3 QW_ACTIVE_CODE CHAR(1),
    3 BIO_ACTIVE_CODE CHAR(1),
    3 PHY_ACTIVE_CODE CHAR(1),
    3 SED_ACTIVE_CODE CHAR(1),
    3 CHM_ACTIVE_CODE CHAR(1),
    3 MET_ACTIVE_CODE CHAR(1),
    3 RESERVED6 CHAR(3),
    3 DATE_OF_LAST_UPDATE FIXED BINARY(31),
```

```
2 NODE_LOCATION_INFO,
  3 DIS_NODE_IDENT CHAR(6),
  3 SW_DATA_CODE CHAR(1),
  3 GW_DATA_CODE CHAR(1),
  3 QW_DATA_CODE CHAR(1),
  3 MET_DATA_CODE CHAR(1),
  3 OTHER_DATA_CODE CHAR(1),
  3 RESERVED7 CHAR(1),
  3 NODE_DATE_OF_LAST_UPDATE FIXED BINARY(31),
2 SURFACE_WATER_DATA,
  3 SW_BEGIN_YEAR PIC '999R',
  3 SW_END_YEAR PIC '999R',
  3 SW_INTERRUPTED_CODE CHAR(1),
  3 COMPLETE_STAGE CHAR(1),
  3 PEAK_STAGE CHAR(1),
  3 LOW_STAGE CHAR(1),
  3 STAGE_STORAGE_MEDIA CHAR(1),
  3 COMPLETE_FLOW CHAR(1),
  3 PEAK_FLOW CHAR(1),
  3 LOW_FLOW CHAR(1),
  3 MISC_FLOW_MEAS CHAR(1),
  3 FLOW_STORAGE_MEDIA CHAR(1),
  3 VOLUME CHAR(1),
  3 VOLUME_CHANGE CHAR(1),
  3 VOLUME_STORAGE_MEDIA CHAR(1),
  3 UNIT_FLOW CHAR(1),
  3 UNIT_STAGE CHAR(1),
  3 UNIT_VOLUME CHAR(1),
  3 SW_OTHER_DATA_AVAIL CHAR(12),
  3 SW_TELEMETRY_CODE PIC 'R',
  3 SW_DATE_OF_LAST_UPDATE CHAR(4),
  3 SW_PURPOSE_CODE CHAR(9),
  3 SW_RECORDER_TYPE_CODE CHAR(1),
  3 SW_RECORDER_FREQ_CODE CHAR(1),
2 SW_MODIFIERS(10),
  3 SW_POINTER CHAR(7),
  3 SW_MOD_FILE CHAR(1),
2 GROUND_WATER_DATA,
  3 GW_BEGIN_YEAR PIC '999R',
  3 GW_END_YEAR PIC '999R',
  3 GW_INTERRUPTED_CODE CHAR(1),
  3 WATER_LEVEL_FREQ CHAR(1),
  3 WATER_LEVEL_STORAGE_MEDIA CHAR(1),
  3 DISCHARGE_FREQUENCY CHAR(1),
  3 DISCHARGE_STORAGE_MEDIA CHAR(1),
  3 SUBSIDENCE_FREQ CHAR(1),
  3 SUBSIDENCE_STORAGE_MEDIA CHAR(1),
  3 GW_OTHER_DATA_AVAILABLE CHAR(12),
  3 MAJOR_VARIATIONS_CODE CHAR(4),
  3 GW_TELEMETRY_CODE PIC 'R',
  3 GW_DATE_OF_LAST_UPDATE CHAR(4),
  3 GW_PURPOSE_CODE CHAR(9),
```

```
  3 GW_RECORDER_TYPE_CODE CHAR(1),
  3 GW_RECORDER_FREQ_CODE CHAR(1),
2 GW_MODIFIERS(10),
  3 GW_POINTER CHAR(7),
  3 GW_MOD_FILE CHAR(1),
2 QUALITY_OF_WATER_DATA,
  3 QW_BEGIN_YEAR PIC '999R',
  3 QW_END_YEAR PIC '999R',
  3 QW_INTERRUPTED_CODE CHAR(1),
  3 QW_TELEMETRY_CODE CHAR(1),
  3 QW_DATE_OF_LAST_UPDATE CHAR(4),
  3 QW_PURPOSE_CODE CHAR(9),
  3 QW_RECORDER_TYPE_CODE CHAR(1),
  3 QW_RECORDER_FREQ_CODE CHAR(1),
  3 STORET_POINTER CHAR(7),
2 QW_MODIFIERS(10),
  3 QW_POINTER CHAR(7),
  3 QW_MOD_FILE CHAR(1),
2 BIOLOGICAL_DATA,
  3 ENTERIC_BACTERIA CHAR(1),
  3 NATIVE_BACTERIA CHAR(1),
  3 PHYTOPLANKTON CHAR(1),
  3 ZOOPLANKTON CHAR(1),
  3 PERIPHYTON CHAR(1),
  3 MACROPHYTON CHAR(1),
  3 MICROINVERTEBRATES CHAR(1),
  3 MACROINVERTEBRATES CHAR(1),
  3 VERTEBRATES CHAR(1),
  3 FUNGI CHAR(1),
  3 VIRUSES CHAR(1),
  3 BIO_BEGIN_YEAR CHAR(4),
  3 BIO_END_YEAR CHAR(4),
  3 BIO_DATE_OF_LAST_UPDATE CHAR(4),
  3 BIO_STORAGE_MEDIA CHAR(1),
  3 PRIMARY_PRODUCTIVITY CHAR(1),
  3 SECONDARY_PRODUCTIVITY CHAR(1),
  3 CHEMOSYNTHETIC_ACTIVITY CHAR(1),
  3 BIOSTIMULATORY_TEST CHAR(1),
  3 TOXICITY_TEST CHAR(1),
  3 OTHER_BIO_ASSAY_TEST CHAR(1),
  3 CHEMICAL_TISSUE_ANALYSIS CHAR(1),
  3 HISTOPATHIC_ANALYSIS CHAR(1),
  3 OTHER_TISSUE_ANALYSES CHAR(1),
2 BIO_MODIFIERS(10),
  3 BIO_POINTER CHAR(7),
  3 BIO_MOD_FILE CHAR(1),
2 QW_PHYSICAL_DATA,
  3 TEMPERATURE CHAR(1),
  3 SPECIFIC_CONDUCTANCE CHAR(1),
  3 TURBIDITY CHAR(1),
  3 COLOR CHAR(1),
```

```
   3 ODOR CHAR(1),
   3 PH CHAR(1),
   3 SUSPENDED_SOLIDS CHAR(1),
   3 PHY_BEGIN_YEAR CHAR(4),
   3 PHY_END_YEAR CHAR(4),
   3 PHY_DATE_OF_LAST_UPDATE CHAR(4),
   3 PHY_STORAGE_MEDIA CHAR(1),
 2 PHY_MODIFIERS(10),
   3 PHY_POINTER CHAR(7),
   3 PHY_MOD_FILE CHAR(1),
 2 QW_SEDIMENT_DATA,
   3 BED_LOAD CHAR(1),
   3 SUSPENDED_CONC CHAR(1),
   3 TOTAL_CONC CHAR(1),
   3 SUSPENDED_PARTICLE_SIZE CHAR(1),
   3 BED_PARTICLE_SIZE CHAR(1),
   3 SUSPENDED_SEDIMENT_DISCH CHAR(1),
   3 TOTAL_SEDIMENT_DISCH CHAR(1),
   3 SED_BEGIN_YEAR CHAR(4),
   3 SED_END_YEAR CHAR(4),
   3 SED_DATE_OF_LAST_UPDATE CHAR(4),
   3 SED_STORAGE_MEDIA CHAR(1),
 2 SED_MODIFIERS(10),
   3 SED_POINTER CHAR(7),
   3 SED_MOD_FILE CHAR(1),
 2 QW_CHEMICAL_DATA,
   3 DISSOLVED_SOLIDS CHAR(1),
   3 MAJOR_IONS CHAR(1),
   3 HARDNESS CHAR(1),
   3 SILICA CHAR(1),
   3 PHOSPHORUS CHAR(1),
   3 PHOSPHORUS_SPECIES CHAR(1),
   3 NITROGEN CHAR(1),
   3 NITROGEN_SPECIES CHAR(1),
   3 DETERGENTS CHAR(1),
   3 OTHER_MINOR_CONSTITUENTS CHAR(1),
   3 RADIOACTIVITY CHAR(1),
   3 RADIOCHEMICAL_SPECIES CHAR(1),
   3 CARBON CHAR(1),
   3 ORGANIC_GROUPS CHAR(1),
   3 PESTICIDE_SPECIES CHAR(1),
   3 OTHER_ORGANIC_SPECIES CHAR(1),
   3 BIOCHEMICAL_OXYGEN_DEMAND CHAR(1),
   3 CHEMICAL_OXYGEN_DEMAND CHAR(1),
   3 DISSOLVED_OXYGEN CHAR(1),
   3 OTHER_DISSOLVED_GASES CHAR(1),
   3 CHM_BEGIN_YEAR CHAR(4),
   3 CHM_END_YEAR CHAR(4),
   3 CHM_DATE_OF_LAST_UPDATE CHAR(4),
   3 CHM_STORAGE_MEDIA CHAR(1),
 2 CHM_MODIFIERS(10),
```

```
   3 CHM_POINTER CHAR(7),
   3 CHM_MOD_FILE CHAR(1),
 2 PROJECTS(10),
   3 WRD_PROJECT_NUMBER CHAR(5),
 2 NETWORKS(10),
   3 NETWORK_CODE CHAR(4),
 2 OTHER_SOURCE(10),
   3 OTHER_SOURCE_AGENCY CHAR(5),
 2 SOURCE_INFORMATION(10),
   3 SOURCE_FILE_IDENTIFIER CHAR(4),
   3 SOURCE_FILE_AGENCY CHAR(8),
 2 METEOROLOGICAL_DATA,
   3 MET_BEGIN_DATE CHAR(4),
   3 MET_END_DATE CHAR(4),
   3 MET_INTERRUPTED_CODE CHAR(1),
   3 RAINFALL CHAR(1),
   3 UNIT_RAINFALL CHAR(1),
   3 AIR_TEMPERATURE CHAR(1),
   3 WIND_VELOCITY CHAR(1),
   3 MET_OTHER_DATA_AVAILABLE CHAR(12),
   3 MET_TELEMETRY_CODE CHAR(1),
   3 MET_DATE_OF_LAST_UPDATE CHAR(4),
   3 MET_STORAGE_MEDIA CHAR(1),
   3 MET_RECORDER_TYPE_CODE CHAR(1),
   3 MET_RECORDER_FREQ_CODE CHAR(1),
 2 MET_MODIFIERS(10),
   3 MET_POINTER CHAR(7),
   3 MET_MOD_FILE CHAR(1);
```

Structure 2.--Data structure variables for Master Water Data Index (MWDI) output file.

```
1 MWDI_OUTPUT,
  2 SITE_DESCRIPTOR_DATA,
    3 SITE_ID FIXED BINARY(31),
    3 AGENCY_CODE CHAR(5),
    3 AGENCY_STATION_NO CHAR(15),
    3 LOCAL_WELL_NO CHAR(24),
    3 STATION_NAME CHAR(48),
    3 LATITUDE FIXED BINARY(31),
    3 LONGITUDE FIXED BINARY(31),
    3 LAT_LONG_ACCURACY CHAR(1),
    3 SITE_TYPE CHAR(2),
    3 COUNTRY_CODE CHAR(2),
    3 RESERVED1 CHAR(1),
    3 STATE_CODE FIXED BINARY(15),
    3 COUNTY_CODE FIXED BINARY(15),
    3 RESERVED2 CHAR(2),
    3 STATE_COUNTY_CODE FIXED BINARY(31),
    3 DISTRICT_CODE CHAR(2),
    3 RESERVED3 CHAR(2),
    3 HYDRO_UNIT_CODE FIXED BINARY(31),
    3 OFFICE_CODE FIXED BINARY(31),
    3 TOTAL_DRAINAGE_AREA FLOAT DEC(6),
    3 CONTRIB_DRAINAGE_AREA FLOAT DEC(6),
    3 ALTITUDE_DATUM FLOAT DEC(6),
    3 ALTITUDE_METH_OF_MEAS CHAR(1),
    3 RESERVED4 CHAR(3),
    3 ALTITUDE_ACCURACY FLOAT DEC(6),
    3 RIVER_REACH_NO FIXED BINARY(15),
    3 RIVER_REACH_SEG_FLAG FIXED BINARY(15),
    3 DEPTH_OF_WELL FLOAT DEC(6),
    3 SOURCE_WELL_DEPTH_DATA CHAR(1),
    3 RESERVED5 CHAR(3),
    3 DEPTH_OF_HOLE FLOAT DEC(6),
    3 PRINCIPAL_AQUIFER_CODE CHAR(8),
    3 AQUIFER_TYPE_CODE CHAR(1),
    3 CONSOLIDATED_AQUIFER_CODE CHAR(1),
    3 BASIN_DESCRIPTOR(3) CHAR(1),
    3 USE_OF_WATER(3) CHAR(1),
    3 USE_OF_SITE(3) CHAR(1),
    3 OTHER_DATA_AVAILABLE(6) CHAR(1),
    3 SW_ACTIVE_CODE CHAR(1),
    3 GW_ACTIVE_CODE CHAR(1),
    3 QW_ACTIVE_CODE CHAR(1),
    3 BIO_ACTIVE_CODE CHAR(1),
    3 PHY_ACTIVE_CODE CHAR(1),
    3 SED_ACTIVE_CODE CHAR(1),
    3 CHM_ACTIVE_CODE CHAR(1),
    3 MET_ACTIVE_CODE CHAR(1),
    3 RESERVED6 CHAR(3),
    3 DATE_OF_LAST_UPDATE FIXED BINARY(31),
  2 NODE_LOCATION_INFO,
```

```
   3 DIS_NODE_IDENT CHAR(6),
   3 SW_DATA_CODE CHAR(1),
   3 GW_DATA_CODE CHAR(1),
   3 QW_DATA_CODE CHAR(1),
   3 MET_DATA_CODE CHAR(1),
   3 OTHER_DATA_CODE CHAR(1),
   3 RESERVED7 CHAR(1),
   3 NODE_DATE_OF_LAST_UPDATE FIXED BINARY(31),
 2 SURFACE_WATER_DATA,
   3 SW_BEGIN_YEAR PIC '999R',
   3 SW_END_YEAR PIC '999R',
   3 SW_INTERRUPTED_CODE CHAR(1),
   3 COMPLETE_STAGE CHAR(1),
   3 PEAK_STAGE CHAR(1),
   3 LOW_STAGE CHAR(1),
   3 STAGE_STORAGE_MEDIA CHAR(1),
   3 COMPLETE_FLOW CHAR(1),
   3 PEAK_FLOW CHAR(1),
   3 LOW_FLOW CHAR(1),
   3 MISC_FLOW_MEAS CHAR(1),
   3 FLOW_STORAGE_MEDIA CHAR(1),
   3 VOLUME CHAR(1),
   3 VOLUME_CHANGE CHAR(1),
   3 VOLUME_STORAGE_MEDIA CHAR(1),
   3 STAGE_TYPE CHAR(2),
   3 STAGE_FREQ_CODE CHAR(1),
   3 FLOW_TYPE CHAR(2),
   3 FLOW_FREQ_CODE CHAR(1),
   3 VOLUME_TYPE CHAR(2),
   3 VOLUME_FREQ_CODE CHAR(1),
   3 VELOCITY_TYPE CHAR(2),
   3 VELOCITY_FREQ_CODE CHAR(1),
   3 BASIN_CHARAC_DATA_AVAIL_CODE CHAR(1),
   3 SW_DATA_STORAGE_MEDIA CHAR(1),
   3 SW_TELEMETRY_CODE CHAR(1),
   3 SW_RECORDER_TYPE_CODE CHAR(1),
   3 SW_DATA_PURPOSE_CODE(4) CHAR(1),
   3 OTHER_SW_RELATED_DATA(12) CHAR(1),
   3 SW_SEASONAL_CODE CHAR(1),
   3 RESERVED8 CHAR(2),
   3 SW_DATE_OF_LAST_UPDATE FIXED BINARY(31),
 2 SW_DATA_BASE_IDENTIFIERS(4),
   3 SW_DATA_BASE_IDENTIFIER CHAR(4),
   3 SW_DATA_BASE_SUBSET_IDENT CHAR(4),
   3 SW_DB_DATE_OF_LAST_UPDATE FIXED BINARY(31),
 2 GROUND_WATER_DATA,
   3 GW_BEGIN_YEAR PIC '999R',
   3 GW_END_YEAR PIC '999R',
   3 GW_INTERRUPTED_CODE CHAR(1),
   3 WATER_LEVEL_FREQ CHAR(1),
   3 WATER_LEVEL_STORAGE_MEDIA CHAR(1),
```

```
   3 DISCHARGE_FREQUENCY CHAR(1),
   3 DISCHARGE_STORAGE_MEDIA CHAR(1),
   3 SUBSIDENCE_FREQ CHAR(1),
   3 SUBSIDENCE_STORAGE_MEDIA CHAR(1),
   3 GW_OTHER_DATA_AVAILABLE CHAR(12),
   3 MAJOR_VARIATIONS_CODE CHAR(4),
   3 GW_TELEMETRY_CODE PIC 'R',
   3 GW_DATE_OF_LAST_UPDATE CHAR(4),
   3 GW_PURPOSE_CODE CHAR(9),
   3 GW_RECORDER_TYPE_CODE CHAR(1),
   3 GW_RECORDER_FREQ_CODE CHAR(1),
 2 GW_MODIFIERS(10),
   3 GW_POINTER CHAR(7),
   3 GW_MOD_FILE CHAR(1),
 2 QUALITY_OF_WATER_DATA,
   3 QW_BEGIN_YEAR PIC '999R',
   3 QW_END_YEAR PIC '999R',
   3 QW_INTERRUPTED_CODE CHAR(1),
   3 QW_TELEMETRY_CODE CHAR(1),
   3 QW_DATE_OF_LAST_UPDATE CHAR(4),
   3 QW_PURPOSE_CODE CHAR(9),
   3 QW_RECORDER_TYPE_CODE CHAR(1),
   3 QW_RECORDER_FREQ_CODE CHAR(1),
   3 STORET_POINTER CHAR(7),
 2 QW_MODIFIERS(10),
   3 QW_POINTER CHAR(7),
   3 QW_MOD_FILE CHAR(1),
 2 BIOLOGICAL_DATA,
   3 ENTERIC_BACTERIA CHAR(1),
   3 NATIVE_BACTERIA CHAR(1),
   3 PHYTOPLANKTON CHAR(1),
   3 ZOOPLANKTON CHAR(1),
   3 PERIPHYTON CHAR(1),
   3 MACROPHYTON CHAR(1),
   3 MICROINVERTEBRATES CHAR(1),
   3 MACROINVERTEBRATES CHAR(1),
   3 VERTEBRATES CHAR(1),
   3 FUNGI CHAR(1),
   3 VIRUSES CHAR(1),
   3 BIO_BEGIN_YEAR CHAR(4),
   3 BIO_END_YEAR CHAR(4),
   3 BIO_DATE_OF_LAST_UPDATE CHAR(4),
   3 BIO_STORAGE_MEDIA CHAR(1),
   3 PRIMARY_PRODUCTIVITY CHAR(1),
   3 SECONDARY_PRODUCTIVITY CHAR(1),
   3 CHEMOSYNTHETIC_ACTIVITY CHAR(1),
   3 BIOSTIMULATORY_TEST CHAR(1),
   3 TOXICITY_TEST CHAR(1),
   3 OTHER_BIO_ASSAY_TEST CHAR(1),
   3 CHEMICAL_TISSUE_ANALYSIS CHAR(1),
   3 HISTOPATHIC_ANALYSIS CHAR(1),
```

```
  3 OTHER_TISSUE_ANALYSES CHAR(1),
2 BIO_MODIFIERS(10),
  3 BIO_POINTER CHAR(7),
  3 BIO_MOD_FILE CHAR(1),
2 QW_PHYSICAL_DATA,
  3 TEMPERATURE CHAR(1),
  3 SPECIFIC_CONDUCTANCE CHAR(1),
  3 TURBIDITY CHAR(1),
  3 COLOR CHAR(1),
  3 ODOR CHAR(1),
  3 PH CHAR(1),
  3 SUSPENDED_SOLIDS CHAR(1),
  3 PHY_BEGIN_YEAR CHAR(4),
  3 PHY_END_YEAR CHAR(4),
  3 PHY_DATE_OF_LAST_UPDATE CHAR(4),
  3 PHY_STORAGE_MEDIA CHAR(1),
2 PHY_MODIFIERS(10),
  3 PHY_POINTER CHAR(7),
  3 PHY_MOD_FILE CHAR(1),
2 QW_SEDIMENT_DATA,
  3 BED_LOAD CHAR(1),
  3 SUSPENDED_CONC CHAR(1),
  3 TOTAL_CONC CHAR(1),
  3 SUSPENDED_PARTICLE_SIZE CHAR(1),
  3 BED_PARTICLE_SIZE CHAR(1),
  3 SUSPENDED_SEDIMENT_DISCH CHAR(1),
  3 TOTAL_SEDIMENT_DISCH CHAR(1),
  3 SED_BEGIN_YEAR CHAR(4),
  3 SED_END_YEAR CHAR(4),
  3 SED_DATE_OF_LAST_UPDATE CHAR(4),
  3 SED_STORAGE_MEDIA CHAR(1),
2 SED_MODIFIERS(10),
  3 SED_POINTER CHAR(7),
  3 SED_MOD_FILE CHAR(1),
2 QW_CHEMICAL_DATA,
  3 DISSOLVED_SOLIDS CHAR(1),
  3 MAJOR_IONS CHAR(1),
  3 HARDNESS CHAR(1),
  3 SILICA CHAR(1),
  3 PHOSPHORUS CHAR(1),
  3 PHOSPHORUS_SPECIES CHAR(1),
  3 NITROGEN CHAR(1),
  3 NITROGEN_SPECIES CHAR(1),
  3 DETERGENTS CHAR(1),
  3 OTHER_MINOR_CONSTITUENTS CHAR(1),
  3 RADIOACTIVITY CHAR(1),
  3 RADIOCHEMICAL_SPECIES CHAR(1),
  3 CARBON CHAR(1),
  3 ORGANIC_GROUPS CHAR(1),
  3 PESTICIDE_SPECIES CHAR(1),
  3 OTHER_ORGANIC_SPECIES CHAR(1),
```

```
   3 BIOCHEMICAL_OXYGEN_DEMAND CHAR(1),
   3 CHEMICAL_OXYGEN_DEMAND CHAR(1),
   3 DISSOLVED_OXYGEN CHAR(1),
   3 OTHER_DISSOLVED_GASES CHAR(1),
   3 CHM_BEGIN_YEAR CHAR(4),
   3 CHM_END_YEAR CHAR(4),
   3 CHM_DATE_OF_LAST_UPDATE CHAR(4),
   3 CHM_STORAGE_MEDIA CHAR(1),
 2 CHM_MODIFIERS(10),
   3 CHM_POINTER CHAR(7),
   3 CHM_MOD_FILE CHAR(1),
 2 PROJECTS(10),
   3 WRD_PROJECT_NUMBER CHAR(5),
 2 NETWORKS(10),
   3 NETWORK_CODE CHAR(4),
 2 OTHER_SOURCE(10),
   3 OTHER_SOURCE_AGENCY CHAR(5),
 2 SOURCE_INFORMATION(10),
   3 SOURCE_FILE_IDENTIFIER CHAR(4),
   3 SOURCE_FILE_AGENCY CHAR(8),
 2 METEOROLOGICAL_DATA,
   3 MET_BEGIN_DATE CHAR(4),
   3 MET_END_DATE CHAR(4),
   3 MET_INTERRUPTED_CODE CHAR(1),
   3 RAINFALL CHAR(1),
   3 UNIT_RAINFALL CHAR(1),
   3 AIR_TEMPERATURE CHAR(1),
   3 WIND_VELOCITY CHAR(1),
   3 MET_OTHER_DATA_AVAILABLE CHAR(12),
   3 MET_TELEMETRY_CODE CHAR(1),
   3 MET_DATE_OF_LAST_UPDATE CHAR(4),
   3 MET_STORAGE_MEDIA CHAR(1),
   3 MET_RECORDER_TYPE_CODE CHAR(1),
   3 MET_RECORDER_FREQ_CODE CHAR(1),
 2 MET_MODIFIERS(10),
   3 MET_POINTER CHAR(7),
   3 MET_MOD_FILE CHAR(1);
```

Structure 3.--Data structure variables for Master Water Data Index (MWDI) output template.

```
1 MWDI_TEMPLATE,
  2 SITE_DESCRIPTOR_DATA,
    3 SITE_ID FIXED BINARY(31),
    3 AGENCY_CODE CHAR(5),
    3 AGENCY_STATION_NO CHAR(15),
    3 LOCAL_WELL_NO CHAR(24),
    3 STATION_NAME CHAR(48),
    3 LATITUDE FIXED BINARY(31),
    3 LONGITUDE FIXED BINARY(31),
    3 LAT_LONG_ACCURACY CHAR(1),
    3 SITE_TYPE CHAR(2),
    3 COUNTRY_CODE CHAR(2),
    3 RESERVED1 CHAR(1),
    3 STATE_CODE FIXED BINARY(15),
    3 COUNTY_CODE FIXED BINARY(15),
    3 RESERVED2 CHAR(2),
    3 STATE_COUNTY_CODE FIXED BINARY(31),
    3 DISTRICT_CODE CHAR(2),
    3 RESERVED3 CHAR(2),
    3 HYDRO_UNIT_CODE FIXED BINARY(31),
    3 OFFICE_CODE FIXED BINARY(31),
    3 TOTAL_DRAINAGE_AREA FLOAT DEC(6),
    3 CONTRIB_DRAINAGE_AREA FLOAT DEC(6),
    3 ALTITUDE_DATUM FLOAT DEC(6),
    3 ALTITUDE_METH_OF_MEAS CHAR(1),
    3 RESERVED4 CHAR(3),
    3 ALTITUDE_ACCURACY FLOAT DEC(6),
    3 RIVER_REACH_NO FIXED BINARY(15),
    3 RIVER_REACH_SEG_FLAG FIXED BINARY(15),
    3 DEPTH_OF_WELL FLOAT DEC(6),
    3 SOURCE_WELL_DEPTH_DATA CHAR(1),
    3 RESERVED5 CHAR(3),
    3 DEPTH_OF_HOLE FLOAT DEC(6),
    3 PRINCIPAL_AQUIFER_CODE CHAR(8),
    3 AQUIFER_TYPE_CODE CHAR(1),
    3 CONSOLIDATED_AQUIFER_CODE CHAR(1),
    3 BASIN_DESCRIPTOR(3) CHAR(1),
    3 USE_OF_WATER(3) CHAR(1),
    3 USE_OF_SITE(3) CHAR(1),
    3 OTHER_DATA_AVAILABLE(6) CHAR(1),
    3 SW_ACTIVE_CODE CHAR(1),
    3 GW_ACTIVE_CODE CHAR(1),
    3 QW_ACTIVE_CODE CHAR(1),
    3 BIO_ACTIVE_CODE CHAR(1),
    3 PHY_ACTIVE_CODE CHAR(1),
    3 SED_ACTIVE_CODE CHAR(1),
    3 CHM_ACTIVE_CODE CHAR(1),
    3 MET_ACTIVE_CODE CHAR(1),
    3 RESERVED6 CHAR(3),
    3 DATE_OF_LAST_UPDATE FIXED BINARY(31),
  2 NODE_LOCATION_INFO,
```

```
   3 DIS_NODE_IDENT CHAR(6),
   3 SW_DATA_CODE CHAR(1),
   3 GW_DATA_CODE CHAR(1),
   3 QW_DATA_CODE CHAR(1),
   3 MET_DATA_CODE CHAR(1),
   3 OTHER_DATA_CODE CHAR(1),
   3 RESERVED7 CHAR(1),
   3 NODE_DATE_OF_LAST_UPDATE FIXED BINARY(31),
 2 SURFACE_WATER_DATA,
   3 SW_BEGIN_YEAR PIC '999R',
   3 SW_END_YEAR PIC '999R',
   3 SW_INTERRUPTED_CODE CHAR(1),
   3 COMPLETE_STAGE CHAR(1),
   3 PEAK_STAGE CHAR(1),
   3 LOW_STAGE CHAR(1),
   3 STAGE_STORAGE_MEDIA CHAR(1),
   3 COMPLETE_FLOW CHAR(1),
   3 PEAK_FLOW CHAR(1),
   3 LOW_FLOW CHAR(1),
   3 MISC_FLOW_MEAS CHAR(1),
   3 FLOW_STORAGE_MEDIA CHAR(1),
   3 VOLUME CHAR(1),
   3 VOLUME_CHANGE CHAR(1),
   3 VOLUME_STORAGE_MEDIA CHAR(1),
   3 STAGE_TYPE CHAR(2),
   3 STAGE_FREQ_CODE CHAR(1),
   3 FLOW_TYPE CHAR(2),
   3 FLOW_FREQ_CODE CHAR(1),
   3 VOLUME_TYPE CHAR(2),
   3 VOLUME_FREQ_CODE CHAR(1),
   3 VELOCITY_TYPE CHAR(2),
   3 VELOCITY_FREQ_CODE CHAR(1),
   3 BASIN_CHARAC_DATA_AVAIL_CODE CHAR(1),
   3 SW_DATA_STORAGE_MEDIA CHAR(1),
   3 SW_TELEMETRY_CODE CHAR(1),
   3 SW_RECORDER_TYPE_CODE CHAR(1),
   3 SW_DATA_PURPOSE_CODE(4) CHAR(1),
   3 OTHER_SW_RELATED_DATA(12) CHAR(1),
   3 SW_SEASONAL_CODE CHAR(1),
   3 RESERVED8 CHAR(2),
   3 SW_DATE_OF_LAST_UPDATE FIXED BINARY(31),
 2 SW_DATA_BASE_IDENTIFIERS(4),
   3 SW_DATA_BASE_IDENTIFIER CHAR(4),
   3 SW_DATA_BASE_SUBSET_IDENT CHAR(4),
   3 SW_DB_DATE_OF_LAST_UPDATE FIXED BINARY(31),
 2 GROUND_WATER_DATA,
   3 GW_BEGIN_YEAR PIC '999R',
   3 GW_END_YEAR PIC '999R',
   3 GW_INTERRUPTED_CODE CHAR(1),
   3 WATER_LEVEL_FREQ CHAR(1),
   3 WATER_LEVEL_STORAGE_MEDIA CHAR(1),
```

```
   3 DISCHARGE_FREQUENCY CHAR(1),
   3 DISCHARGE_STORAGE_MEDIA CHAR(1),
   3 SUBSIDENCE_FREQ CHAR(1),
   3 SUBSIDENCE_STORAGE_MEDIA CHAR(1),
   3 GW_OTHER_DATA_AVAILABLE CHAR(12),
   3 MAJOR_VARIATIONS_CODE CHAR(4),
   3 GW_TELEMETRY_CODE PIC 'R',
   3 GW_DATE_OF_LAST_UPDATE CHAR(4),
   3 GW_PURPOSE_CODE CHAR(9),
   3 GW_RECORDER_TYPE_CODE CHAR(1),
   3 GW_RECORDER_FREQ_CODE CHAR(1),
 2 GW_MODIFIERS(10),
   3 GW_POINTER CHAR(7),
   3 GW_MOD_FILE CHAR(1),
 2 QUALITY_OF_WATER_DATA,
   3 QW_BEGIN_YEAR PIC '999R',
   3 QW_END_YEAR PIC '999R',
   3 QW_INTERRUPTED_CODE CHAR(1),
   3 QW_TELEMETRY_CODE CHAR(1),
   3 QW_DATE_OF_LAST_UPDATE CHAR(4),
   3 QW_PURPOSE_CODE CHAR(9),
   3 QW_RECORDER_TYPE_CODE CHAR(1),
   3 QW_RECORDER_FREQ_CODE CHAR(1),
   3 STORET_POINTER CHAR(7),
 2 QW_MODIFIERS(10),
   3 QW_POINTER CHAR(7),
   3 QW_MOD_FILE CHAR(1),
 2 BIOLOGICAL_DATA,
   3 ENTERIC_BACTERIA CHAR(1),
   3 NATIVE_BACTERIA CHAR(1),
   3 PHYTOPLANKTON CHAR(1),
   3 ZOOPLANKTON CHAR(1),
   3 PERIPHYTON CHAR(1),
   3 MACROPHYTON CHAR(1),
   3 MICROINVERTEBRATES CHAR(1),
   3 MACROINVERTEBRATES CHAR(1),
   3 VERTEBRATES CHAR(1),
   3 FUNGI CHAR(1),
   3 VIRUSES CHAR(1),
   3 BIO_BEGIN_YEAR CHAR(4),
   3 BIO_END_YEAR CHAR(4),
   3 BIO_DATE_OF_LAST_UPDATE CHAR(4),
   3 BIO_STORAGE_MEDIA CHAR(1),
   3 PRIMARY_PRODUCTIVITY CHAR(1),
   3 SECONDARY_PRODUCTIVITY CHAR(1),
   3 CHEMOSYNTHETIC_ACTIVITY CHAR(1),
   3 BIOSTIMULATORY_TEST CHAR(1),
   3 TOXICITY_TEST CHAR(1),
   3 OTHER_BIO_ASSAY_TEST CHAR(1),
   3 CHEMICAL_TISSUE_ANALYSIS CHAR(1),
   3 HISTOPATHIC_ANALYSIS CHAR(1),
```

```
  3 OTHER_TISSUE_ANALYSES CHAR(1),
2 BIO_MODIFIERS(10),
  3 BIO_POINTER CHAR(7),
  3 BIO_MOD_FILE CHAR(1),
2 QW_PHYSICAL_DATA,
  3 TEMPERATURE CHAR(1),
  3 SPECIFIC_CONDUCTANCE CHAR(1),
  3 TURBIDITY CHAR(1),
  3 COLOR CHAR(1),
  3 ODOR CHAR(1),
  3 PH CHAR(1),
  3 SUSPENDED_SOLIDS CHAR(1),
  3 PHY_BEGIN_YEAR CHAR(4),
  3 PHY_END_YEAR CHAR(4),
  3 PHY_DATE_OF_LAST_UPDATE CHAR(4),
  3 PHY_STORAGE_MEDIA CHAR(1),
2 PHY_MODIFIERS(10),
  3 PHY_POINTER CHAR(7),
  3 PHY_MOD_FILE CHAR(1),
2 QW_SEDIMENT_DATA,
  3 BED_LOAD CHAR(1),
  3 SUSPENDED_CONC CHAR(1),
  3 TOTAL_CONC CHAR(1),
  3 SUSPENDED_PARTICLE_SIZE CHAR(1),
  3 BED_PARTICLE_SIZE CHAR(1),
  3 SUSPENDED_SEDIMENT_DISCH CHAR(1),
  3 TOTAL_SEDIMENT_DISCH CHAR(1),
  3 SED_BEGIN_YEAR CHAR(4),
  3 SED_END_YEAR CHAR(4),
  3 SED_DATE_OF_LAST_UPDATE CHAR(4),
  3 SED_STORAGE_MEDIA CHAR(1),
2 SED_MODIFIERS(10),
  3 SED_POINTER CHAR(7),
  3 SED_MOD_FILE CHAR(1),
2 QW_CHEMICAL_DATA,
  3 DISSOLVED_SOLIDS CHAR(1),
  3 MAJOR_IONS CHAR(1),
  3 HARDNESS CHAR(1),
  3 SILICA CHAR(1),
  3 PHOSPHORUS CHAR(1),
  3 PHOSPHORUS_SPECIES CHAR(1),
  3 NITROGEN CHAR(1),
  3 NITROGEN_SPECIES CHAR(1),
  3 DETERGENTS CHAR(1),
  3 OTHER_MINOR_CONSTITUENTS CHAR(1),
  3 RADIOACTIVITY CHAR(1),
  3 RADIOCHEMICAL_SPECIES CHAR(1),
  3 CARBON CHAR(1),
  3 ORGANIC_GROUPS CHAR(1),
  3 PESTICIDE_SPECIES CHAR(1),
  3 OTHER_ORGANIC_SPECIES CHAR(1),
```

```
    3 BIOCHEMICAL_OXYGEN_DEMAND CHAR(1),
    3 CHEMICAL_OXYGEN_DEMAND CHAR(1),
    3 DISSOLVED_OXYGEN CHAR(1),
    3 OTHER_DISSOLVED_GASES CHAR(1),
    3 CHM_BEGIN_YEAR CHAR(4),
    3 CHM_END_YEAR CHAR(4),
    3 CHM_DATE_OF_LAST_UPDATE CHAR(4),
    3 CHM_STORAGE_MEDIA CHAR(1),
  2 CHM_MODIFIERS(10),
    3 CHM_POINTER CHAR(7),
    3 CHM_MOD_FILE CHAR(1),
  2 PROJECTS(10),
    3 WRD_PROJECT_NUMBER CHAR(5),
  2 NETWORKS(10),
    3 NETWORK_CODE CHAR(4),
  2 OTHER_SOURCE(10),
    3 OTHER_SOURCE_AGENCY CHAR(5),
  2 SOURCE_INFORMATION(10),
    3 SOURCE_FILE_IDENTIFIER CHAR(4),
    3 SOURCE_FILE_AGENCY CHAR(8),
  2 METEOROLOGICAL_DATA,
    3 MET_BEGIN_DATE CHAR(4),
    3 MET_END_DATE CHAR(4),
    3 MET_INTERRUPTED_CODE CHAR(1),
    3 RAINFALL CHAR(1),
    3 UNIT_RAINFALL CHAR(1),
    3 AIR_TEMPERATURE CHAR(1),
    3 WIND_VELOCITY CHAR(1),
    3 MET_OTHER_DATA_AVAILABLE CHAR(12),
    3 MET_TELEMETRY_CODE CHAR(1),
    3 MET_DATE_OF_LAST_UPDATE CHAR(4),
    3 MET_STORAGE_MEDIA CHAR(1),
    3 MET_RECORDER_TYPE_CODE CHAR(1),
    3 MET_RECORDER_FREQ_CODE CHAR(1),
  2 MET_MODIFIERS(10),
    3 MET_POINTER CHAR(7),
    3 MET_MOD_FILE CHAR(1);
```

Structure 4.--Data structure variables for National Water Data Storage and Retrieval System (WATSTORE) output file.

```
1 WATSTORE_UV_RECORD,
  2 RESERVED_SPACE CHAR(1),
  2 RECORD_FORMAT CHAR(1),
  2 STATE_CODE CHAR(2),
  2 AGENCY_CODE CHAR(5),
  2 STATION_IDENTIFIER CHAR(15),
  2 CROSS_SECTION_LOCATOR FLOAT DEC(6),
  2 DEPTH_LOCATOR FLOAT DEC(6),
  2 PARAMETER_CODE FIXED BINARY(31),
  2 YEAR_NUMBER FIXED BINARY(15),
  2 MONTH_NUMBER FIXED BINARY(15),
  2 DAY_NUMBER FIXED BINARY(15),
  2 STATISTIC_CODE FIXED BINARY(15),
  2 READINGS_PER_DAY FIXED BINARY(15),
  2 FIRST_READING_POSITION FIXED BINARY(15),
  2 NUMBER_OF_READINGS FIXED BINARY(15),
  2 RETRIEVAL_SEQUENCE_NUMBER FIXED BINARY(15),
  2 MORE_RESERVED_SPACE CHAR(9),
  2 SITE_CODE CHAR(2),
  2 DISTRICT_CODE CHAR(2),
  2 COUNTY_CODE CHAR(3),
  2 CREATE_DATE FIXED BINARY(31),
  2 EVEN_MORE_RESERVED_SPACE CHAR(4),
  2 MISSING_VALUE_INDICATOR FLOAT DEC(6),
  2 UNIT_VALUES(2880) FLOAT DEC(6);
```

Structure 5.--Data structure variables for unique site identification or identifier file.

```
1 UNIQUE_SITE_IDENTIFIER,
  2 DEL CHAR(1),
  2 STATE_CODE CHAR(2),
  2 RESERVED CHAR(1),
  2 NEXT_AVAIL_UNIQUE_SITE_ID FIXED BINARY(31);
```

Structure 6.--Data structure variables for surface-water frequency output file.

```
1 SW_FREQ_HISTORY_OUTPUT,
  2 UNIQUE_SITE_ID FIXED BINARY(31),
  2 AGENCY_CODE CHAR(5),
  2 AGENCY_STATION_NUMBER CHAR(15),
  2 SW_YEAR_OF_MEASUREMENT FIXED BIN(31),
  2 STAGE_TYPE_HISTORY CHAR(2),
  2 STAGE_FREQ_HISTORY CHAR(1),
  2 FLOW_TYPE_HISTORY CHAR(2),
  2 FLOW_FREQ_HISTORY CHAR(1),
  2 VOLUME_TYPE_HISTORY CHAR(2),
  2 VOLUME_FREQ_HISTORY CHAR(1),
  2 VELOCITY_TYPE_HISTORY CHAR(2),
  2 VELOCITY_FREQ_HISTORY CHAR(1),
  2 SW_HISTORY_DATE_OF_LAST_UPDATE FIXED BINARY(31);
```

Structure 7.--Data structure variables for other data transactions output file.

```
1 GW_QW_MET_TRANSACTION_OUTPUT,
  2 UNIQUE_SITE_ID FIXED BINARY(31),
  2 AGENCY_CODE CHAR(5),
  2 AGENCY_STATION_NUMBER CHAR(15),
  2 PARAMETER_CODE FIXED BINARY(31),
  2 YEAR_OF_MEASUREMENT FIXED BINARY(15),
  2 FREQUENCY_CODE CHAR(1),
  2 SEASONAL_CODE CHAR(1),
  2 DATA_BASE_IDENTIFIER CHAR(4),
  2 DATA_BASE_SUBSET_IDENTIFIER CHAR(4);
```

Structure 8.--Data structure for element-counter variables on a per state basis.

```
1 PER_STATE_DATA_ELEMENT_COUNTS,
  2 INPUT,
    3 UNIQUE_SITE_IDENTIFIER FIXED DEC(7),
    3 AGENCY_STATION_NUMBER FIXED DEC(7),
    3 SW_ACTIVE_CODE FIXED DEC(7),
    3 SW_BEGIN_YEAR FIXED DEC(7),
    3 SW_END_YEAR FIXED DEC(7),
    3 SW_INTERRUPTED_CODE FIXED DEC(7),
    3 UNIT_FLOWS FIXED DEC(7),
    3 UNIT_STAGE FIXED DEC(7),
    3 UNIT_VOLUME FIXED DEC(7),
    3 SW_OTHER_DATA_AVAILABLE FIXED DEC(7),
    3 SW_TELEMETRY_CODE FIXED DEC(7),
    3 SW_PURPOSE_CODE FIXED DEC(7),
    3 SW_RECORDER_TYPE_CODE FIXED DEC(7),
  2 OUTPUT,
    3 UNIQUE_SITE_IDENTIFIER FIXED DEC(7),
    3 AGENCY_STATION_NUMBER FIXED DEC(7),
    3 SW_ACTIVE_CODE FIXED DEC(7),
    3 SW_BEGIN_YEAR FIXED DEC(7),
    3 SW_END_YEAR FIXED DEC(7),
    3 SW_INTERRUPTED_CODE FIXED DEC(7),
    3 OTHER_SW_RELATED_DATA FIXED DEC(7),
    3 SW_TELEMETRY_CODE FIXED DEC(7),
    3 SW_DATA_PURPOSE_CODE(4) FIXED DEC(7),
    3 SW_RECORDER_TYPE_CODE FIXED DEC(7),
    3 STAGE_TYPE FIXED DEC(7),
    3 STAGE_FREQ_CODE FIXED DEC(7),
    3 FLOW_TYPE FIXED DEC(7),
    3 FLOW_FREQ_CODE  FIXED DEC(7),
    3 VOLUME_TYPE FIXED DEC(7),
    3 VOLUME_FREQ_CODE FIXED DEC(7),
    3 VELOCITY_TYPE FIXED DEC(7),
    3 VELOCITY_FREQ_CODE FIXED DEC(7),
    3 SW_DATA_STORAGE_MEDIA FIXED DEC(7),
    3 SW_SEASONAL_CODE FIXED DEC(7),
    3 SW_DATE_OF_LAST_UPDATE FIXED DEC(7),
    3 SW_DATA_CODE FIXED DEC(7),
    3 SW_DATA_BASE_IDENTIFIER(4) FIXED DEC(7),
    3 SW_DATA_BASE_SUBSET_IDENTIFIER(4) FIXED DEC(7);
```

Structure 9.--Data structure for element-counter variables on an overall basis.

```
1 DATA_ELEMENT_COUNTS,
  2 INPUT,
    3 UNIQUE_SITE_IDENTIFIER FIXED DEC(7),
    3 AGENCY_STATION_NUMBER FIXED DEC(7),
    3 SW_ACTIVE_CODE FIXED DEC(7),
    3 SW_BEGIN_YEAR FIXED DEC(7),
    3 SW_END_YEAR FIXED DEC(7),
    3 SW_INTERRUPTED_CODE FIXED DEC(7),
    3 UNIT_FLOWS FIXED DEC(7),
    3 UNIT_STAGE FIXED DEC(7),
    3 UNIT_VOLUME FIXED DEC(7),
    3 SW_OTHER_DATA_AVAILABLE FIXED DEC(7),
    3 SW_TELEMETRY_CODE FIXED DEC(7),
    3 SW_PURPOSE_CODE FIXED DEC(7),
    3 SW_RECORDER_TYPE_CODE FIXED DEC(7),
  2 OUTPUT,
    3 UNIQUE_SITE_IDENTIFIER FIXED DEC(7),
    3 AGENCY_STATION_NUMBER FIXED DEC(7),
    3 SW_ACTIVE_CODE FIXED DEC(7),
    3 SW_BEGIN_YEAR FIXED DEC(7),
    3 SW_END_YEAR FIXED DEC(7),
    3 SW_INTERRUPTED_CODE FIXED DEC(7),
    3 OTHER_SW_RELATED_DATA FIXED DEC(7),
    3 SW_TELEMETRY_CODE FIXED DEC(7),
    3 SW_DATA_PURPOSE_CODE(4) FIXED DEC(7),
    3 SW_RECORDER_TYPE_CODE FIXED DEC(7),
    3 STAGE_TYPE FIXED DEC(7),
    3 STAGE_FREQ_CODE FIXED DEC(7),
    3 FLOW_TYPE FIXED DEC(7),
    3 FLOW_FREQ_CODE  FIXED DEC(7),
    3 VOLUME_TYPE FIXED DEC(7),
    3 VOLUME_FREQ_CODE FIXED DEC(7),
    3 VELOCITY_TYPE FIXED DEC(7),
    3 VELOCITY_FREQ_CODE FIXED DEC(7),
    3 SW_DATA_STORAGE_MEDIA FIXED DEC(7),
    3 SW_SEASONAL_CODE FIXED DEC(7),
    3 SW_DATE_OF_LAST_UPDATE FIXED DEC(7),
    3 SW_DATA_CODE FIXED DEC(7),
    3 SW_DATA_BASE_IDENTIFIER(4) FIXED DEC(7),
    3 SW_DATA_BASE_SUBSET_IDENTIFIER(4) FIXED DEC(7);
```

Other variables used in the program.

```
MWDI_NOT_EOF                          BIT(1)
WATSTORE_NOT_EOF                      BIT(1)
FIRST_TIME_THROUGH                    BIT(1)
NOT_COUNTED                           BIT(1)
ITS_SW_DATA                           BIT(1)
ITS_SW_DATA2                          BIT(1)
NEW_RECORD                            BIT(1)
OTHER_NEW_RECORD                      BIT(1)
LAST_PASS                             BIT(1)
OTHER_RECORD_WRITTEN                  BIT(1)
FIRST_SW_VAL                          BIT(1)
CURRENT_STATE FIXED                   BINARY(15)
PREVIOUS_STATE FIXED                  BINARY(15)
PREVIOUS_STATION_ID                   CHAR(15)
VALIDATE_ID                           CHAR(15)
CONDITION_FLAG                        FIXED DEC(1)
PER_STATE_TOTAL_SITES_IN              FIXED DEC(7)
PER_STATE_TOTAL_SITES_OUT             FIXED DEC(7)
PER_STATE_TOTAL_SITES_ADDED           FIXED DEC(7)
TOTAL_SITES_IN                        FIXED DEC(7)
TOTAL_SITES_OUT                       FIXED DEC(7)
TOTAL_SITES_ADDED                     FIXED DEC(7)
PREV_ID                               CHAR(15)
SW_DAY                                FIXED DEC
OTHER_DAY                             FIXED DEC
CURRENT_PARAMETER_CODE                FIXED DEC
NUMSTATES                             FIXED DEC
SW_YEAR                               FIXED BIN(15)
OTHER_YEAR                            FIXED BIN(15)
NUM_DAYS_THIS_PARM                    FIXED BIN(15)
SW_VALIDATE_PARM                      FIXED BIN(31)
SW_VALIDATE_YEAR                      FIXED BIN(15)
PER_STATE_SITES_WITH_FREQ_HIST        FIXED DEC(7)
PER_STATE_NUM_FREQ_HIST_RECORDS       FIXED DEC(7)
SITES_WITH_FREQ_HIST                  FIXED DEC(7)
NUM_FREQ_HIST_RECORDS                 FIXED DEC(7)
PER_STATE_GW_TRANSACTIONS             FIXED DEC(7)
PER_STATE_QW_TRANSACTIONS             FIXED DEC(7)
PER_STATE_MET_TRANSACTIONS            FIXED DEC(7)
PER_STATE_TOTAL_TRANSACTIONS          FIXED DEC(7)
GW_TRANSACTIONS                       FIXED DEC(7)
QW_TRANSACTIONS                       FIXED DEC(7)
MET_TRANSACTIONS                      FIXED DEC(7)
TOTAL_TRANSACTIONS                    FIXED DEC(7)
```

APPENDIX D. EXAMPLE OUTPUT LISTINGS

Example 1.--Data-element counts for a state.

STATE: 42

TOTAL SITES IN: 5

TOTAL SITES OUT: 7

TOTAL SITES ADDED: 2

DATA ELEMENT COUNTS:

STATE: 42

DATA ELEMENT	TOTAL IN	TOTAL OUT
UNIQUE SITE IDENTIFIER	5	7
AGENCY STATION NUMBER	4	6
SW ACTIVE CODE	4	6
SW BEGIN YEAR	4	6
SW END YEAR	0	0
SW INTERRUPTED CODE	1	4
UNIT FLOWS	0	
UNIT STAGE	0	
UNIT VOLUMES	0	
SW OTHER DATA AVAILABLE	1	1
SW TELEMETRY CODE	0	0
SW PURPOSE CODE	2	
SW RECORDER TYPE CODE	0	4
STAGE TYPE		4
STAGE FREQUENCY CODE		4
FLOW TYPE		0
FLOW FREQUENCY CODE		0

VOLUME TYPE	0
VOLUME FREQUENCY CODE	0
VELOCITY TYPE	0
VELOCITY FREQUENCY CODE	0
SW DATA STORAGE MEDIA	4
SW SEASONAL CODE	4
SW DATE OF LAST UPDATE	6
SW DATA CODE	4
SW DATA PURPOSE CODE(1)	2
SW DATA PURPOSE CODE(2)	0
SW DATA PURPOSE CODE(3)	0
SW DATA PURPOSE CODE(4)	0
SW DATA BASE IDENTIFIER(1)	4
SW DATA BASE SUBSET IDENTIFIER(1)	4
SW DATA BASE IDENTIFIER(2)	1
SW DATA BASE SUBSET IDENTIFIER(2)	1
SW DATA BASE IDENTIFIER(3)	1
SW DATA BASE SUBSET IDENTIFIER(3)	1
SW DATA BASE IDENTIFIER(4)	0
SW DATA BASE SUBSET IDENTIFIER(4)	0

STATE: 42

NUMBER OF SITES WITH FREQUENCY HISTORIES: 4

NUMBER OF FREQUENCY HISTORY RECORDS: 12

STATE: 42

NUMBER OF GROUND-WATER TRANSACTIONS: 0

NUMBER OF QUALITY-OF-WATER TRANSACTIONS: 4

NUMBER OF METEOROLOGICAL TRANSACTIONS: 0

TOTAL NUMBER OF TRANSACTIONS: 4

Example 2.--Data-element counts for a complete run.

PROCESSING COMPLETED

TOTAL SITES IN: 9

TOTAL SITES OUT: 11

TOTAL SITES ADDED: 2

PROCESSING COMPLETED

TOTAL DATA ELEMENT COUNTS:

DATA ELEMENT	TOTAL IN	TOTAL OUT
UNIQUE SITE IDENTIFIER	9	11
AGENCY STATION NUMBER	7	9
SW ACTIVE CODE	6	8
SW BEGIN YEAR	6	8
SW END YEAR	0	0
SW INTERRUPTED CODE	2	5
UNIT FLOWS	1	
UNIT STAGE	1	
UNIT VOLUMES	0	
SW OTHER DATA AVAILABLE	3	3
SW TELEMETRY CODE	0	0
SW PURPOSE CODE	3	
SW RECORDER TYPE CODE	0	4
STAGE TYPE		5
STAGE FREQUENCY CODE		5
FLOW TYPE		1

FLOW FREQUENCY CODE	1
VOLUME TYPE	0
VOLUME FREQUENCY CODE	0
VELOCITY TYPE	0
VELOCITY FREQUENCY CODE	0
SW DATA STORAGE MEDIA	4
SW SEASONAL CODE	4
SW DATE OF LAST UPDATE	8
SW DATA CODE	4
SW DATA PURPOSE CODE(1)	3
SW DATA PURPOSE CODE(2)	0
SW DATA PURPOSE CODE(3)	0
SW DATA PURPOSE CODE(4)	0
SW DATA BASE IDENTIFIER(1)	5
SW DATA BASE SUBSET IDENTIFIER(1)	5
SW DATA BASE IDENTIFIER(2)	2
SW DATA BASE SUBSET IDENTIFIER(2)	2
SW DATA BASE IDENTIFIER(3)	1
SW DATA BASE SUBSET IDENTIFIER(3)	1
SW DATA BASE IDENTIFIER(4)	0
SW DATA BASE SUBSET IDENTIFIER(4)	0

PROCESSING COMPLETED

TOTAL NUMBER OF SITES WITH FREQUENCY HISTORIES: 4

TOTAL NUMBER OF FREQUENCY HISTORY RECORDS: 12

PROCESSING COMPLETED

TOTAL NUMBER OF GROUND-WATER TRANSACTIONS: 0

TOTAL NUMBER OF QUALITY-OF-WATER TRANSACTIONS: 4

TOTAL NUMBER OF METEOROLOGICAL TRANSACTIONS: 7

TOTAL NUMBER OF TRANSACTIONS: 11

Example 3.--Validation listing for Master Water Data Index (MWDI) output.

UNIQUE SITE IDENTIFIER: 420000004

AGENCY CODE: USGS

AGENCY STATION NUMBER: 01547200

SITE TYPE: SW

STATE CODE: 42

COUNTY CODE: 27

STATE-COUNTY CODE: 42027

DISTRICT CODE:

OTHER DATA AVAILABLE: 6

SW ACTIVE CODE: Y

SW DATA CODE: Y

SW BEGIN YEAR: 1954

SW END YEAR: 0000

SW INTERRUPTED CODE: N

STAGE TYPE: C

STAGE FREQUENCY CODE: C

FLOW TYPE:

FLOW FREQUENCY CODE:

VOLUME TYPE:

VOLUME FREQUENCY CODE:

VELOCITY TYPE:

VELOCITY FREQUENCY CODE:

SW DATA STORAGE MEDIA: C

SW TELEMETRY CODE:

SW RECORDER TYPE CODE: A

SW DATA PURPOSE CODE: A

OTHER SURFACE-WATER RELATED DATA:

SW SEASONAL CODE: S

SW DATE OF LAST UPDATE: 19870109

SW DATA BASE IDENTIFIER(1): NWIS

SW DATA BASE SUBSET IDENTIFIER(1): EE

SW DATA BASE DATE OF LAST UPDATE(1): 19870109

SW DATA BASE IDENTIFIER(2): NWIS

SW DATA BASE SUBSET IDENTIFIER(2): DV

SW DATA BASE DATE OF LAST UPDATE(2): 19870109

SW DATA BASE IDENTIFIER(3): NWIS

SW DATA BASE SUBSET IDENTIFIER(3): UV

SW DATA BASE DATE OF LAST UPDATE(3): 19870109

SW DATA BASE IDENTIFIER(4):

SW DATA BASE SUBSET IDENTIFIER(4):

SW DATA BASE DATE OF LAST UPDATE(4): 0

Example 4.--Validation listing for surface-water frequency output.

UNIQUE SITE IDENTIFIER: 420000004

AGENCY CODE: USGS

AGENCY STATION NUMBER: 01547200

SW YEAR OF MEASUREMENT: 1983

STAGE TYPE HISTORY: C

STAGE FREQUENCY HISTORY: C

FLOW TYPE HISTORY:

FLOW FREQUENCY HISTORY:

VOLUME TYPE HISTORY:

VOLUME FREQUENCY HISTORY:

VELOCITY TYPE HISTORY:

VELOCITY FREQUENCY HISTORY:

SW HISTORY DATE OF LAST UPDATE: 19870109

Example 5.--Validation listing for ground-water, quality-of-water, and meteorological transactions output.

UNIQUE SITE IDENTIFIER: 390000003

AGENCY CODE: USGS

AGENCY STATION NUMBER: 390425084230000

PARAMETER CODE: 45

YEAR OF MEASUREMENT: 1978

FREQUENCY CODE: C

SEASONAL CODE: S

DATA BASE IDENTIFIER: NWIS

DATA BASE SUBSET IDENTIFIER: UV

APPENDIX E. PSEUDO CODE

```
Declare variables and structures
Initialize variables and common blocks
Open files
Read MWDI input record
Read WATSTORE unit values record
Perform INPUT_SCREENING
Perform WATSTORE_SCREEN
WHILE (MWDI input file not empty and WATSTORE unit value file not empty)
  IF (state code has changed)
    THEN Perform PRINT_CONTROL_SUBS
  END IF
  Clear MWDI output record
  CASE (MWDI state code < WATSTORE state code or (MWDI state code =
        WATSTORE state code & MWDI agency code < WATSTORE agency code) or
       (MWDI state code = WATSTORE state code & MWDI agency code =
        WATSTORE agency code & MWDI station no < WATSTORE station no) or
        WATSTORE file empty)
    Perform INPUT_COUNTS
    Perform MOVE_ELEMENTS_TO_BE_KEPT
    Perform SW_RELATED_ELEMENTS
    Perform CONVERT_UV_ELEMENTS
    Perform ASSIGN_SW_DB_IDENTIFIERS
    Perform OUTPUT_COUNTS
    Write MWDI output record
    Read MWDI input record
    Call INPUT_SCREENING
  CASE (MWDI state code = WATSTORE state code &
        MWDI agency code = WATSTORE agency code &
        MWDI station number = WATSTORE station number)
    Perform INPUT_COUNTS
    Perform MOVE_ELEMENTS_TO_BE_KEPT
    Perform SW_RELATED_ELEMENTS
    Perform CONVERT_UV_ELEMENTS
    WHILE (MWDI station no = WATSTORE station no & WATSTORE not empty)
      If (WATSTORE record is surface water)
      THEN DO 1
        IF (previous record processed is not surface water)
          THEN DO 2
            Set flag to write record into non-surface water transaction file
            Perform OTHER_DATA_TRANSACTIONS
            Reset flag
          END THEN 2
        Set flag indicating that this is a surface water record
        Perform SW_FREQUENCY_COMPS
        Perform SW_ANN_FREQ_COMPS
        Perform SW_PERIOD_OF_RECORD
        Perform OTHER_SW_ACTIONS
        IF (flag indicating that this is a new surface water record is set)
          THEN Perform SW_ANN_FREQ_COMPS
        Read WATSTORE unit values record
        Perform WATSTORE_SCREEN
        IF (last record's station number = new record's station number &
            WATSTORE file is not empty)
```

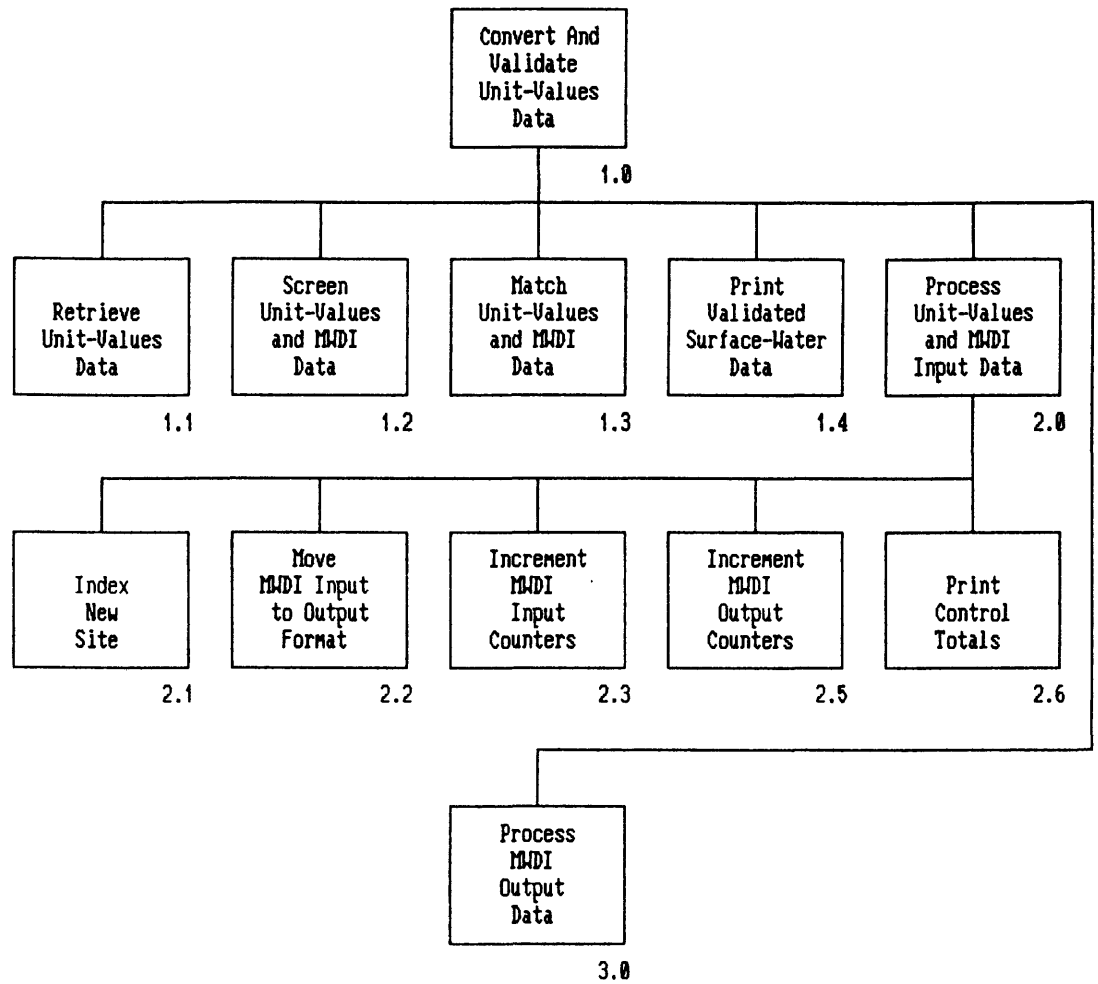
```
  THEN DO 3
    IF (new record's year number is not equal to last record's
          year number)
        THEN Perform SW_ANN_FREQ_COMPS
      END THEN 3
      ELSE DO 1
        Set flag to write out surface water transaction record
        Perform SW_ANN_FREQ_COMPS
        Perform SW_PERIOD_OF_RECORD
        Reset flags for writing out data and to denote that it is a
        surface water station
      END ELSE 1
  END THEN 1
  ELSE DO 2
    IF (WATSTORE record is not surface water data)
      THEN DO 4
        IF (last WATSTORE record was surface water data)
          THEN DO 5
            Set flag to write out surface water transaction record
            Perform SW_ANN_FREQ_COMPS
            Reset flags for writing out data and to denote that it is
            a surface water station
          END THEN 5
        Perform OTHER_DATA_TRANSACTIONS
        Read new WATSTORE unit values record
        Perform WATSTORE_SCREEN
        IF (last record's station number = new record's station
            number and WATSTORE file not empty)
          THEN DO 6
            IF (new record's year number not equal to last record's
                year number)
              THEN Perform OTHER_DATA_TRANSACTIONS
          END THEN 6
          ELSE DO 3
            Set flag to print non-surface water transaction record
            Perform OTHER_DATA_TRANSACTIONS
            Reset flag
          END ELSE 3
      END THEN 4
    ELSE DO 4
      IF (WATSTORE record parameter code = 74207)
       THEN DO 7
          Perform OTHER_DATA_AVAILABLE
          Read WATSTORE unit values record
          Perform WATSTORE_SCREEN
        END THEN 7
        ELSE DO 5
          Read WATSTORE unit values record
          Perform WATSTORE_SCREEN
        END ELSE 5
    END ELSE 4
  END ELSE 2
END WHILE
Perform ASSIGN_SW_DB_IDENTIFIERS
```

```
  Reset flag indicating current record is surface water data
  Perform OUTPUT_COUNTS
  Write MWDI output record
  Read MWDI input record
  Perform INPUT_SCREENING
CASE (MWDI state code > WATSTORE state code or (MWDI state code =
      WATSTORE state code & MWDI agency code > WATSTORE agency code) or
     (MWDI state code = WATSTORE state code & MWDI agency code =
      WATSTORE agency code & MWDI station no > WATSTORE station no) or
      MWDI file empty)
  Perform INDEX_NEW_SITE
  WHILE (MWDI station no = WATSTORE station no & WATSTORE not empty)
    If (WATSTORE record is surface water)
    THEN DO 8
      IF (previous record processed is not surface water)
     THEN DO 9
       Set flag to write record into non-surface water transaction file
          Perform OTHER_DATA_TRANSACTIONS
          Reset flag
        END THEN 9
      Set flag indicating that this is a surface water record
      Perform SW_FREQUENCY_COMPS
      Perform SW_ANN_FREQ_COMPS
      Perform SW_PERIOD_OF_RECORD
      Perform OTHER_SW_ACTIONS
      IF (flag indicating that this is a new surface water record is set)
        THEN Perform SW_ANN_FREQ_COMPS
      Read WATSTORE unit values record
      Perform WATSTORE_SCREEN
      IF (last record's station number = new record's station number &
          WATSTORE file is not empty)
        THEN DO 10
          IF (new record's year number is not equal to last record's
              year number)
            THEN Perform SW_ANN_FREQ_COMPS
        END THEN 10
        ELSE DO 6
          Set flag to write out surface water transaction record
          Perform SW_ANN_FREQ_COMPS
          Perform SW_PERIOD_OF_RECORD
          Reset flags for writing out data and to denote that it is a
          surface water station
        END ELSE 6
    END THEN 8
    ELSE DO 7
      IF (WATSTORE record is not surface water data)
        THEN DO 11
          IF (last WATSTORE record was surface water data)
            THEN DO 12
              Set flag to write out surface water transaction record
              Perform SW_ANN_FREQ_COMPS
              Reset flags for writing out data and to denote that it is
              a surface water station
            END THEN 12
```

```
        Perform OTHER_DATA_TRANSACTIONS
        Read new WATSTORE unit values record
         Perform WATSTORE_SCREEN
           IF (last record's station number = new record's station
               number and WATSTORE file not empty)
             THEN DO 13
               IF (new record's year number not equal to last record's
                   year number)
                 THEN Perform OTHER_DATA_TRANSACTIONS
             END THEN 13
             ELSE DO 8
               Set flag to print non-surface water transaction record
               Perform OTHER_DATA_TRANSACTIONS
               Reset flag
             END ELSE 8
         END THEN 11
       ELSE DO 9
         IF (WATSTORE record parameter code = 74207)
          THEN DO 14
             Perform OTHER_DATA_AVAILABLE
             Read WATSTORE unit values record
             Perform WATSTORE_SCREEN
           END THEN 14
           ELSE DO 10
             Read WASTORE unit values record
             Perform WATSTORE_SCREEN
           END ELSE 10
       END ELSE 9
     END ELSE 7
   END WHILE
   Perform ASSIGN_SW_DB_IDENTIFIERS
   Reset flag indicating current record is surface water data
   Perform OUTPUT_COUNTS
   Write MWDI output record
  OTHERWISE
   Print error message
   Stop program
  END CASE
END WHILE
Perform PRINT_CONTROL_SUBS
Perform PRINT_CONTROL_TOTALS
Perform PRINT_VALIDATED_SW_DATA
Close files
Stop program
```

APPENDIX F. ORGANIZATION CHART AND DATA FLOW DIAGRAMS

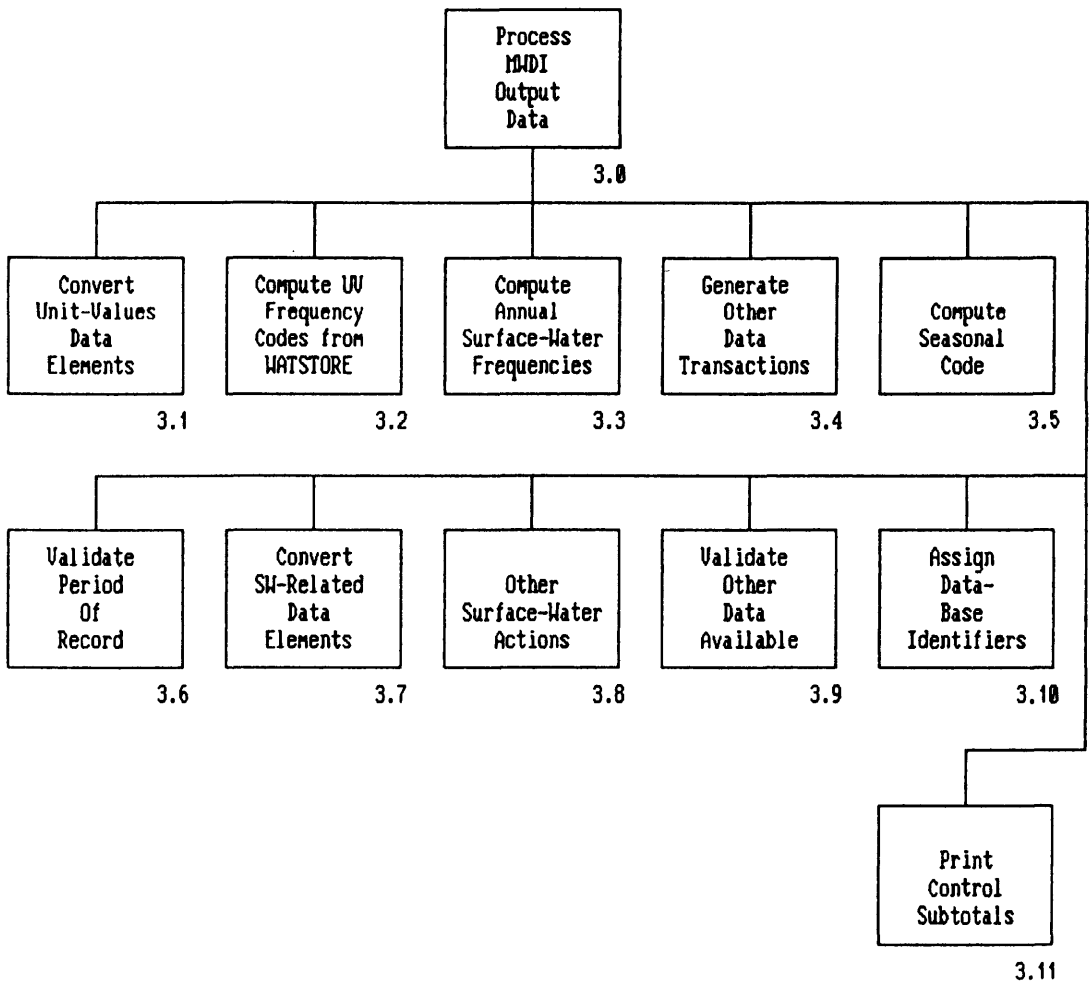
VTOC NO. 1.0,2.0	Convert and Validate Unit-Values Data for the MWDI	Page 1 of 14



Developed:
Date: 06/01/1987

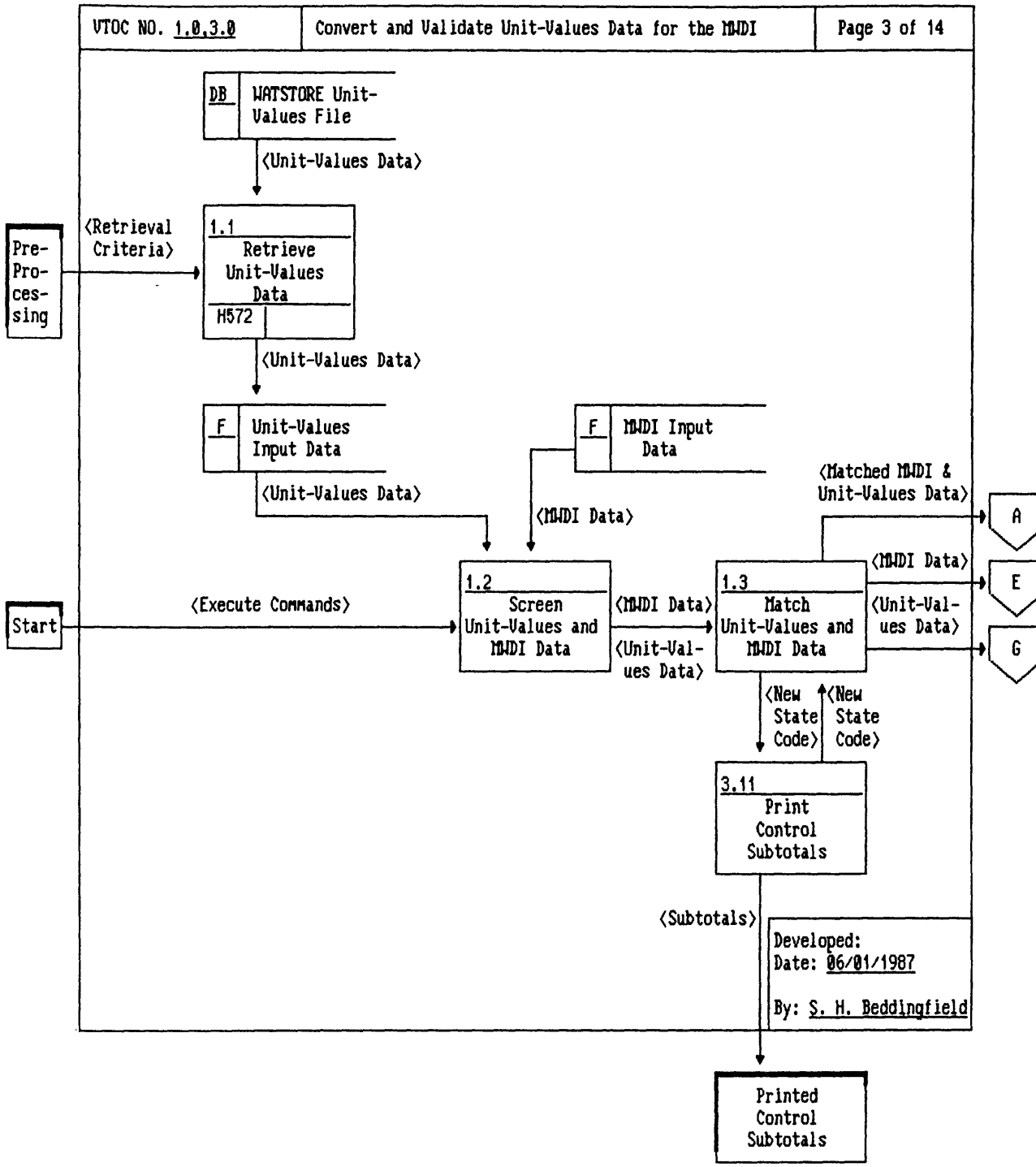
By: S. H. Beddingfield

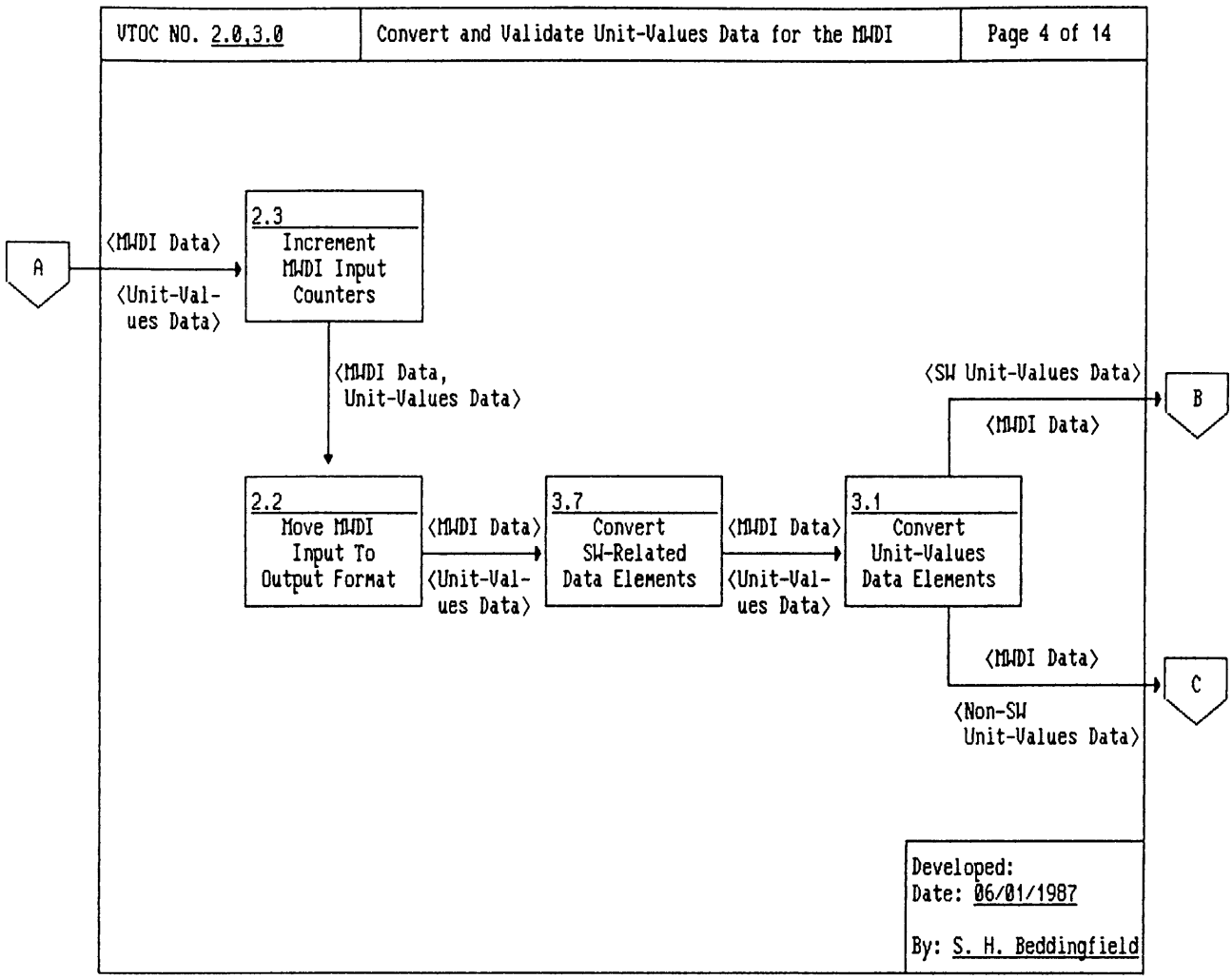
VTOC NO. 3.0	Convert and Validate Unit-Values Data for the MWDI	Page 2 of 14

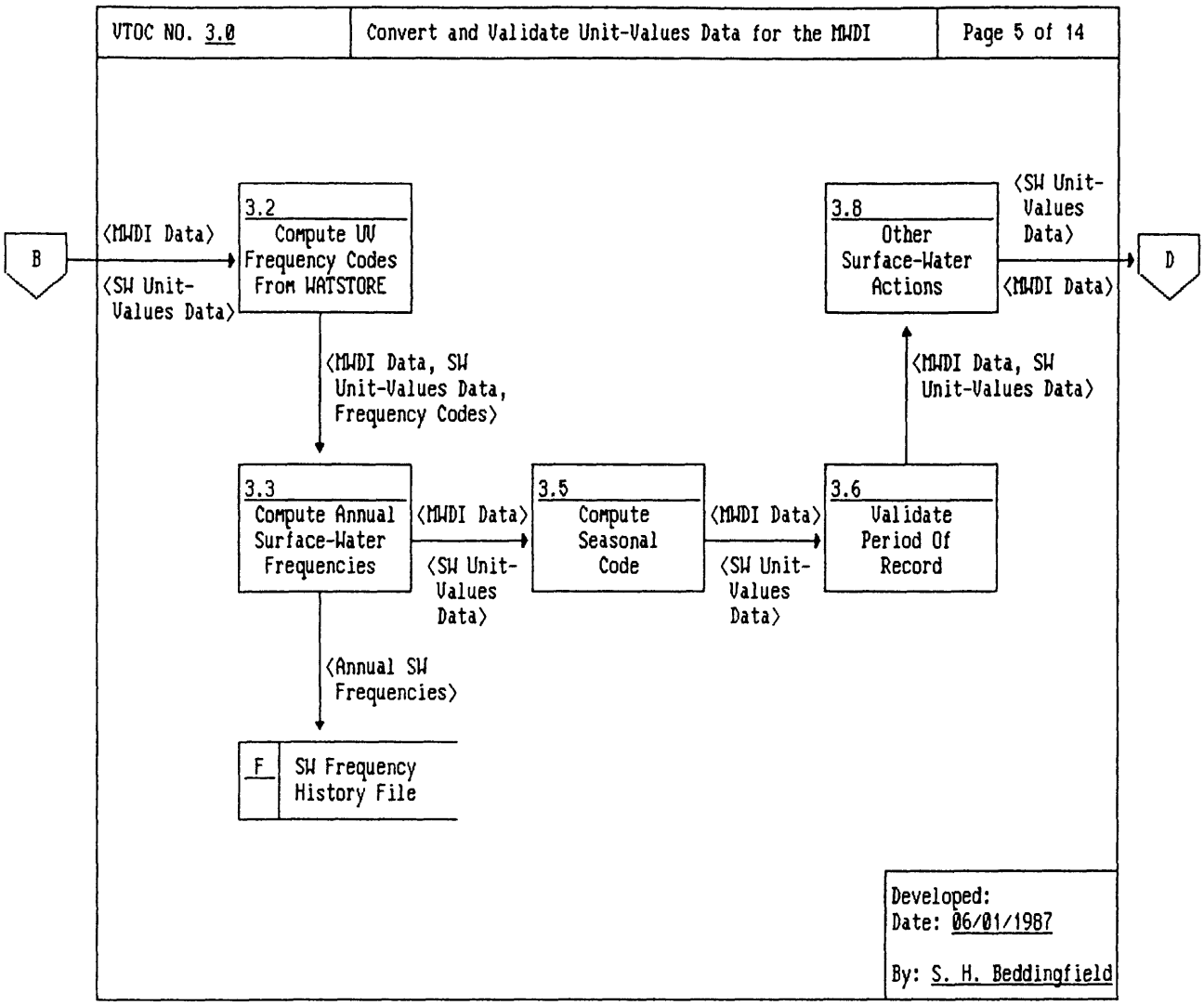


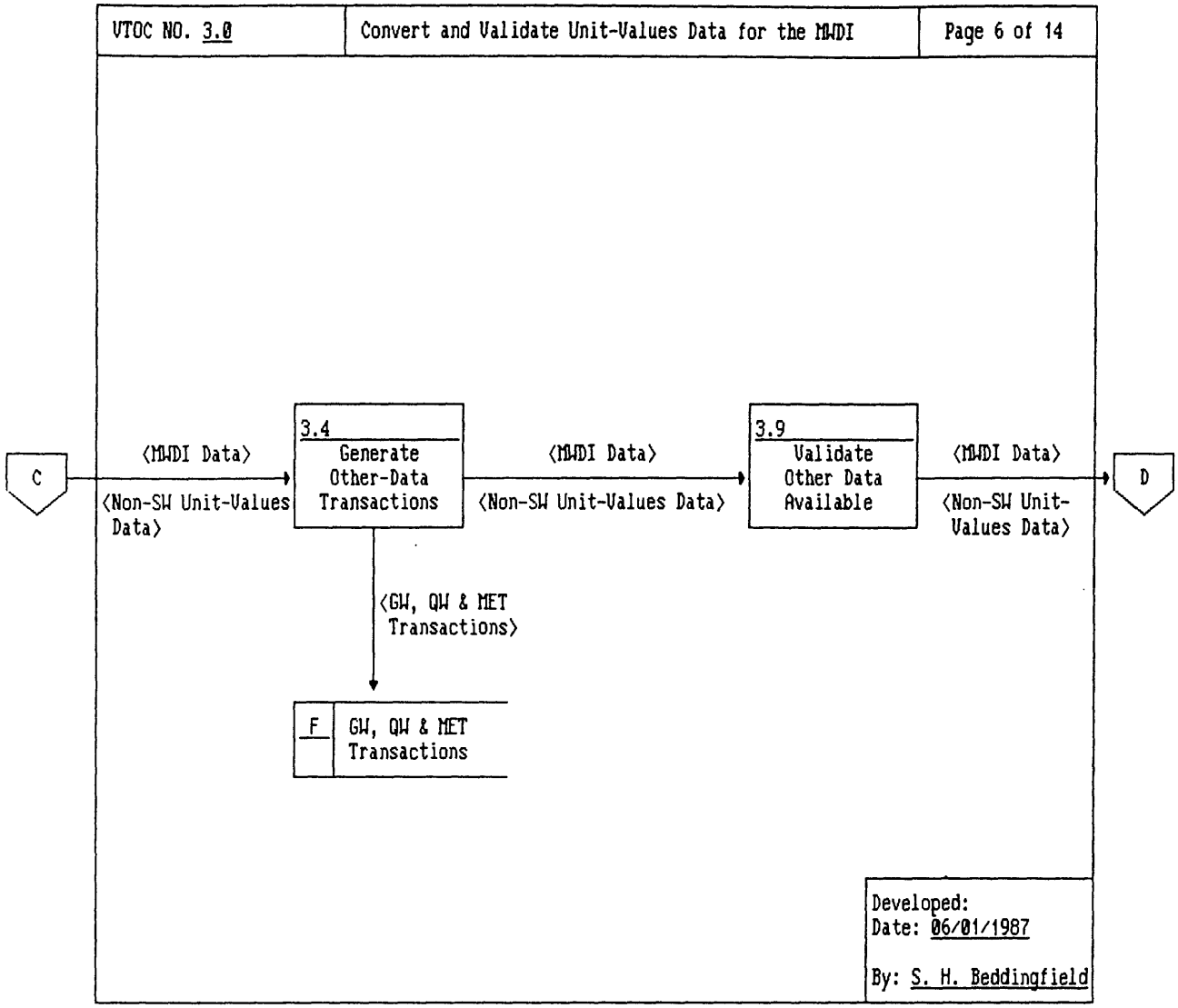
Developed:
Date: 06/01/1987

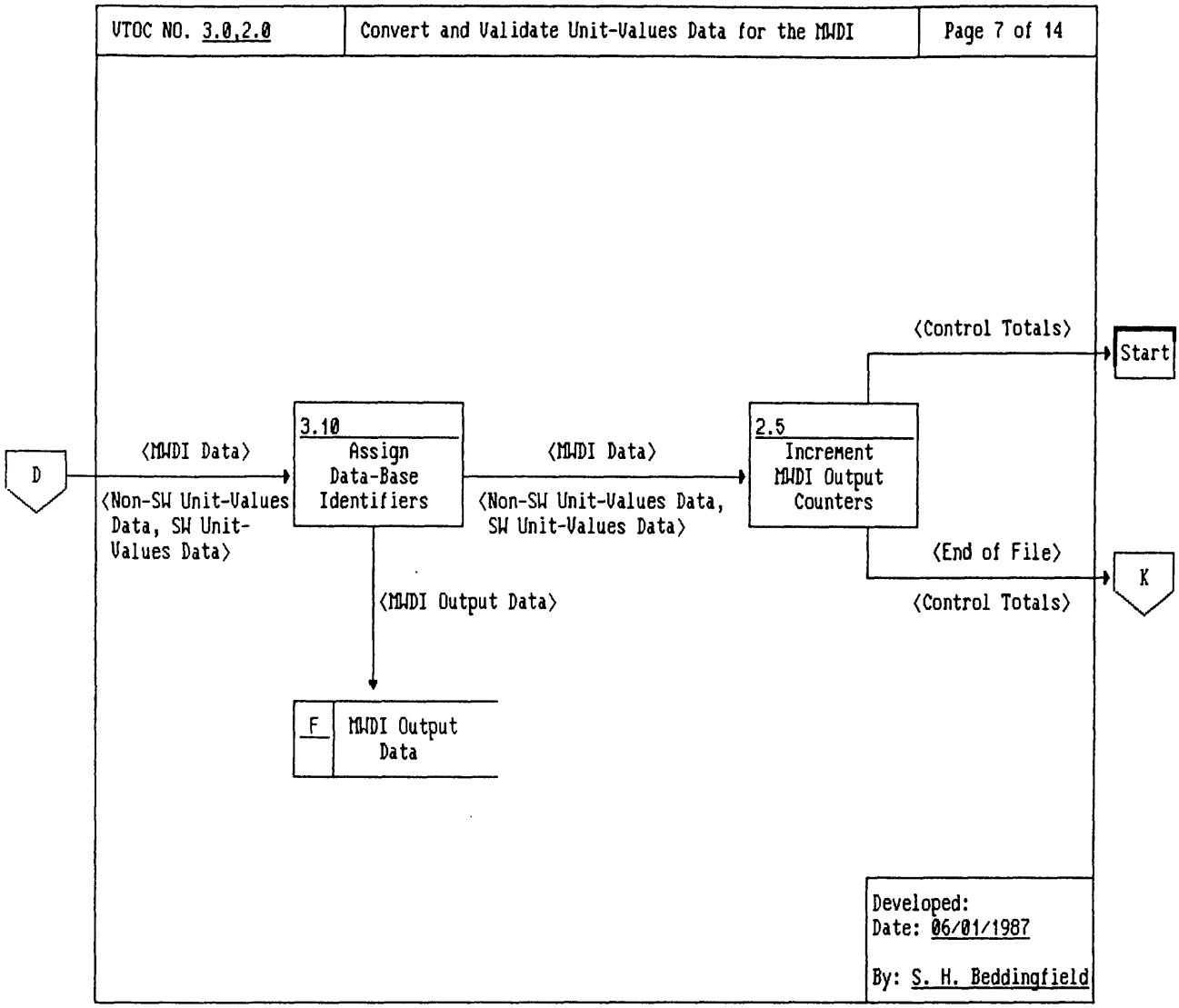
By: S. H. Beddingfield

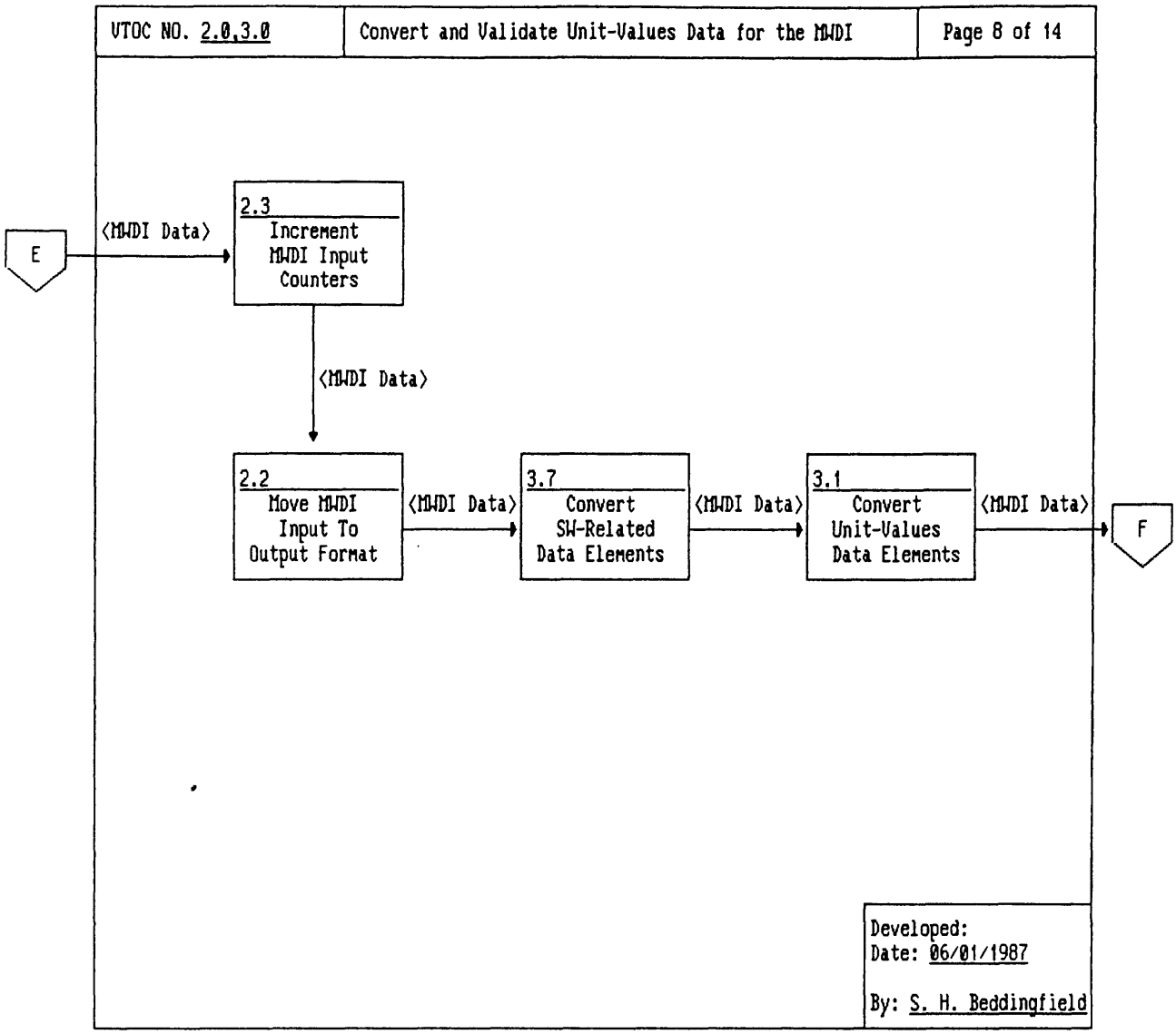

VTOC NO. 1.0,3.0
Convert and Validate Unit-Values Data for the MWDI
Page 3 of 14
DB
WATSTORE Unit-Values File
⟨Unit-Values Data⟩
Pre-Processing
⟨Retrieval Criteria⟩
1.1
Retrieve Unit-Values Data
H572
⟨Unit-Values Data⟩
F
Unit-Values Input Data
⟨Unit-Values Data⟩
F
MWDI Input Data
⟨MWDI Data⟩
⟨Matched MWDI & Unit-Values Data⟩
A
⟨MWDI Data⟩
E
Start
⟨Execute Commands⟩
1.2
Screen Unit-Values and MWDI Data
⟨MWDI Data⟩
⟨Unit-Values Data⟩
1.3
Match Unit-Values and MWDI Data
⟨Unit-Values Data⟩
G
⟨New State Code⟩
⟨New State Code⟩
3.11
Print Control Subtotals
⟨Subtotals⟩
Developed:
Date: 06/01/1987
By: S. H. Beddingfield
Printed Control Subtotals

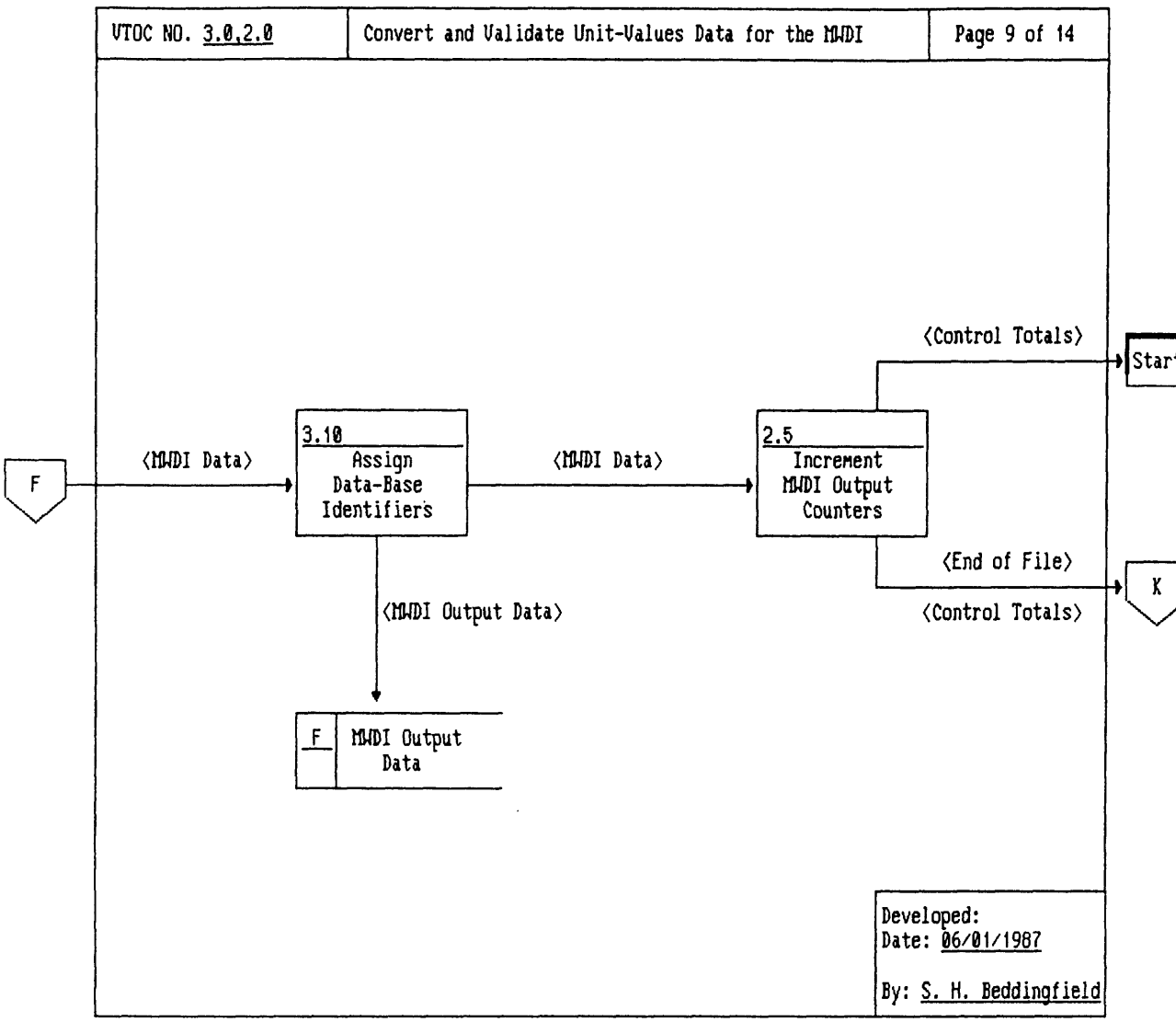

VTOC NO. 2.0,3.0
Convert and Validate Unit-Values Data for the MWDI
Page 4 of 14
A
⟨MWDI Data⟩
⟨Unit-Val-ues Data⟩
2.3
Increment MWDI Input Counters
⟨MWDI Data, Unit-Values Data⟩
2.2
Move MWDI Input To Output Format
⟨MWDI Data⟩
⟨Unit-Val-ues Data⟩
3.7
Convert SW-Related Data Elements
⟨MWDI Data⟩
⟨Unit-Val-ues Data⟩
3.1
Convert Unit-Values Data Elements
⟨SW Unit-Values Data⟩
B
⟨MWDI Data⟩
⟨MWDI Data⟩
C
⟨Non-SW Unit-Values Data⟩
Developed:
Date: 06/01/1987
By: S. H. Beddingfield

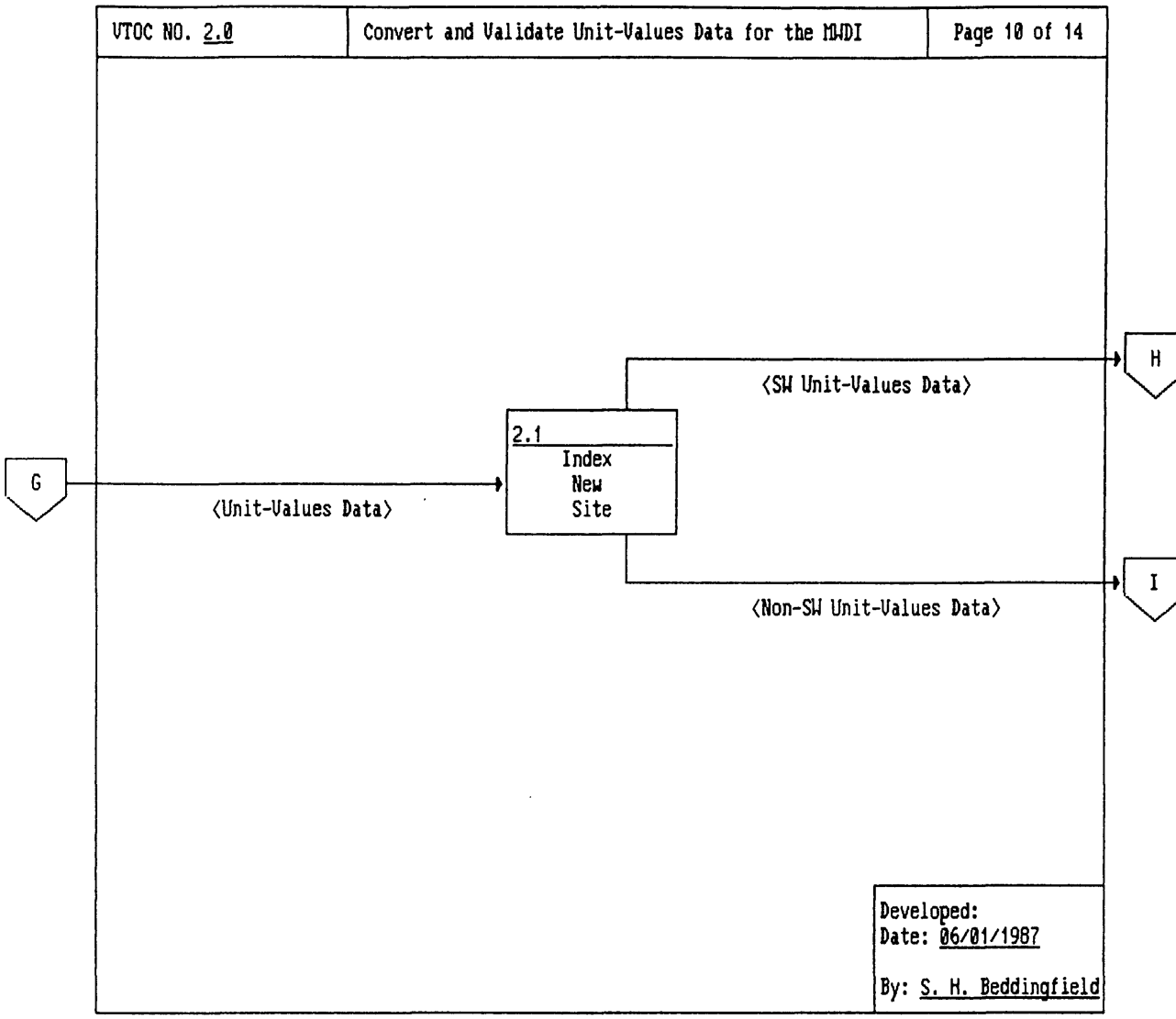

VTOC NO. 3.0
Convert and Validate Unit-Values Data for the MWDI
Page 5 of 14
B
<MWDI Data>
<SW Unit-Values Data>
3.2
Compute UV Frequency Codes From WATSTORE
<MWDI Data, SW Unit-Values Data, Frequency Codes>
3.3
Compute Annual Surface-Water Frequencies
<MWDI Data>
<SW Unit-Values Data>
3.5
Compute Seasonal Code
<MWDI Data>
<SW Unit-Values Data>
3.6
Validate Period Of Record
<MWDI Data, SW Unit-Values Data>
3.8
Other Surface-Water Actions
<SW Unit-Values Data>
<MWDI Data>
D
<Annual SW Frequencies>
F
SW Frequency History File
Developed:
Date: 06/01/1987
By: S. H. Beddingfield

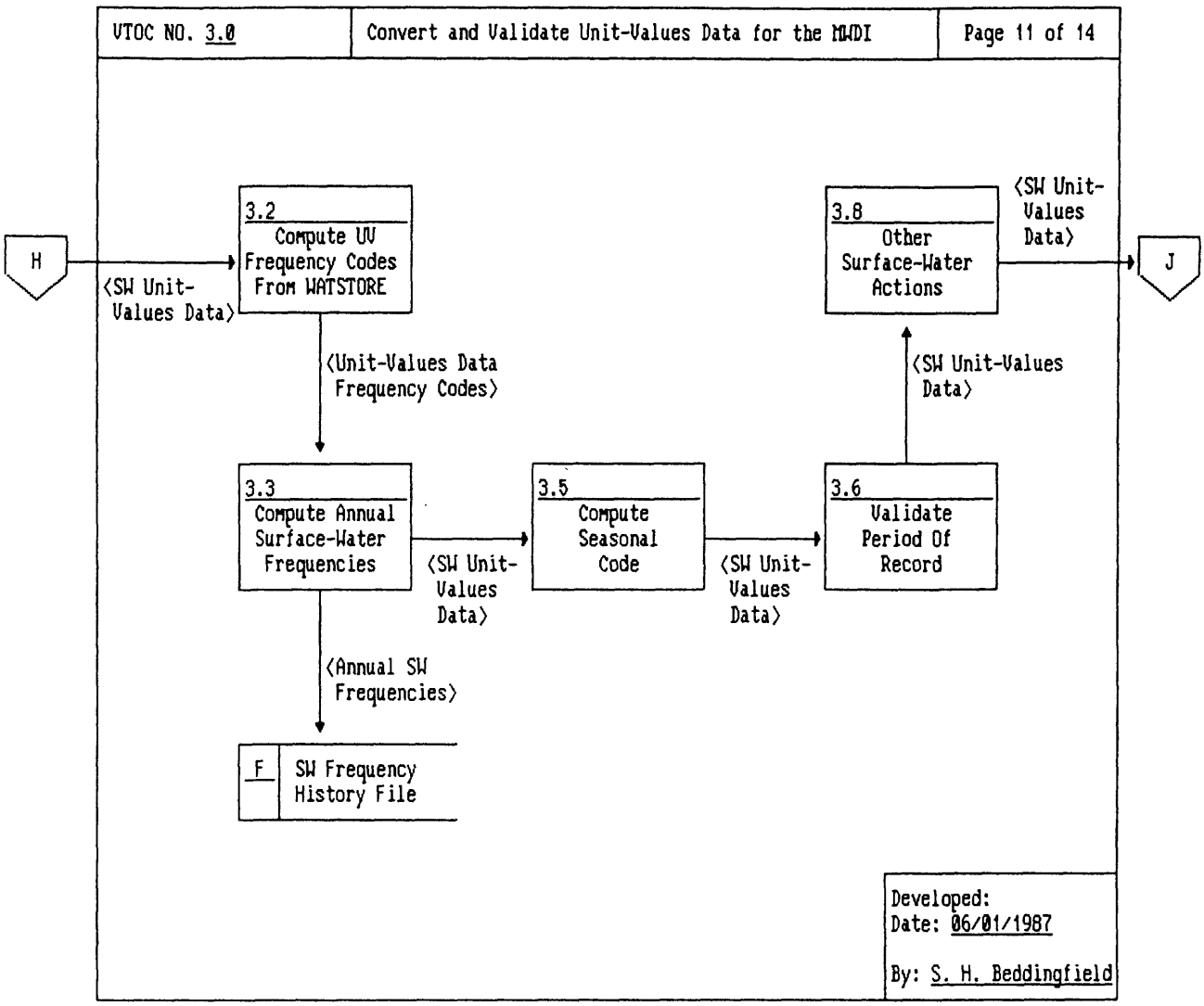

VTOC NO. 3.0
Convert and Validate Unit-Values Data for the MWDI
Page 6 of 14
C
⟨MWDI Data⟩
⟨Non-SW Unit-Values Data⟩
3.4
Generate Other-Data Transactions
⟨MWDI Data⟩
⟨Non-SW Unit-Values Data⟩
3.9
Validate Other Data Available
⟨MWDI Data⟩
⟨Non-SW Unit-Values Data⟩
D
⟨GW, QW & MET Transactions⟩
F
GW, QW & MET Transactions
Developed:
Date: 06/01/1987
By: S. H. Beddingfield

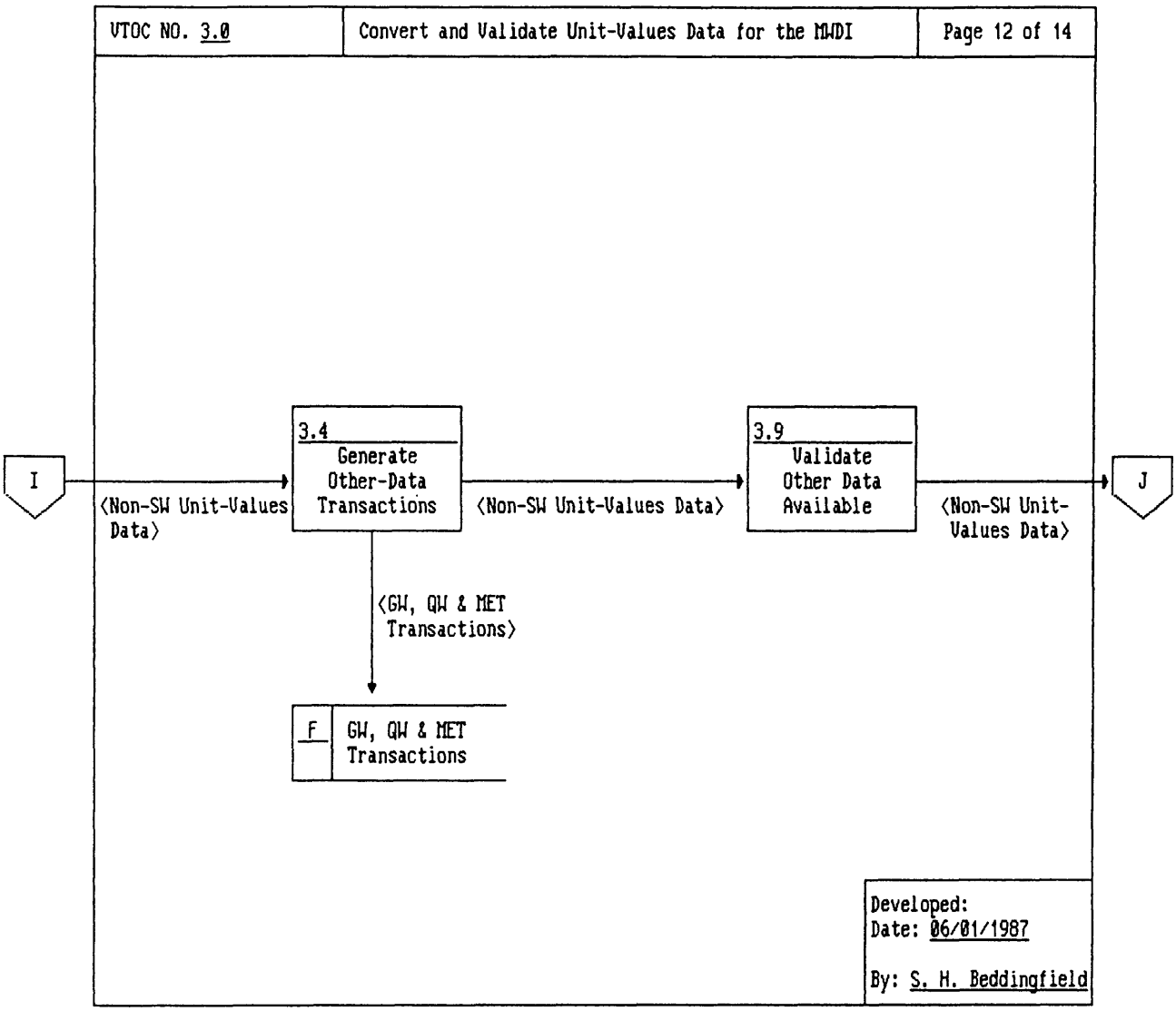

VTOC NO. 3.0,2.0
Convert and Validate Unit-Values Data for the MWDI
Page 7 of 14
⟨Control Totals⟩
Start
D
⟨MWDI Data⟩
⟨Non-SW Unit-Values Data, SW Unit-Values Data⟩
3.10
Assign Data-Base Identifiers
⟨MWDI Data⟩
⟨Non-SW Unit-Values Data, SW Unit-Values Data⟩
2.5
Increment MWDI Output Counters
⟨End of File⟩
⟨Control Totals⟩
K
⟨MWDI Output Data⟩
F
MWDI Output Data
Developed:
Date: 06/01/1987
By: S. H. Beddingfield

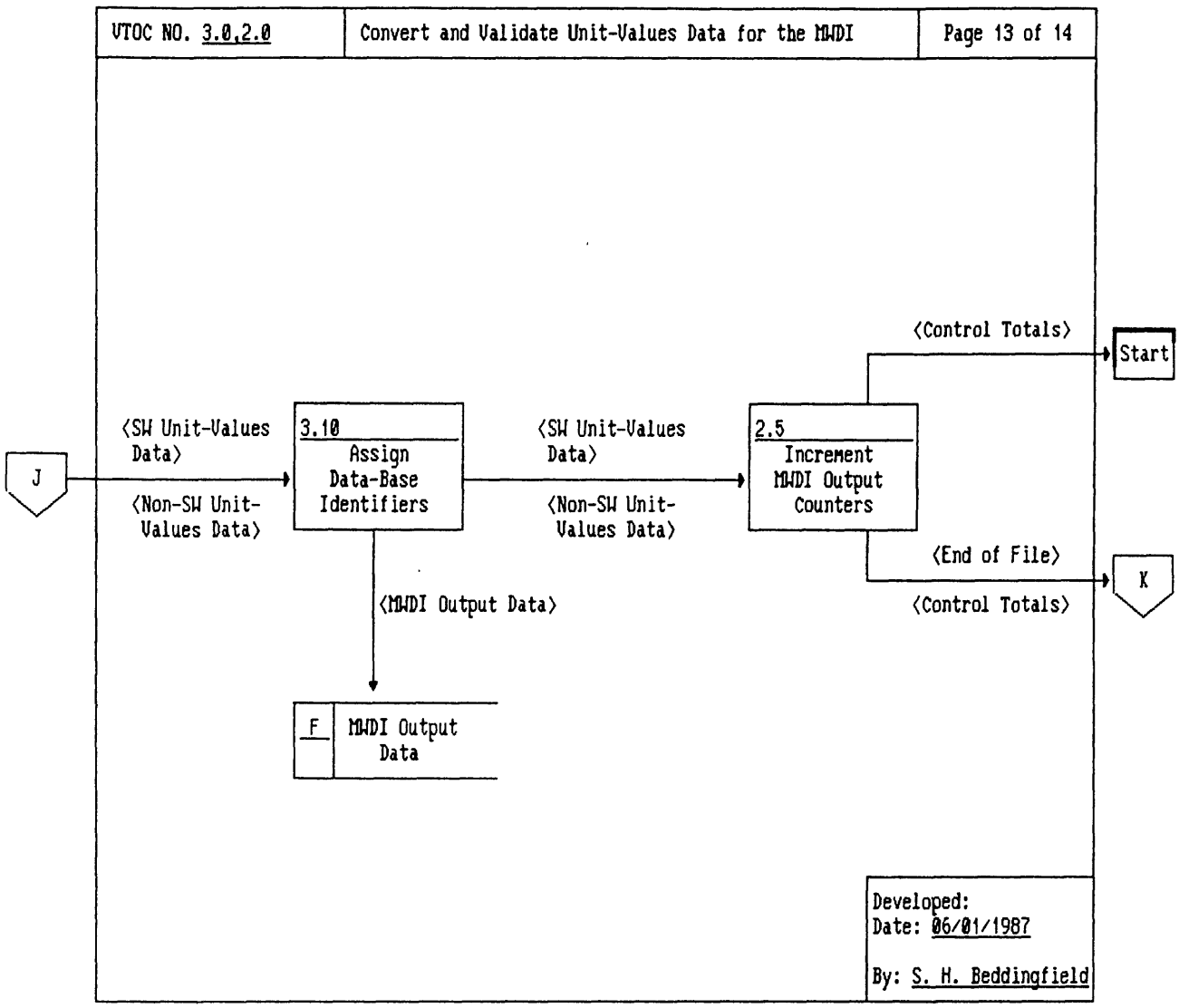

UTOC NO. 2.0,3.0
Convert and Validate Unit-Values Data for the MWDI
Page 8 of 14
E
<MWDI Data>
2.3
Increment MWDI Input Counters
<MWDI Data>
2.2
Move MWDI Input To Output Format
<MWDI Data>
3.7
Convert SW-Related Data Elements
<MWDI Data>
3.1
Convert Unit-Values Data Elements
<MWDI Data>
F
Developed:
Date: 06/01/1987
By: S. H. Beddingfield

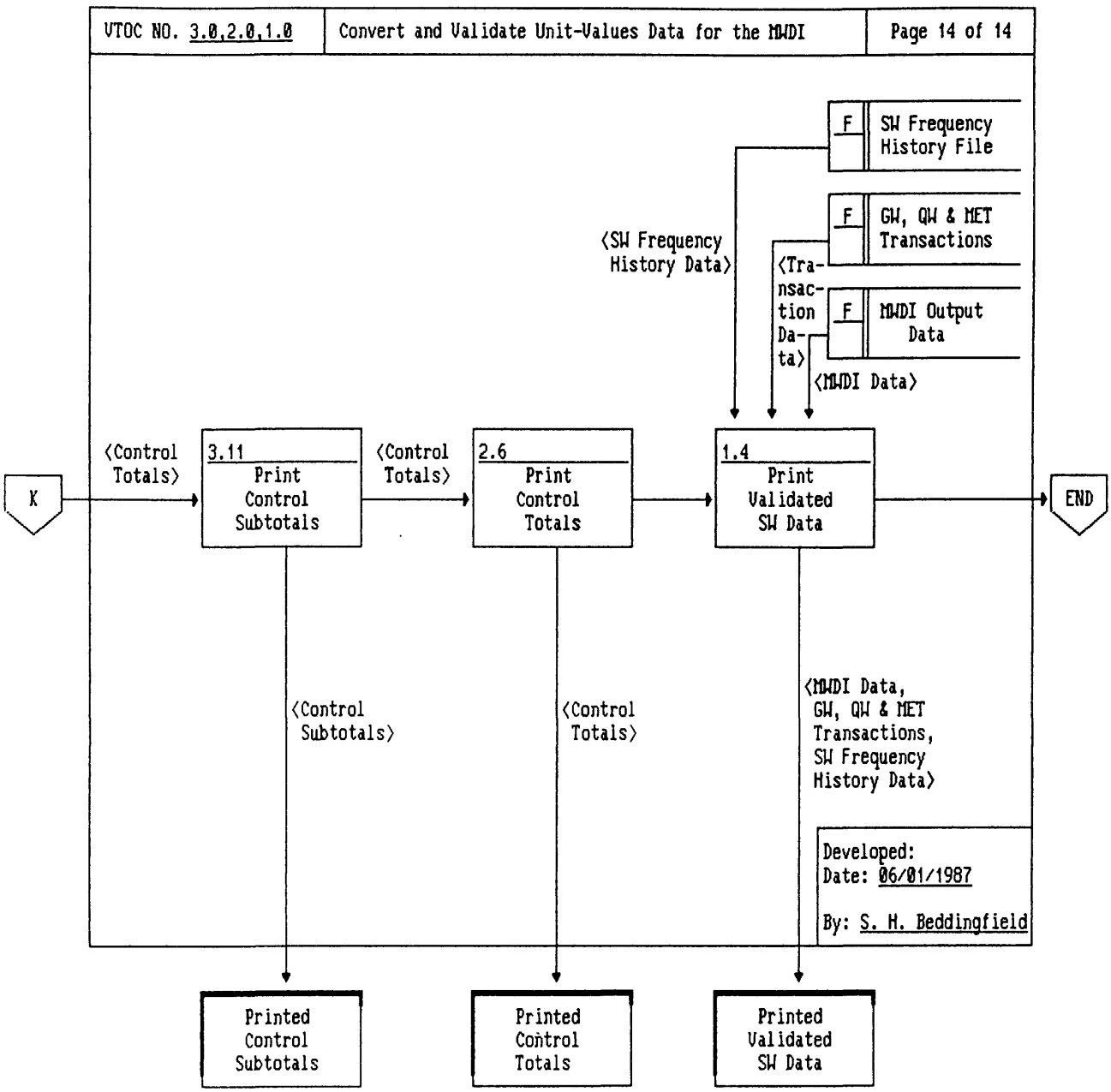

VTOC NO. 3.0,2.0
Convert and Validate Unit-Values Data for the MWDI
Page 9 of 14
⟨Control Totals⟩
Start
F
⟨MWDI Data⟩
3.10
Assign
Data-Base
Identifiers
⟨MWDI Data⟩
2.5
Increment
MWDI Output
Counters
⟨End of File⟩
K
⟨MWDI Output Data⟩
⟨Control Totals⟩
F
MWDI Output
Data
Developed:
Date: 06/01/1987
By: S. H. Beddingfield


VTOC NO. 2.0
Convert and Validate Unit-Values Data for the MWDI
Page 10 of 14
G
⟨Unit-Values Data⟩
2.1
Index
New
Site
⟨SW Unit-Values Data⟩
H
⟨Non-SW Unit-Values Data⟩
I
Developed:
Date: 06/01/1987
By: S. H. Beddingfield


VTOC NO. 3.0
Convert and Validate Unit-Values Data for the MWDI
Page 11 of 14
H
3.2
Compute UV Frequency Codes From WATSTORE
⟨SW Unit-Values Data⟩
⟨Unit-Values Data Frequency Codes⟩
3.3
Compute Annual Surface-Water Frequencies
⟨SW Unit-Values Data⟩
3.5
Compute Seasonal Code
⟨SW Unit-Values Data⟩
3.6
Validate Period Of Record
⟨SW Unit-Values Data⟩
3.8
Other Surface-Water Actions
⟨SW Unit-Values Data⟩
J
⟨Annual SW Frequencies⟩
F
SW Frequency History File
Developed:
Date: 06/01/1987
By: S. H. Beddingfield


VTOC NO. 3.0
Convert and Validate Unit-Values Data for the MWDI
Page 12 of 14
I
<Non-SW Unit-Values Data>
3.4
Generate Other-Data Transactions
<Non-SW Unit-Values Data>
3.9
Validate Other Data Available
<Non-SW Unit-Values Data>
J
<GW, QW & MET Transactions>
F
GW, QW & MET Transactions
Developed:
Date: 06/01/1987
By: S. H. Beddingfield


UTOC NO. 3.0,2.0
Convert and Validate Unit-Values Data for the MWDI
Page 13 of 14
⟨Control Totals⟩
Start
J
⟨SW Unit-Values Data⟩
⟨Non-SW Unit-Values Data⟩
3.10
Assign Data-Base Identifiers
⟨SW Unit-Values Data⟩
⟨Non-SW Unit-Values Data⟩
2.5
Increment MWDI Output Counters
⟨End of File⟩
⟨Control Totals⟩
K
⟨MWDI Output Data⟩
F
MWDI Output Data
Developed:
Date: 06/01/1987
By: S. H. Beddingfield


UTOC NO. 3.0,2.0,1.0
Convert and Validate Unit-Values Data for the MWDI
Page 14 of 14
F
SW Frequency History File
F
GW, QW & MET Transactions
F
MWDI Output Data
⟨SW Frequency History Data⟩
⟨Transaction Data⟩
⟨MWDI Data⟩
K
⟨Control Totals⟩
3.11
Print Control Subtotals
⟨Control Totals⟩
2.6
Print Control Totals
1.4
Print Validated SW Data
END
⟨Control Subtotals⟩
⟨Control Totals⟩
⟨MWDI Data, GW, QW & MET Transactions, SW Frequency History Data⟩
Developed:
Date: 06/01/1987
By: S. H. Beddingfield
Printed Control Subtotals
Printed Control Totals
Printed Validated SW Data

APPENDIX G. CROSS-REFERENCE LISTING

Specifications (VTOCS)	Structure Chart	Program	Subroutine/Section
1.9.3.2.1.2.1	1.0	IXCVUV.DB	
1.9.3.2.1.2.1.1	1.1	H572 (WATSTORE)	
1.9.3.2.1.2.1.2	2.2	IXCVUV.DB	MOVE_ELEMENTS_TO_BE_KEPT
1.9.3.2.1.2.1.3	3.1	IXCVUV.DB	CONVERT_UV_ELEMENTS
1.9.3.2.1.2.1.4	3.2	IXCVUV.DB	SW_FREQUENCY_COMPS
1.9.3.2.1.2.1.5	3.3	IXCVUV.DB	SW_ANN_FREQ_COMPS
1.9.3.2.1.2.1.6	3.4	IXCVUV.DB	OTHER_DATA_TRANSACTIONS
1.9.3.2.1.2.1.7	3.5	IXCVUV.DB	SW_ANN_FREQ_COMPS
1.9.3.2.1.2.1.8	3.6	IXCVUV.DB	SW_PERIOD_OF_RECORD
1.9.3.2.1.2.1.9	3.7	IXCVUV.DB	SW_RELATED_ELEMENTS
1.9.3.2.1.2.1.10	3.8	IXCVUV.DB	OTHER_SW_ACTIONS
1.9.3.2.1.2.1.11	3.8	IXCVUV.DB	OTHER_SW_ACTIONS
1.9.3.2.1.2.1.12	2.2	IXCVUV.DB	MOVE_ELEMENTS_TO_BE_KEPT
1.9.3.2.1.2.1.13	3.9	IXCVUV.DB	OTHER_DATA_AVAILABLE
1.9.3.2.1.2.1.14	3.8	IXCVUV.DB	OTHER_SW_ACTIONS
1.9.3.2.1.2.1.15	3.8	IXCVUV.DB	OTHER_SW_ACTIONS
1.9.3.2.1.2.1.16	3.10	IXCVUV.DB	ASSIGN_SW_DB_IDENTIFIERS
1.9.3.2.1.2.1.17	2.1	IXCVUV.DB	INDEX_NEW_SITE
1.9.3.2.1.2.1.18	2.6	IXCVUV.DB	PRINT_CONTROL_TOTALS
1.9.3.2.1.2.1.19	1.4	IXCVUV.DB	PRINT_VALIDATED_SW_DATA
	1.2	IXCVUV.DB	INPUT_SCREENING
	1.2	IXCVUV.DB	STRIP_BLANKS
	1.2	IXCVUV.DB	WATSTORE_SCREEN
	1.3	IXCVUV.DB	MAIN ROUTINE
	2.3	IXCVUV.DB	INPUT_COUNTS
	2.4	IXCVUV.DB	ALL SUBROUTINES EXCEPT THOSE CONCERNED WITH I/O
	2.5	IXCVUV.DB	OUTPUT_COUNTS
	3.11	IXCVUV.DB	PRINT_CONTROL_SUBS

APPENDIX H. SOURCE CODE

```
IXCVUV:PROC OPTIONS(MAIN);
DECLARE 1 MWDI_INPUT,
          2 SITE_DESCRIPTOR_DATA,
            3 SITE_ID FIXED BINARY(31) INIT(0),
            3 AGENCY_CODE CHAR(5) INIT(''),
            3 AGENCY_STATION_NO CHAR(15) INIT(''),
            3 LOCAL_WELL_NO CHAR(24) INIT(''),
            3 STATION_NAME CHAR(48) INIT(''),
            3 LATITUDE FIXED BINARY(31) INIT(0),
            3 LONGITUDE FIXED BINARY(31) INIT(0),
            3 LAT_LONG_ACCURACY CHAR(1) INIT(''),
            3 SITE_TYPE CHAR(2) INIT(''),
            3 COUNTRY_CODE CHAR(2) INIT(''),
            3 RESERVED1 CHAR(1) INIT(''),
            3 STATE_CODE FIXED BINARY(15) INIT(0),
            3 COUNTY_CODE FIXED BINARY(15) INIT(0),
            3 RESERVED2 CHAR(2) INIT(''),
            3 STATE_COUNTY_CODE FIXED BINARY(31) INIT(0),
            3 DISTRICT_CODE CHAR(2) INIT(''),
            3 RESERVED3 CHAR(2) INIT(''),
            3 HYDRO_UNIT_CODE FIXED BINARY(31) INIT(0),
            3 OFFICE_CODE FIXED BINARY(31) INIT(0),
            3 TOTAL_DRAINAGE_AREA FLOAT DEC(6) INIT(0),
            3 CONTRIB_DRAINAGE_AREA FLOAT DEC(6) INIT(0),
            3 ALTITUDE_DATUM FLOAT DEC(6) INIT(0),
            3 ALTITUDE_METH_OF_MEAS CHAR(1) INIT(''),
            3 RESERVED4 CHAR(3) INIT(''),
            3 ALTITUDE_ACCURACY FLOAT DEC(6) INIT(0),
            3 RIVER_REACH_NO FIXED BINARY(15) INIT(0),
            3 RIVER_REACH_SEG_FLAG FIXED BINARY(15) INIT(0),
            3 DEPTH_OF_WELL FLOAT DEC(6) INIT(0),
            3 SOURCE_WELL_DEPTH_DATA CHAR(1) INIT(''),
            3 RESERVED5 CHAR(3) INIT(''),
            3 DEPTH_OF_HOLE FLOAT DEC(6) INIT(0),
            3 PRINCIPAL_AQUIFER_CODE CHAR(8) INIT(''),
            3 AQUIFER_TYPE_CODE CHAR(1) INIT(''),
            3 CONSOLIDATED_AQUIFER_CODE CHAR(1) INIT(''),
            3 BASIN_DESCRIPTOR(3) CHAR(1) INIT((3)''),
            3 USE_OF_WATER(3) CHAR(1) INIT((3)''),
            3 USE_OF_SITE(3) CHAR(1) INIT((3)''),
            3 OTHER_DATA_AVAILABLE(6) CHAR(1) INIT((6)''),
            3 SW_ACTIVE_CODE CHAR(1) INIT(''),
            3 GW_ACTIVE_CODE CHAR(1) INIT(''),
            3 QW_ACTIVE_CODE CHAR(1) INIT(''),
            3 BIO_ACTIVE_CODE CHAR(1) INIT(''),
            3 PHY_ACTIVE_CODE CHAR(1) INIT(''),
            3 SED_ACTIVE_CODE CHAR(1) INIT(''),
            3 CHM_ACTIVE_CODE CHAR(1) INIT(''),
            3 MET_ACTIVE_CODE CHAR(1) INIT(''),
            3 RESERVED6 CHAR(3) INIT(''),
            3 DATE_OF_LAST_UPDATE FIXED BINARY(31) INIT(0),
          2 NODE_LOCATION_INFO,
            3 DIS_NODE_IDENT CHAR(6) INIT(''),
            3 SW_DATA_CODE CHAR(1) INIT(''),
```

```
  3 GW_DATA_CODE CHAR(1) INIT(''),
  3 QW_DATA_CODE CHAR(1) INIT(''),
  3 MET_DATA_CODE CHAR(1) INIT(''),
  3 OTHER_DATA_CODE CHAR(1) INIT(''),
  3 RESERVED7 CHAR(1) INIT(''),
  3 NODE_DATE_OF_LAST_UPDATE FIXED BINARY(31) INIT(0),
2 SURFACE_WATER_DATA,
  3 SW_BEGIN_YEAR PIC '999R' INIT('0000'),
  3 SW_END_YEAR PIC '999R' INIT('0000'),
  3 SW_INTERRUPTED_CODE CHAR(1) INIT(''),
  3 COMPLETE_STAGE CHAR(1) INIT(''),
  3 PEAK_STAGE CHAR(1) INIT(''),
  3 LOW_STAGE CHAR(1) INIT(''),
  3 STAGE_STORAGE_MEDIA CHAR(1) INIT(''),
  3 COMPLETE_FLOW CHAR(1) INIT(''),
  3 PEAK_FLOW CHAR(1) INIT(''),
  3 LOW_FLOW CHAR(1) INIT(''),
  3 MISC_FLOW_MEAS CHAR(1) INIT(''),
  3 FLOW_STORAGE_MEDIA CHAR(1) INIT(''),
  3 VOLUME CHAR(1) INIT(''),
  3 VOLUME_CHANGE CHAR(1) INIT(''),
  3 VOLUME_STORAGE_MEDIA CHAR(1) INIT(''),
  3 UNIT_FLOW CHAR(1) INIT(''),
  3 UNIT_STAGE CHAR(1) INIT(''),
  3 UNIT_VOLUME CHAR(1) INIT(''),
  3 SW_OTHER_DATA_AVAIL CHAR(12) INIT(''),
  3 SW_TELEMETRY_CODE PIC 'R' INIT('0'),
  3 SW_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 SW_PURPOSE_CODE CHAR(9) INIT(''),
  3 SW_RECORDER_TYPE_CODE CHAR(1) INIT(''),
  3 SW_RECORDER_FREQ_CODE CHAR(1) INIT(''),
2 SW_MODIFIERS(10),
  3 SW_POINTER CHAR(7) INIT((10)''),
  3 SW_MOD_FILE CHAR(1) INIT((10)''),
2 GROUND_WATER_DATA,
  3 GW_BEGIN_YEAR PIC '999R' INIT('0000'),
  3 GW_END_YEAR PIC '999R' INIT('0000'),
  3 GW_INTERRUPTED_CODE CHAR(1) INIT(''),
  3 WATER_LEVEL_FREQ CHAR(1) INIT(''),
  3 WATER_LEVEL_STORAGE_MEDIA CHAR(1) INIT(''),
  3 DISCHARGE_FREQUENCY CHAR(1) INIT(''),
  3 DISCHARGE_STORAGE_MEDIA CHAR(1) INIT(''),
  3 SUBSIDENCE_FREQ CHAR(1) INIT(''),
  3 SUBSIDENCE_STORAGE_MEDIA CHAR(1) INIT(''),
  3 GW_OTHER_DATA_AVAILABLE CHAR(12) INIT(''),
  3 MAJOR_VARIATIONS_CODE CHAR(4) INIT(''),
  3 GW_TELEMETRY_CODE PIC 'R' INIT('0'),
  3 GW_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 GW_PURPOSE_CODE CHAR(9) INIT(''),
  3 GW_RECORDER_TYPE_CODE CHAR(1) INIT(''),
  3 GW_RECORDER_FREQ_CODE CHAR(1) INIT(''),
2 GW_MODIFIERS(10),
  3 GW_POINTER CHAR(7) INIT((10)''),
  3 GW_MOD_FILE CHAR(1) INIT((10)''),
```

```
2 QUALITY_OF_WATER_DATA,
  3 QW_BEGIN_YEAR PIC '999R' INIT('0000'),
  3 QW_END_YEAR PIC '999R' INIT('0000'),
  3 QW_INTERRUPTED_CODE CHAR(1) INIT(''),
  3 QW_TELEMETRY_CODE CHAR(1) INIT(''),
  3 QW_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 QW_PURPOSE_CODE CHAR(9) INIT(''),
  3 QW_RECORDER_TYPE_CODE CHAR(1) INIT(''),
  3 QW_RECORDER_FREQ_CODE CHAR(1) INIT(''),
  3 STORET_POINTER CHAR(7) INIT(''),
2 QW_MODIFIERS(10),
  3 QW_POINTER CHAR(7) INIT((10)''),
  3 QW_MOD_FILE CHAR(1) INIT((10)''),
2 BIOLOGICAL_DATA,
  3 ENTERIC_BACTERIA CHAR(1) INIT(''),
  3 NATIVE_BACTERIA CHAR(1) INIT(''),
  3 PHYTOPLANKTON CHAR(1) INIT(''),
  3 ZOOPLANKTON CHAR(1) INIT(''),
  3 PERIPHYTON CHAR(1) INIT(''),
  3 MACROPHYTON CHAR(1) INIT(''),
  3 MICROINVERTEBRATES CHAR(1) INIT(''),
  3 MACROINVERTEBRATES CHAR(1) INIT(''),
  3 VERTEBRATES CHAR(1) INIT(''),
  3 FUNGI CHAR(1) INIT(''),
  3 VIRUSES CHAR(1) INIT(''),
  3 BIO_BEGIN_YEAR CHAR(4) INIT(''),
  3 BIO_END_YEAR CHAR(4) INIT(''),
  3 BIO_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 BIO_STORAGE_MEDIA CHAR(1) INIT(''),
  3 PRIMARY_PRODUCTIVITY CHAR(1) INIT(''),
  3 SECONDARY_PRODUCTIVITY CHAR(1) INIT(''),
  3 CHEMOSYNTHETIC_ACTIVITY CHAR(1) INIT(''),
  3 BIOSTIMULATORY_TEST CHAR(1) INIT(''),
  3 TOXICITY_TEST CHAR(1) INIT(''),
  3 OTHER_BIO_ASSAY_TEST CHAR(1) INIT(''),
  3 CHEMICAL_TISSUE_ANALYSIS CHAR(1) INIT(''),
  3 HISTOPATHIC_ANALYSIS CHAR(1) INIT(''),
  3 OTHER_TISSUE_ANALYSES CHAR(1) INIT(''),
2 BIO_MODIFIERS(10),
  3 BIO_POINTER CHAR(7) INIT((10)''),
  3 BIO_MOD_FILE CHAR(1) INIT((10)''),
2 QW_PHYSICAL_DATA,
  3 TEMPERATURE CHAR(1) INIT(''),
  3 SPECIFIC_CONDUCTANCE CHAR(1) INIT(''),
  3 TURBIDITY CHAR(1) INIT(''),
  3 COLOR CHAR(1) INIT(''),
  3 ODOR CHAR(1) INIT(''),
  3 PH CHAR(1) INIT(''),
  3 SUSPENDED_SOLIDS CHAR(1) INIT(''),
  3 PHY_BEGIN_YEAR CHAR(4) INIT(''),
  3 PHY_END_YEAR CHAR(4) INIT(''),
  3 PHY_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 PHY_STORAGE_MEDIA CHAR(1) INIT(''),
2 PHY_MODIFIERS(10),
```

```
  3 PHY_POINTER CHAR(7) INIT((10)''),
  3 PHY_MOD_FILE CHAR(1) INIT((10)''),
2 QW_SEDIMENT_DATA,
  3 BED_LOAD CHAR(1) INIT(''),
  3 SUSPENDED_CONC CHAR(1) INIT(''),
  3 TOTAL_CONC CHAR(1) INIT(''),
  3 SUSPENDED_PARTICLE_SIZE CHAR(1) INIT(''),
  3 BED_PARTICLE_SIZE CHAR(1) INIT(''),
  3 SUSPENDED_SEDIMENT_DISCH CHAR(1) INIT(''),
  3 TOTAL_SEDIMENT_DISCH CHAR(1) INIT(''),
  3 SED_BEGIN_YEAR CHAR(4) INIT(''),
  3 SED_END_YEAR CHAR(4) INIT(''),
  3 SED_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 SED_STORAGE_MEDIA CHAR(1) INIT(''),
2 SED_MODIFIERS(10),
  3 SED_POINTER CHAR(7) INIT((10)''),
  3 SED_MOD_FILE CHAR(1) INIT((10)''),
2 QW_CHEMICAL_DATA,
  3 DISSOLVED_SOLIDS CHAR(1) INIT(''),
  3 MAJOR_IONS CHAR(1) INIT(''),
  3 HARDNESS CHAR(1) INIT(''),
  3 SILICA CHAR(1) INIT(''),
  3 PHOSPHORUS CHAR(1) INIT(''),
  3 PHOSPHORUS_SPECIES CHAR(1) INIT(''),
  3 NITROGEN CHAR(1) INIT(''),
  3 NITROGEN_SPECIES CHAR(1) INIT(''),
  3 DETERGENTS CHAR(1) INIT(''),
  3 OTHER_MINOR_CONSTITUENTS CHAR(1) INIT(''),
  3 RADIOACTIVITY CHAR(1) INIT(''),
  3 RADIOCHEMICAL_SPECIES CHAR(1) INIT(''),
  3 CARBON CHAR(1) INIT(''),
  3 ORGANIC_GROUPS CHAR(1) INIT(''),
  3 PESTICIDE_SPECIES CHAR(1) INIT(''),
  3 OTHER_ORGANIC_SPECIES CHAR(1) INIT(''),
  3 BIOCHEMICAL_OXYGEN_DEMAND CHAR(1) INIT(''),
  3 CHEMICAL_OXYGEN_DEMAND CHAR(1) INIT(''),
  3 DISSOLVED_OXYGEN CHAR(1) INIT(''),
  3 OTHER_DISSOLVED_GASES CHAR(1) INIT(''),
  3 CHM_BEGIN_YEAR CHAR(4) INIT(''),
  3 CHM_END_YEAR CHAR(4) INIT(''),
  3 CHM_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 CHM_STORAGE_MEDIA CHAR(1) INIT(''),
2 CHM_MODIFIERS(10),
  3 CHM_POINTER CHAR(7) INIT((10)''),
  3 CHM_MOD_FILE CHAR(1) INIT((10)''),
2 PROJECTS(10),
  3 WRD_PROJECT_NUMBER CHAR(5) INIT((10)''),
2 NETWORKS(10),
  3 NETWORK_CODE CHAR(4) INIT((10)''),
2 OTHER_SOURCE(10),
  3 OTHER_SOURCE_AGENCY CHAR(5) INIT((10)''),
2 SOURCE_INFORMATION(10),
  3 SOURCE_FILE_IDENTIFIER CHAR(4) INIT((10)''),
  3 SOURCE_FILE_AGENCY CHAR(8) INIT((10)''),
```

```
          2 METEOROLOGICAL_DATA,
            3 MET_BEGIN_DATE CHAR(4) INIT(''),
            3 MET_END_DATE CHAR(4) INIT(''),
            3 MET_INTERRUPTED_CODE CHAR(1) INIT(''),
            3 RAINFALL CHAR(1) INIT(''),
            3 UNIT_RAINFALL CHAR(1) INIT(''),
            3 AIR_TEMPERATURE CHAR(1) INIT(''),
            3 WIND_VELOCITY CHAR(1) INIT(''),
            3 MET_OTHER_DATA_AVAILABLE CHAR(12) INIT(''),
            3 MET_TELEMETRY_CODE CHAR(1) INIT(''),
            3 MET_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
            3 MET_STORAGE_MEDIA CHAR(1) INIT(''),
            3 MET_RECORDER_TYPE_CODE CHAR(1) INIT(''),
            3 MET_RECORDER_FREQ_CODE CHAR(1) INIT(''),
          2 MET_MODIFIERS(10),
            3 MET_POINTER CHAR(7) INIT((10)''),
            3 MET_MOD_FILE CHAR(1) INIT((10)'');
DECLARE 1 MWDI_OUTPUT,
          2 SITE_DESCRIPTOR_DATA,
            3 SITE_ID FIXED BINARY(31) INIT(0),
            3 AGENCY_CODE CHAR(5) INIT(''),
            3 AGENCY_STATION_NO CHAR(15) INIT(''),
            3 LOCAL_WELL_NO CHAR(24) INIT(''),
            3 STATION_NAME CHAR(48) INIT(''),
            3 LATITUDE FIXED BINARY(31) INIT(0),
            3 LONGITUDE FIXED BINARY(31) INIT(0),
            3 LAT_LONG_ACCURACY CHAR(1) INIT(''),
            3 SITE_TYPE CHAR(2) INIT(''),
            3 COUNTRY_CODE CHAR(2) INIT(''),
            3 RESERVED1 CHAR(1) INIT(''),
            3 STATE_CODE FIXED BINARY(15) INIT(0),
            3 COUNTY_CODE FIXED BINARY(15) INIT(0),
            3 RESERVED2 CHAR(2) INIT(''),
            3 STATE_COUNTY_CODE FIXED BINARY(31) INIT(0),
            3 DISTRICT_CODE CHAR(2) INIT(''),
            3 RESERVED3 CHAR(2) INIT(''),
            3 HYDRO_UNIT_CODE FIXED BINARY(31) INIT(0),
            3 OFFICE_CODE FIXED BINARY(31) INIT(0),
            3 TOTAL_DRAINAGE_AREA FLOAT DEC(6) INIT(0),
            3 CONTRIB_DRAINAGE_AREA FLOAT DEC(6) INIT(0),
            3 ALTITUDE_DATUM FLOAT DEC(6) INIT(0),
            3 ALTITUDE_METH_OF_MEAS CHAR(1) INIT(''),
            3 RESERVED4 CHAR(3) INIT(''),
            3 ALTITUDE_ACCURACY FLOAT DEC(6) INIT(0),
            3 RIVER_REACH_NO FIXED BINARY(15) INIT(0),
            3 RIVER_REACH_SEG_FLAG FIXED BINARY(15) INIT(0),
            3 DEPTH_OF_WELL FLOAT DEC(6) INIT(0),
            3 SOURCE_WELL_DEPTH_DATA CHAR(1) INIT(''),
            3 RESERVED5 CHAR(3) INIT(''),
            3 DEPTH_OF_HOLE FLOAT DEC(6) INIT(0),
            3 PRINCIPAL_AQUIFER_CODE CHAR(8) INIT(''),
            3 AQUIFER_TYPE_CODE CHAR(1) INIT(''),
            3 CONSOLIDATED_AQUIFER_CODE CHAR(1) INIT(''),
            3 BASIN_DESCRIPTOR(3) CHAR(1) INIT((3)''),
```

```
  3 USE_OF_WATER(3) CHAR(1) INIT((3)''),
  3 USE_OF_SITE(3) CHAR(1) INIT((3)''),
  3 OTHER_DATA_AVAILABLE(6) CHAR(1) INIT((6)''),
  3 SW_ACTIVE_CODE CHAR(1) INIT(''),
  3 GW_ACTIVE_CODE CHAR(1) INIT(''),
  3 QW_ACTIVE_CODE CHAR(1) INIT(''),
  3 BIO_ACTIVE_CODE CHAR(1) INIT(''),
  3 PHY_ACTIVE_CODE CHAR(1) INIT(''),
  3 SED_ACTIVE_CODE CHAR(1) INIT(''),
  3 CHM_ACTIVE_CODE CHAR(1) INIT(''),
  3 MET_ACTIVE_CODE CHAR(1) INIT(''),
  3 RESERVED6 CHAR(3) INIT(''),
  3 DATE_OF_LAST_UPDATE FIXED BINARY(31) INIT(0),
2 NODE_LOCATION_INFO,
  3 DIS_NODE_IDENT CHAR(6) INIT(''),
  3 SW_DATA_CODE CHAR(1) INIT(''),
  3 GW_DATA_CODE CHAR(1) INIT(''),
  3 QW_DATA_CODE CHAR(1) INIT(''),
  3 MET_DATA_CODE CHAR(1) INIT(''),
  3 OTHER_DATA_CODE CHAR(1) INIT(''),
  3 RESERVED7 CHAR(1) INIT(''),
  3 NODE_DATE_OF_LAST_UPDATE FIXED BINARY(31) INIT(0),
2 SURFACE_WATER_DATA,
  3 SW_BEGIN_YEAR PIC '999R' INIT('0000'),
  3 SW_END_YEAR PIC '999R' INIT('0000'),
  3 SW_INTERRUPTED_CODE CHAR(1) INIT(''),
  3 COMPLETE_STAGE CHAR(1) INIT(''),
  3 PEAK_STAGE CHAR(1) INIT(''),
  3 LOW_STAGE CHAR(1) INIT(''),
  3 STAGE_STORAGE_MEDIA CHAR(1) INIT(''),
  3 COMPLETE_FLOW CHAR(1) INIT(''),
  3 PEAK_FLOW CHAR(1) INIT(''),
  3 LOW_FLOW CHAR(1) INIT(''),
  3 MISC_FLOW_MEAS CHAR(1) INIT(''),
  3 FLOW_STORAGE_MEDIA CHAR(1) INIT(''),
  3 VOLUME CHAR(1) INIT(''),
  3 VOLUME_CHANGE CHAR(1) INIT(''),
  3 VOLUME_STORAGE_MEDIA CHAR(1) INIT(''),
  3 STAGE_TYPE CHAR(2) INIT(''),
  3 STAGE_FREQ_CODE CHAR(1) INIT(''),
  3 FLOW_TYPE CHAR(2) INIT(''),
  3 FLOW_FREQ_CODE CHAR(1) INIT(''),
  3 VOLUME_TYPE CHAR(2) INIT(''),
  3 VOLUME_FREQ_CODE CHAR(1) INIT(''),
  3 VELOCITY_TYPE CHAR(2) INIT(''),
  3 VELOCITY_FREQ_CODE CHAR(1) INIT(''),
  3 BASIN_CHARAC_DATA_AVAIL_CODE CHAR(1) INIT(''),
  3 SW_DATA_STORAGE_MEDIA CHAR(1) INIT(''),
  3 SW_TELEMETRY_CODE CHAR(1) INIT(''),
  3 SW_RECORDER_TYPE_CODE CHAR(1) INIT(''),
  3 SW_DATA_PURPOSE_CODE(4) CHAR(1) INIT((4)''),
  3 OTHER_SW_RELATED_DATA(12) CHAR(1) INIT((12)' '),
  3 SW_SEASONAL_CODE CHAR(1) INIT(''),
  3 RESERVED8 CHAR(2) INIT(''),
```

```
  3 SW_DATE_OF_LAST_UPDATE FIXED BINARY(31) INIT(0),
2 SW_DATA_BASE_IDENTIFIERS(4),
  3 SW_DATA_BASE_IDENTIFIER CHAR(4) INIT((4)''),
  3 SW_DATA_BASE_SUBSET_IDENT CHAR(4) INIT((4)''),
  3 SW_DB_DATE_OF_LAST_UPDATE FIXED BINARY(31) INIT(0),
2 GROUND_WATER_DATA,
  3 GW_BEGIN_YEAR PIC '999R' INIT('0000'),
  3 GW_END_YEAR PIC '999R' INIT('0000'),
  3 GW_INTERRUPTED_CODE CHAR(1) INIT(''),
  3 WATER_LEVEL_FREQ CHAR(1) INIT(''),
  3 WATER_LEVEL_STORAGE_MEDIA CHAR(1) INIT(''),
  3 DISCHARGE_FREQUENCY CHAR(1) INIT(''),
  3 DISCHARGE_STORAGE_MEDIA CHAR(1) INIT(''),
  3 SUBSIDENCE_FREQ CHAR(1) INIT(''),
  3 SUBSIDENCE_STORAGE_MEDIA CHAR(1) INIT(''),
  3 GW_OTHER_DATA_AVAILABLE CHAR(12) INIT(''),
  3 MAJOR_VARIATIONS_CODE CHAR(4) INIT(''),
  3 GW_TELEMETRY_CODE PIC 'R' INIT('0'),
  3 GW_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 GW_PURPOSE_CODE CHAR(9) INIT(''),
  3 GW_RECORDER_TYPE_CODE CHAR(1) INIT(''),
  3 GW_RECORDER_FREQ_CODE CHAR(1) INIT(''),
2 GW_MODIFIERS(10),
  3 GW_POINTER CHAR(7) INIT((10)''),
  3 GW_MOD_FILE CHAR(1) INIT((10)''),
2 QUALITY_OF_WATER_DATA,
  3 QW_BEGIN_YEAR PIC '999R' INIT('0000'),
  3 QW_END_YEAR PIC '999R' INIT('0000'),
  3 QW_INTERRUPTED_CODE CHAR(1) INIT(''),
  3 QW_TELEMETRY_CODE CHAR(1) INIT(''),
  3 QW_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 QW_PURPOSE_CODE CHAR(9) INIT(''),
  3 QW_RECORDER_TYPE_CODE CHAR(1) INIT(''),
  3 QW_RECORDER_FREQ_CODE CHAR(1) INIT(''),
  3 STORET_POINTER CHAR(7) INIT(''),
2 QW_MODIFIERS(10),
  3 QW_POINTER CHAR(7) INIT((10)''),
  3 QW_MOD_FILE CHAR(1) INIT((10)''),
2 BIOLOGICAL_DATA,
  3 ENTERIC_BACTERIA CHAR(1) INIT(''),
  3 NATIVE_BACTERIA CHAR(1) INIT(''),
  3 PHYTOPLANKTON CHAR(1) INIT(''),
  3 ZOOPLANKTON CHAR(1) INIT(''),
  3 PERIPHYTON CHAR(1) INIT(''),
  3 MACROPHYTON CHAR(1) INIT(''),
  3 MICROINVERTEBRATES CHAR(1) INIT(''),
  3 MACROINVERTEBRATES CHAR(1) INIT(''),
  3 VERTEBRATES CHAR(1) INIT(''),
  3 FUNGI CHAR(1) INIT(''),
  3 VIRUSES CHAR(1) INIT(''),
  3 BIO_BEGIN_YEAR CHAR(4) INIT(''),
  3 BIO_END_YEAR CHAR(4) INIT(''),
  3 BIO_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 BIO_STORAGE_MEDIA CHAR(1) INIT(''),
```

```
  3 PRIMARY_PRODUCTIVITY CHAR(1) INIT(''),
  3 SECONDARY_PRODUCTIVITY CHAR(1) INIT(''),
  3 CHEMOSYNTHETIC_ACTIVITY CHAR(1) INIT(''),
  3 BIOSTIMULATORY_TEST CHAR(1) INIT(''),
  3 TOXICITY_TEST CHAR(1) INIT(''),
  3 OTHER_BIO_ASSAY_TEST CHAR(1) INIT(''),
  3 CHEMICAL_TISSUE_ANALYSIS CHAR(1) INIT(''),
  3 HISTOPATHIC_ANALYSIS CHAR(1) INIT(''),
  3 OTHER_TISSUE_ANALYSES CHAR(1) INIT(''),
2 BIO_MODIFIERS(10),
  3 BIO_POINTER CHAR(7) INIT((10)''),
  3 BIO_MOD_FILE CHAR(1) INIT((10)''),
2 QW_PHYSICAL_DATA,
  3 TEMPERATURE CHAR(1) INIT(''),
  3 SPECIFIC_CONDUCTANCE CHAR(1) INIT(''),
  3 TURBIDITY CHAR(1) INIT(''),
  3 COLOR CHAR(1) INIT(''),
  3 ODOR CHAR(1) INIT(''),
  3 PH CHAR(1) INIT(''),
  3 SUSPENDED_SOLIDS CHAR(1) INIT(''),
  3 PHY_BEGIN_YEAR CHAR(4) INIT(''),
  3 PHY_END_YEAR CHAR(4) INIT(''),
  3 PHY_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 PHY_STORAGE_MEDIA CHAR(1) INIT(''),
2 PHY_MODIFIERS(10),
  3 PHY_POINTER CHAR(7) INIT((10)''),
  3 PHY_MOD_FILE CHAR(1) INIT((10)''),
2 QW_SEDIMENT_DATA,
  3 BED_LOAD CHAR(1) INIT(''),
  3 SUSPENDED_CONC CHAR(1) INIT(''),
  3 TOTAL_CONC CHAR(1) INIT(''),
  3 SUSPENDED_PARTICLE_SIZE CHAR(1) INIT(''),
  3 BED_PARTICLE_SIZE CHAR(1) INIT(''),
  3 SUSPENDED_SEDIMENT_DISCH CHAR(1) INIT(''),
  3 TOTAL_SEDIMENT_DISCH CHAR(1) INIT(''),
  3 SED_BEGIN_YEAR CHAR(4) INIT(''),
  3 SED_END_YEAR CHAR(4) INIT(''),
  3 SED_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 SED_STORAGE_MEDIA CHAR(1) INIT(''),
2 SED_MODIFIERS(10),
  3 SED_POINTER CHAR(7) INIT((10)''),
  3 SED_MOD_FILE CHAR(1) INIT((10)''),
2 QW_CHEMICAL_DATA,
  3 DISSOLVED_SOLIDS CHAR(1) INIT(''),
  3 MAJOR_IONS CHAR(1) INIT(''),
  3 HARDNESS CHAR(1) INIT(''),
  3 SILICA CHAR(1) INIT(''),
  3 PHOSPHORUS CHAR(1) INIT(''),
  3 PHOSPHORUS_SPECIES CHAR(1) INIT(''),
  3 NITROGEN CHAR(1) INIT(''),
  3 NITROGEN_SPECIES CHAR(1) INIT(''),
  3 DETERGENTS CHAR(1) INIT(''),
  3 OTHER_MINOR_CONSTITUENTS CHAR(1) INIT(''),
  3 RADIOACTIVITY CHAR(1) INIT(''),
```

```
    3 RADIOCHEMICAL_SPECIES CHAR(1) INIT(''),
    3 CARBON CHAR(1) INIT(''),
    3 ORGANIC_GROUPS CHAR(1) INIT(''),
    3 PESTICIDE_SPECIES CHAR(1) INIT(''),
    3 OTHER_ORGANIC_SPECIES CHAR(1) INIT(''),
    3 BIOCHEMICAL_OXYGEN_DEMAND CHAR(1) INIT(''),
    3 CHEMICAL_OXYGEN_DEMAND CHAR(1) INIT(''),
    3 DISSOLVED_OXYGEN CHAR(1) INIT(''),
    3 OTHER_DISSOLVED_GASES CHAR(1) INIT(''),
    3 CHM_BEGIN_YEAR CHAR(4) INIT(''),
    3 CHM_END_YEAR CHAR(4) INIT(''),
    3 CHM_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
    3 CHM_STORAGE_MEDIA CHAR(1) INIT(''),
  2 CHM_MODIFIERS(10),
    3 CHM_POINTER CHAR(7) INIT((10)''),
    3 CHM_MOD_FILE CHAR(1) INIT((10)''),
  2 PROJECTS(10),
    3 WRD_PROJECT_NUMBER CHAR(5) INIT((10)''),
  2 NETWORKS(10),
    3 NETWORK_CODE CHAR(4) INIT((10)''),
  2 OTHER_SOURCE(10),
    3 OTHER_SOURCE_AGENCY CHAR(5) INIT((10)''),
  2 SOURCE_INFORMATION(10),
    3 SOURCE_FILE_IDENTIFIER CHAR(4) INIT((10)''),
    3 SOURCE_FILE_AGENCY CHAR(8) INIT((10)''),
  2 METEOROLOGICAL_DATA,
    3 MET_BEGIN_DATE CHAR(4) INIT(''),
    3 MET_END_DATE CHAR(4) INIT(''),
    3 MET_INTERRUPTED_CODE CHAR(1) INIT(''),
    3 RAINFALL CHAR(1) INIT(''),
    3 UNIT_RAINFALL CHAR(1) INIT(''),
    3 AIR_TEMPERATURE CHAR(1) INIT(''),
    3 WIND_VELOCITY CHAR(1) INIT(''),
    3 MET_OTHER_DATA_AVAILABLE CHAR(12) INIT(''),
    3 MET_TELEMETRY_CODE CHAR(1) INIT(''),
    3 MET_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
    3 MET_STORAGE_MEDIA CHAR(1) INIT(''),
    3 MET_RECORDER_TYPE_CODE CHAR(1) INIT(''),
    3 MET_RECORDER_FREQ_CODE CHAR(1) INIT(''),
  2 MET_MODIFIERS(10),
    3 MET_POINTER CHAR(7) INIT((10)''),
    3 MET_MOD_FILE CHAR(1) INIT((10)'');
DECLARE 1 MWDI_TEMPLATE,
  2 SITE_DESCRIPTOR_DATA,
    3 SITE_ID FIXED BINARY(31) INIT(0),
    3 AGENCY_CODE CHAR(5) INIT(''),
    3 AGENCY_STATION_NO CHAR(15) INIT(''),
    3 LOCAL_WELL_NO CHAR(24) INIT(''),
    3 STATION_NAME CHAR(48) INIT(''),
    3 LATITUDE FIXED BINARY(31) INIT(0),
    3 LONGITUDE FIXED BINARY(31) INIT(0),
    3 LAT_LONG_ACCURACY CHAR(1) INIT(''),
    3 SITE_TYPE CHAR(2) INIT(''),
    3 COUNTRY_CODE CHAR(2) INIT(''),
```

```
  3 RESERVED1 CHAR(1) INIT(''),
  3 STATE_CODE FIXED BINARY(15) INIT(''),
  3 COUNTY_CODE FIXED BINARY(15) INIT(''),
  3 RESERVED2 CHAR(2) INIT(''),
  3 STATE_COUNTY_CODE FIXED BINARY(31) INIT(0),
  3 DISTRICT_CODE CHAR(2) INIT(''),
  3 RESERVED3 CHAR(2) INIT(''),
  3 HYDRO_UNIT_CODE FIXED BINARY(31) INIT(0),
  3 OFFICE_CODE FIXED BINARY(31) INIT(0),
  3 TOTAL_DRAINAGE_AREA FLOAT DEC(6) INIT(0),
  3 CONTRIB_DRAINAGE_AREA FLOAT DEC(6) INIT(0),
  3 ALTITUDE_DATUM FLOAT DEC(6) INIT(0),
  3 ALTITUDE_METH_OF_MEAS CHAR(1) INIT(''),
  3 RESERVED4 CHAR(3) INIT(''),
  3 ALTITUDE_ACCURACY FLOAT DEC(6) INIT(0),
  3 RIVER_REACH_NO FIXED BINARY(15) INIT(0),
  3 RIVER_REACH_SEG_FLAG FIXED BINARY(15) INIT(0),
  3 DEPTH_OF_WELL FLOAT DEC(6) INIT(0),
  3 SOURCE_WELL_DEPTH_DATA CHAR(1) INIT(''),
  3 RESERVED5 CHAR(3) INIT(''),
  3 DEPTH_OF_HOLE FLOAT DEC(6) INIT(0),
  3 PRINCIPAL_AQUIFER_CODE CHAR(8) INIT(''),
  3 AQUIFER_TYPE_CODE CHAR(1) INIT(''),
  3 CONSOLIDATED_AQUIFER_CODE CHAR(1) INIT(''),
  3 BASIN_DESCRIPTOR(3) CHAR(1) INIT((3)''),
  3 USE_OF_WATER(3) CHAR(1) INIT((3)''),
  3 USE_OF_SITE(3) CHAR(1) INIT((3)''),
  3 OTHER_DATA_AVAILABLE(6) CHAR(1) INIT((6)''),
  3 SW_ACTIVE_CODE CHAR(1) INIT(''),
  3 GW_ACTIVE_CODE CHAR(1) INIT(''),
  3 QW_ACTIVE_CODE CHAR(1) INIT(''),
  3 BIO_ACTIVE_CODE CHAR(1) INIT(''),
  3 PHY_ACTIVE_CODE CHAR(1) INIT(''),
  3 SED_ACTIVE_CODE CHAR(1) INIT(''),
  3 CHM_ACTIVE_CODE CHAR(1) INIT(''),
  3 MET_ACTIVE_CODE CHAR(1) INIT(''),
  3 RESERVED6 CHAR(3) INIT(''),
  3 DATE_OF_LAST_UPDATE FIXED BINARY(31) INIT(0),
2 NODE_LOCATION_INFO,
  3 DIS_NODE_IDENT CHAR(6) INIT(''),
  3 SW_DATA_CODE CHAR(1) INIT(''),
  3 GW_DATA_CODE CHAR(1) INIT(''),
  3 QW_DATA_CODE CHAR(1) INIT(''),
  3 MET_DATA_CODE CHAR(1) INIT(''),
  3 OTHER_DATA_CODE CHAR(1) INIT(''),
  3 RESERVED7 CHAR(1) INIT(''),
  3 NODE_DATE_OF_LAST_UPDATE FIXED BINARY(31) INIT(0),
2 SURFACE_WATER_DATA,
  3 SW_BEGIN_YEAR PIC '999R' INIT('0000'),
  3 SW_END_YEAR PIC '999R' INIT('0000'),
  3 SW_INTERRUPTED_CODE CHAR(1) INIT(''),
  3 COMPLETE_STAGE CHAR(1) INIT(''),
  3 PEAK_STAGE CHAR(1) INIT(''),
  3 LOW_STAGE CHAR(1) INIT(''),
```

```
  3 STAGE_STORAGE_MEDIA CHAR(1) INIT(''),
  3 COMPLETE_FLOW CHAR(1) INIT(''),
  3 PEAK_FLOW CHAR(1) INIT(''),
  3 LOW_FLOW CHAR(1) INIT(''),
  3 MISC_FLOW_MEAS CHAR(1) INIT(''),
  3 FLOW_STORAGE_MEDIA CHAR(1) INIT(''),
  3 VOLUME CHAR(1) INIT(''),
  3 VOLUME_CHANGE CHAR(1) INIT(''),
  3 VOLUME_STORAGE_MEDIA CHAR(1) INIT(''),
  3 STAGE_TYPE CHAR(2) INIT(''),
  3 STAGE_FREQ_CODE CHAR(1) INIT(''),
  3 FLOW_TYPE CHAR(2) INIT(''),
  3 FLOW_FREQ_CODE CHAR(1) INIT(''),
  3 VOLUME_TYPE CHAR(2) INIT(''),
  3 VOLUME_FREQ_CODE CHAR(1) INIT(''),
  3 VELOCITY_TYPE CHAR(2) INIT(''),
  3 VELOCITY_FREQ_CODE CHAR(1) INIT(''),
  3 BASIN_CHARAC_DATA_AVAIL_CODE CHAR(1) INIT(''),
  3 SW_DATA_STORAGE_MEDIA CHAR(1) INIT(''),
  3 SW_TELEMETRY_CODE CHAR(1) INIT(''),
  3 SW_RECORDER_TYPE_CODE CHAR(1) INIT(''),
  3 SW_DATA_PURPOSE_CODE(4) CHAR(1) INIT((4)''),
  3 OTHER_SW_RELATED_DATA(12) CHAR(1) INIT((12)' '),
  3 SW_SEASONAL_CODE CHAR(1) INIT(''),
  3 RESERVED8 CHAR(2) INIT(''),
  3 SW_DATE_OF_LAST_UPDATE FIXED BINARY(31) INIT(0),
2 SW_DATA_BASE_IDENTIFIERS(4),
  3 SW_DATA_BASE_IDENTIFIER CHAR(4) INIT((4)''),
  3 SW_DATA_BASE_SUBSET_IDENT CHAR(4) INIT((4)''),
  3 SW_DB_DATE_OF_LAST_UPDATE FIXED BINARY(31) INIT(0),
2 GROUND_WATER_DATA,
  3 GW_BEGIN_YEAR PIC '999R' INIT('0000'),
  3 GW_END_YEAR PIC '999R' INIT('0000'),
  3 GW_INTERRUPTED_CODE CHAR(1) INIT(''),
  3 WATER_LEVEL_FREQ CHAR(1) INIT(''),
  3 WATER_LEVEL_STORAGE_MEDIA CHAR(1) INIT(''),
  3 DISCHARGE_FREQUENCY CHAR(1) INIT(''),
  3 DISCHARGE_STORAGE_MEDIA CHAR(1) INIT(''),
  3 SUBSIDENCE_FREQ CHAR(1) INIT(''),
  3 SUBSIDENCE_STORAGE_MEDIA CHAR(1) INIT(''),
  3 GW_OTHER_DATA_AVAILABLE CHAR(12) INIT(''),
  3 MAJOR_VARIATIONS_CODE CHAR(4) INIT(''),
  3 GW_TELEMETRY_CODE PIC 'R' INIT('0'),
  3 GW_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 GW_PURPOSE_CODE CHAR(9) INIT(''),
  3 GW_RECORDER_TYPE_CODE CHAR(1) INIT(''),
  3 GW_RECORDER_FREQ_CODE CHAR(1) INIT(''),
2 GW_MODIFIERS(10),
  3 GW_POINTER CHAR(7) INIT((10)''),
  3 GW_MOD_FILE CHAR(1) INIT((10)''),
2 QUALITY_OF_WATER_DATA,
  3 QW_BEGIN_YEAR PIC '999R' INIT('0000'),
  3 QW_END_YEAR PIC '999R' INIT('0000'),
  3 QW_INTERRUPTED_CODE CHAR(1) INIT(''),
```

```
  3 QW_TELEMETRY_CODE CHAR(1) INIT(''),
  3 QW_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 QW_PURPOSE_CODE CHAR(9) INIT(''),
  3 QW_RECORDER_TYPE_CODE CHAR(1) INIT(''),
  3 QW_RECORDER_FREQ_CODE CHAR(1) INIT(''),
  3 STORET_POINTER CHAR(7) INIT(''),
2 QW_MODIFIERS(10),
  3 QW_POINTER CHAR(7) INIT((10)''),
  3 QW_MOD_FILE CHAR(1) INIT((10)''),
2 BIOLOGICAL_DATA,
  3 ENTERIC_BACTERIA CHAR(1) INIT(''),
  3 NATIVE_BACTERIA CHAR(1) INIT(''),
  3 PHYTOPLANKTON CHAR(1) INIT(''),
  3 ZOOPLANKTON CHAR(1) INIT(''),
  3 PERIPHYTON CHAR(1) INIT(''),
  3 MACROPHYTON CHAR(1) INIT(''),
  3 MICROINVERTEBRATES CHAR(1) INIT(''),
  3 MACROINVERTEBRATES CHAR(1) INIT(''),
  3 VERTEBRATES CHAR(1) INIT(''),
  3 FUNGI CHAR(1) INIT(''),
  3 VIRUSES CHAR(1) INIT(''),
  3 BIO_BEGIN_YEAR CHAR(4) INIT(''),
  3 BIO_END_YEAR CHAR(4) INIT(''),
  3 BIO_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 BIO_STORAGE_MEDIA CHAR(1) INIT(''),
  3 PRIMARY_PRODUCTIVITY CHAR(1) INIT(''),
  3 SECONDARY_PRODUCTIVITY CHAR(1) INIT(''),
  3 CHEMOSYNTHETIC_ACTIVITY CHAR(1) INIT(''),
  3 BIOSTIMULATORY_TEST CHAR(1) INIT(''),
  3 TOXICITY_TEST CHAR(1) INIT(''),
  3 OTHER_BIO_ASSAY_TEST CHAR(1) INIT(''),
  3 CHEMICAL_TISSUE_ANALYSIS CHAR(1) INIT(''),
  3 HISTOPATHIC_ANALYSIS CHAR(1) INIT(''),
  3 OTHER_TISSUE_ANALYSES CHAR(1) INIT(''),
2 BIO_MODIFIERS(10),
  3 BIO_POINTER CHAR(7) INIT((10)''),
  3 BIO_MOD_FILE CHAR(1) INIT((10)''),
2 QW_PHYSICAL_DATA,
  3 TEMPERATURE CHAR(1) INIT(''),
  3 SPECIFIC_CONDUCTANCE CHAR(1) INIT(''),
  3 TURBIDITY CHAR(1) INIT(''),
  3 COLOR CHAR(1) INIT(''),
  3 ODOR CHAR(1) INIT(''),
  3 PH CHAR(1) INIT(''),
  3 SUSPENDED_SOLIDS CHAR(1) INIT(''),
  3 PHY_BEGIN_YEAR CHAR(4) INIT(''),
  3 PHY_END_YEAR CHAR(4) INIT(''),
  3 PHY_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
  3 PHY_STORAGE_MEDIA CHAR(1) INIT(''),
2 PHY_MODIFIERS(10),
  3 PHY_POINTER CHAR(7) INIT((10)''),
  3 PHY_MOD_FILE CHAR(1) INIT((10)''),
2 QW_SEDIMENT_DATA,
  3 BED_LOAD CHAR(1) INIT(''),
```

```
   3 SUSPENDED_CONC CHAR(1) INIT(''),
   3 TOTAL_CONC CHAR(1) INIT(''),
   3 SUSPENDED_PARTICLE_SIZE CHAR(1) INIT(''),
   3 BED_PARTICLE_SIZE CHAR(1) INIT(''),
   3 SUSPENDED_SEDIMENT_DISCH CHAR(1) INIT(''),
   3 TOTAL_SEDIMENT_DISCH CHAR(1) INIT(''),
   3 SED_BEGIN_YEAR CHAR(4) INIT(''),
   3 SED_END_YEAR CHAR(4) INIT(''),
   3 SED_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
   3 SED_STORAGE_MEDIA CHAR(1) INIT(''),
 2 SED_MODIFIERS(10),
   3 SED_POINTER CHAR(7) INIT((10)''),
   3 SED_MOD_FILE CHAR(1) INIT((10)''),
 2 QW_CHEMICAL_DATA,
   3 DISSOLVED_SOLIDS CHAR(1) INIT(''),
   3 MAJOR_IONS CHAR(1) INIT(''),
   3 HARDNESS CHAR(1) INIT(''),
   3 SILICA CHAR(1) INIT(''),
   3 PHOSPHORUS CHAR(1) INIT(''),
   3 PHOSPHORUS_SPECIES CHAR(1) INIT(''),
   3 NITROGEN CHAR(1) INIT(''),
   3 NITROGEN_SPECIES CHAR(1) INIT(''),
   3 DETERGENTS CHAR(1) INIT(''),
   3 OTHER_MINOR_CONSTITUENTS CHAR(1) INIT(''),
   3 RADIOACTIVITY CHAR(1) INIT(''),
   3 RADIOCHEMICAL_SPECIES CHAR(1) INIT(''),
   3 CARBON CHAR(1) INIT(''),
   3 ORGANIC_GROUPS CHAR(1) INIT(''),
   3 PESTICIDE_SPECIES CHAR(1) INIT(''),
   3 OTHER_ORGANIC_SPECIES CHAR(1) INIT(''),
   3 BIOCHEMICAL_OXYGEN_DEMAND CHAR(1) INIT(''),
   3 CHEMICAL_OXYGEN_DEMAND CHAR(1) INIT(''),
   3 DISSOLVED_OXYGEN CHAR(1) INIT(''),
   3 OTHER_DISSOLVED_GASES CHAR(1) INIT(''),
   3 CHM_BEGIN_YEAR CHAR(4) INIT(''),
   3 CHM_END_YEAR CHAR(4) INIT(''),
   3 CHM_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
   3 CHM_STORAGE_MEDIA CHAR(1) INIT(''),
 2 CHM_MODIFIERS(10),
   3 CHM_POINTER CHAR(7) INIT((10)''),
   3 CHM_MOD_FILE CHAR(1) INIT((10)''),
 2 PROJECTS(10),
   3 WRD_PROJECT_NUMBER CHAR(5) INIT((10)''),
 2 NETWORKS(10),
   3 NETWORK_CODE CHAR(4) INIT((10)''),
 2 OTHER_SOURCE(10),
   3 OTHER_SOURCE_AGENCY CHAR(5) INIT((10)''),
 2 SOURCE_INFORMATION(10),
   3 SOURCE_FILE_IDENTIFIER CHAR(4) INIT((10)''),
   3 SOURCE_FILE_AGENCY CHAR(8) INIT((10)''),
 2 METEOROLOGICAL_DATA,
   3 MET_BEGIN_DATE CHAR(4) INIT(''),
   3 MET_END_DATE CHAR(4) INIT(''),
   3 MET_INTERRUPTED_CODE CHAR(1) INIT(''),
```

```
            3 RAINFALL CHAR(1) INIT(''),
            3 UNIT_RAINFALL CHAR(1) INIT(''),
            3 AIR_TEMPERATURE CHAR(1) INIT(''),
            3 WIND_VELOCITY CHAR(1) INIT(''),
            3 MET_OTHER_DATA_AVAILABLE CHAR(12) INIT(''),
            3 MET_TELEMETRY_CODE CHAR(1) INIT(''),
            3 MET_DATE_OF_LAST_UPDATE CHAR(4) INIT(''),
            3 MET_STORAGE_MEDIA CHAR(1) INIT(''),
            3 MET_RECORDER_TYPE_CODE CHAR(1) INIT(''),
            3 MET_RECORDER_FREQ_CODE CHAR(1) INIT(''),
          2 MET_MODIFIERS(10),
            3 MET_POINTER CHAR(7) INIT((10)''),
            3 MET_MOD_FILE CHAR(1) INIT((10)'');
DECLARE 1 WATSTORE_UV_RECORD,
          2 RESERVED_SPACE CHAR(1) INIT(''),
          2 RECORD_FORMAT CHAR(1) INIT(''),
          2 STATE_CODE CHAR(2) INIT(''),
          2 AGENCY_CODE CHAR(5) INIT(''),
          2 STATION_IDENTIFIER CHAR(15) INIT(''),
          2 CROSS_SECTION_LOCATOR FLOAT DEC(6) INIT(0),
          2 DEPTH_LOCATOR FLOAT DEC(6) INIT(0),
          2 PARAMETER_CODE FIXED BINARY(31) INIT(0),
          2 YEAR_NUMBER FIXED BINARY(15) INIT(0),
          2 MONTH_NUMBER FIXED BINARY(15) INIT(0),
          2 DAY_NUMBER FIXED BINARY(15) INIT(0),
          2 STATISTIC_CODE FIXED BINARY(15) INIT(0),
          2 READINGS_PER_DAY FIXED BINARY(15) INIT(0),
          2 FIRST_READING_POSITION FIXED BINARY(15) INIT(0),
          2 NUMBER_OF_READINGS FIXED BINARY(15) INIT(0),
          2 RETRIEVAL_SEQUENCE_NUMBER FIXED BINARY(15) INIT(0),
          2 MORE_RESERVED_SPACE CHAR(9) INIT(''),
          2 SITE_CODE CHAR(2) INIT(''),
          2 DISTRICT_CODE CHAR(2) INIT(''),
          2 COUNTY_CODE CHAR(3) INIT(''),
          2 CREATE_DATE FIXED BINARY(31) INIT(0),
          2 EVEN_MORE_RESERVED_SPACE CHAR(4) INIT(''),
          2 MISSING_VALUE_INDICATOR FLOAT DEC(6) INIT(0),
          2 UNIT_VALUES(2880) FLOAT DEC(6) INIT((2880)0);
DECLARE 1 UNIQUE_SITE_IDENTIFIER,
          2 DEL CHAR(1) INIT(''),
          2 STATE_CODE CHAR(2) INIT(''),
          2 RESERVED CHAR(1) INIT(''),
          2 NEXT_AVAIL_UNIQUE_SITE_ID FIXED BINARY(31) INIT(0);
DECLARE 1 SW_FREQ_HISTORY_OUTPUT,
          2 UNIQUE_SITE_ID FIXED BINARY(31) INIT(0),
          2 AGENCY_CODE CHAR(5) INIT(''),
          2 AGENCY_STATION_NUMBER CHAR(15) INIT(''),
          2 SW_YEAR_OF_MEASUREMENT FIXED BIN(31) INIT(0),
          2 STAGE_TYPE_HISTORY CHAR(2) INIT(''),
          2 STAGE_FREQ_HISTORY CHAR(1) INIT(''),
          2 FLOW_TYPE_HISTORY CHAR(2) INIT(''),
          2 FLOW_FREQ_HISTORY CHAR(1) INIT(''),
          2 VOLUME_TYPE_HISTORY CHAR(2) INIT(''),
          2 VOLUME_FREQ_HISTORY CHAR(1) INIT(''),
```

```
          2 VELOCITY_TYPE_HISTORY CHAR(2) INIT(''),
          2 VELOCITY_FREQ_HISTORY CHAR(1) INIT(''),
          2 SW_HISTORY_DATE_OF_LAST_UPDATE FIXED BINARY(31) INIT(0);
DECLARE 1 GW_QW_MET_TRANSACTION_OUTPUT,
          2 UNIQUE_SITE_ID FIXED BINARY(31) INIT(0),
          2 AGENCY_CODE CHAR(5) INIT(''),
          2 AGENCY_STATION_NUMBER CHAR(15) INIT(''),
          2 PARAMETER_CODE FIXED BINARY(31) INIT(0),
          2 YEAR_OF_MEASUREMENT FIXED BINARY(15) INIT(0),
          2 FREQUENCY_CODE CHAR(1) INIT(''),
          2 SEASONAL_CODE CHAR(1) INIT(''),
          2 DATA_BASE_IDENTIFIER CHAR(4) INIT(''),
          2 DATA_BASE_SUBSET_IDENTIFIER CHAR(4) INIT('');
DECLARE (MWDI_NOT_EOF,
         WATSTORE_NOT_EOF,
         FIRST_TIME_THROUGH,
         NOT_COUNTED,
         ITS_SW_DATA INIT('0'B),
         ITS_SW_DATA2 INIT('0'B),
         NEW_RECORD INIT('1'B),
         OTHER_NEW_RECORD INIT('1'B),
         LAST_PASS INIT('0'B),
         OTHER_RECORD_WRITTEN,
         FIRST_SW_VAL INIT('1'B)) BIT(1);
DECLARE OLDMWDI FILE;
DECLARE WATSTOR FILE;
DECLARE NEWMWDI FILE;
DECLARE SWFRQHT RECORD SEQUENTIAL FILE;
DECLARE OTRTRNS RECORD SEQUENTIAL FILE;
DECLARE SYSPRINT FILE;
DECLARE NXTAVAL UPDATE RECORD KEYED ENV(INDEXED) FILE;
DECLARE CURRENT_STATE FIXED BINARY(15) INIT(0);
DECLARE PREVIOUS_STATE FIXED BINARY(15) INIT(0);
DECLARE PREVIOUS_STATION_ID CHAR(15) INIT('');
DECLARE VALIDATE_ID CHAR(15) INIT('');
DECLARE CONDITION_FLAG FIXED DEC(1) INIT(0);
DECLARE (PER_STATE_TOTAL_SITES_IN,
         PER_STATE_TOTAL_SITES_OUT,
         PER_STATE_TOTAL_SITES_ADDED,
         TOTAL_SITES_IN,
         TOTAL_SITES_OUT,
         TOTAL_SITES_ADDED) FIXED DEC(7) INIT(0);
DECLARE PREV_ID CHAR(15);
DECLARE (SW_DAY,
          OTHER_DAY,
          CURRENT_PARAMETER_CODE,
          NUMSTATES) FIXED DEC INIT(0);
DECLARE (SW_YEAR,
          OTHER_YEAR,
          NUM_DAYS_THIS_PARM) FIXED BIN(15) INIT(0);
DECLARE SW_VALIDATE_PARM FIXED BIN(31);
DECLARE SW_VALIDATE_YEAR FIXED BIN(15);
DECLARE (PER_STATE_SITES_WITH_FREQ_HIST,
          PER_STATE_NUM_FREQ_HIST_RECORDS,
```

```
         SITES_WITH_FREQ_HIST,
         NUM_FREQ_HIST_RECORDS) FIXED DEC(7) INIT(0);
DECLARE (PER_STATE_GW_TRANSACTIONS,
         PER_STATE_QW_TRANSACTIONS,
         PER_STATE_MET_TRANSACTIONS,
         PER_STATE_TOTAL_TRANSACTIONS,
         GW_TRANSACTIONS,
         QW_TRANSACTIONS,
         MET_TRANSACTIONS,
         TOTAL_TRANSACTIONS) FIXED DEC(7) INIT(0);
DECLARE 1 PER_STATE_DATA_ELEMENT_COUNTS,
          2 INPUT,
            3 UNIQUE_SITE_IDENTIFIER FIXED DEC(7) INIT(0),
            3 AGENCY_STATION_NUMBER FIXED DEC(7) INIT(0),
            3 SW_ACTIVE_CODE FIXED DEC(7) INIT(0),
            3 SW_BEGIN_YEAR FIXED DEC(7) INIT(0),
            3 SW_END_YEAR FIXED DEC(7) INIT(0),
            3 SW_INTERRUPTED_CODE FIXED DEC(7) INIT(0),
            3 UNIT_FLOWS FIXED DEC(7) INIT(0),
            3 UNIT_STAGE FIXED DEC(7) INIT(0),
            3 UNIT_VOLUME FIXED DEC(7) INIT(0),
            3 SW_OTHER_DATA_AVAILABLE FIXED DEC(7) INIT(0),
            3 SW_TELEMETRY_CODE FIXED DEC(7) INIT(0),
            3 SW_PURPOSE_CODE FIXED DEC(7) INIT(0),
            3 SW_RECORDER_TYPE_CODE FIXED DEC(7) INIT(0),
          2 OUTPUT,
            3 UNIQUE_SITE_IDENTIFIER FIXED DEC(7) INIT(0),
            3 AGENCY_STATION_NUMBER FIXED DEC(7) INIT(0),
            3 SW_ACTIVE_CODE FIXED DEC(7) INIT(0),
            3 SW_BEGIN_YEAR FIXED DEC(7) INIT(0),
            3 SW_END_YEAR FIXED DEC(7) INIT(0),
            3 SW_INTERRUPTED_CODE FIXED DEC(7) INIT(0),
            3 OTHER_SW_RELATED_DATA FIXED DEC(7) INIT(0),
            3 SW_TELEMETRY_CODE FIXED DEC(7) INIT(0),
            3 SW_DATA_PURPOSE_CODE(4) FIXED DEC(7) INIT((4)0),
            3 SW_RECORDER_TYPE_CODE FIXED DEC(7) INIT(0),
            3 STAGE_TYPE FIXED DEC(7) INIT(0),
            3 STAGE_FREQ_CODE FIXED DEC(7) INIT(0),
            3 FLOW_TYPE FIXED DEC(7) INIT(0),
            3 FLOW_FREQ_CODE  FIXED DEC(7) INIT(0),
            3 VOLUME_TYPE FIXED DEC(7) INIT(0),
            3 VOLUME_FREQ_CODE FIXED DEC(7) INIT(0),
            3 VELOCITY_TYPE FIXED DEC(7) INIT(0),
            3 VELOCITY_FREQ_CODE FIXED DEC(7) INIT(0),
            3 SW_DATA_STORAGE_MEDIA FIXED DEC(7) INIT(0),
            3 SW_SEASONAL_CODE FIXED DEC(7) INIT(0),
            3 SW_DATE_OF_LAST_UPDATE FIXED DEC(7) INIT(0),
            3 SW_DATA_CODE FIXED DEC(7) INIT(0),
            3 SW_DATA_BASE_IDENTIFIER(4) FIXED DEC(7) INIT((4)0),
            3 SW_DATA_BASE_SUBSET_IDENTIFIER(4) FIXED DEC(7)
                    INIT((4)0);
DECLARE 1 DATA_ELEMENT_COUNTS,
          2 INPUT,
            3 UNIQUE_SITE_IDENTIFIER FIXED DEC(7) INIT(0),
```

```
            3 AGENCY_STATION_NUMBER FIXED DEC(7) INIT(0),
            3 SW_ACTIVE_CODE FIXED DEC(7) INIT(0),
            3 SW_BEGIN_YEAR FIXED DEC(7) INIT(0),
            3 SW_END_YEAR FIXED DEC(7) INIT(0),
            3 SW_INTERRUPTED_CODE FIXED DEC(7) INIT(0),
            3 UNIT_FLOWS FIXED DEC(7) INIT(0),
            3 UNIT_STAGE FIXED DEC(7) INIT(0),
            3 UNIT_VOLUME FIXED DEC(7) INIT(0),
            3 SW_OTHER_DATA_AVAILABLE FIXED DEC(7) INIT(0),
            3 SW_TELEMETRY_CODE FIXED DEC(7) INIT(0),
            3 SW_PURPOSE_CODE FIXED DEC(7) INIT(0),
            3 SW_RECORDER_TYPE_CODE FIXED DEC(7) INIT(0),
          2 OUTPUT,
            3 UNIQUE_SITE_IDENTIFIER FIXED DEC(7) INIT(0),
            3 AGENCY_STATION_NUMBER FIXED DEC(7) INIT(0),
            3 SW_ACTIVE_CODE FIXED DEC(7) INIT(0),
            3 SW_BEGIN_YEAR FIXED DEC(7) INIT(0),
            3 SW_END_YEAR FIXED DEC(7) INIT(0),
            3 SW_INTERRUPTED_CODE FIXED DEC(7) INIT(0),
            3 OTHER_SW_RELATED_DATA FIXED DEC(7) INIT(0),
            3 SW_TELEMETRY_CODE FIXED DEC(7) INIT(0),
            3 SW_DATA_PURPOSE_CODE(4) FIXED DEC(7) INIT((4)0),
            3 SW_RECORDER_TYPE_CODE FIXED DEC(7) INIT(0),
            3 STAGE_TYPE FIXED DEC(7) INIT(0),
            3 STAGE_FREQ_CODE FIXED DEC(7) INIT(0),
            3 FLOW_TYPE FIXED DEC(7) INIT(0),
            3 FLOW_FREQ_CODE  FIXED DEC(7) INIT(0),
            3 VOLUME_TYPE FIXED DEC(7) INIT(0),
            3 VOLUME_FREQ_CODE FIXED DEC(7) INIT(0),
            3 VELOCITY_TYPE FIXED DEC(7) INIT(0),
            3 VELOCITY_FREQ_CODE FIXED DEC(7) INIT(0),
            3 SW_DATA_STORAGE_MEDIA FIXED DEC(7) INIT(0),
            3 SW_SEASONAL_CODE FIXED DEC(7) INIT(0),
            3 SW_DATE_OF_LAST_UPDATE FIXED DEC(7) INIT(0),
            3 SW_DATA_CODE FIXED DEC(7) INIT(0),
            3 SW_DATA_BASE_IDENTIFIER(4) FIXED DEC(7) INIT((4)0),
            3 SW_DATA_BASE_SUBSET_IDENTIFIER(4) FIXED DEC(7)
                    INIT((4)0);
DECLARE DATE BUILTIN;
ON ENDFILE (OLDMWDI) MWDI_NOT_EOF='0'B;
ON ENDFILE (WATSTOR)  WATSTORE_NOT_EOF='0'B;
OPEN FILE (OLDMWDI) INPUT RECORD SEQUENTIAL;
OPEN FILE (WATSTOR)  INPUT RECORD SEQUENTIAL;
OPEN FILE (NXTAVAL) UPDATE RECORD DIRECT;
OPEN FILE (NEWMWDI) OUTPUT RECORD SEQUENTIAL;
OPEN FILE (SWFRQHT)  OUTPUT RECORD SEQUENTIAL;
OPEN FILE (OTRTRNS)  OUTPUT RECORD SEQUENTIAL;
FIRST_TIME_THROUGH='1'B;
WATSTORE_NOT_EOF='1'B;
MWDI_NOT_EOF='1'B;
READ FILE(OLDMWDI) INTO(MWDI_INPUT);
READ FILE(WATSTOR)  INTO(WATSTORE_UV_RECORD);
CALL INPUT_SCREENING;
CALL WATSTORE_SCREEN;
```

```
DO WHILE (MWDI_NOT_EOF | WATSTORE_NOT_EOF);
  IF FIRST_TIME_THROUGH
    THEN DO;
      PREVIOUS_STATE=MWDI_INPUT.STATE_CODE;
      VALIDATE_ID = WATSTORE_UV_RECORD.STATION_IDENTIFIER;
      FIRST_TIME_THROUGH='0'B;
    END;
  IF MWDI_INPUT.STATE_CODE ^= PREVIOUS_STATE
    THEN DO;
      CALL PRINT_CONTROL_SUBS;
      PREVIOUS_STATE=MWDI_INPUT.STATE_CODE;
    END;
  IF WATSTORE_UV_RECORD.STATE_CODE = '' |
     WATSTORE_UV_RECORD.STATE_CODE = ' '
     THEN CURRENT_STATE = 999;
     ELSE CURRENT_STATE = WATSTORE_UV_RECORD.STATE_CODE;
  MWDI_OUTPUT=MWDI_TEMPLATE;
  FIRST_SW_VAL = '1'B;
  IF MWDI_INPUT.STATE_CODE < CURRENT_STATE |
     (MWDI_INPUT.STATE_CODE = CURRENT_STATE &
      MWDI_INPUT.AGENCY_CODE < WATSTORE_UV_RECORD.AGENCY_CODE) |
     (MWDI_INPUT.STATE_CODE = CURRENT_STATE &
      MWDI_INPUT.AGENCY_CODE = WATSTORE_UV_RECORD.AGENCY_CODE &
      MWDI_INPUT.AGENCY_STATION_NO <
      WATSTORE_UV_RECORD.STATION_IDENTIFIER) | ^WATSTORE_NOT_EOF
    THEN DO;
      PER_STATE_TOTAL_SITES_IN = PER_STATE_TOTAL_SITES_IN + 1;
      CALL INPUT_COUNTS;
      CONDITION_FLAG=1;
      CALL MOVE_ELEMENTS_TO_BE_KEPT;
      CALL SW_RELATED_ELEMENTS;
      CALL CONVERT_UV_ELEMENTS;
      CALL ASSIGN_SW_DB_IDENTIFIERS;
      PER_STATE_TOTAL_SITES_OUT = PER_STATE_TOTAL_SITES_OUT + 1;
      CALL OUTPUT_COUNTS;
      WRITE FILE(NEWMWDI) FROM(MWDI_OUTPUT);
      READ FILE(OLDMWDI) INTO(MWDI_INPUT);
      CALL INPUT_SCREENING;
    END;
    ELSE IF MWDI_INPUT.STATE_CODE = CURRENT_STATE &
            MWDI_INPUT.AGENCY_CODE = WATSTORE_UV_RECORD.AGENCY_CODE &
            MWDI_INPUT.AGENCY_STATION_NO =
            WATSTORE_UV_RECORD.STATION_IDENTIFIER
      THEN DO;
        PER_STATE_TOTAL_SITES_IN = PER_STATE_TOTAL_SITES_IN + 1;
        CALL INPUT_COUNTS;
        CONDITION_FLAG=2;
        CALL MOVE_ELEMENTS_TO_BE_KEPT;
        CALL SW_RELATED_ELEMENTS;
        CALL CONVERT_UV_ELEMENTS;
        NOT_COUNTED = '1'B;
        DO WHILE (MWDI_INPUT.AGENCY_STATION_NO =
           WATSTORE_UV_RECORD.STATION_IDENTIFIER & WATSTORE_NOT_EOF);
          IF WATSTORE_UV_RECORD.PARAMETER_CODE = 65 |
```

```
    WATSTORE_UV_RECORD.PARAMETER_CODE = 60 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 61 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 54 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 72021 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 72022 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 72023 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 72036 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 55 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 81904
   THEN DO;
     IF OTHER_DAY > 0
      THEN DO;
        LAST_PASS = '1'B;
        CALL OTHER_DATA_TRANSACTIONS;
        LAST_PASS = '0'B;
        OTHER_DAY = 0;
      END;
     ITS_SW_DATA = '1'B;
     ITS_SW_DATA2 = '1'B;
     IF NOT_COUNTED
       THEN DO;
         PER_STATE_SITES_WITH_FREQ_HIST =
           PER_STATE_SITES_WITH_FREQ_HIST + 1;
         NOT_COUNTED = '0'B;
       END;
     CALL SW_FREQUENCY_COMPS;
     CALL SW_ANN_FREQ_COMPS;
     CALL SW_PERIOD_OF_RECORD;
     CALL OTHER_SW_ACTIONS;
     IF NEW_RECORD = '1'B
       THEN CALL SW_ANN_FREQ_COMPS;
     PREV_ID = WATSTORE_UV_RECORD.STATION_IDENTIFIER;
     READ FILE(WATSTOR) INTO (WATSTORE_UV_RECORD);
     CALL WATSTORE_SCREEN;
     IF PREV_ID = WATSTORE_UV_RECORD.STATION_IDENTIFIER &
        WATSTORE_NOT_EOF
       THEN DO;
         IF SW_YEAR ^= WATSTORE_UV_RECORD.YEAR_NUMBER &
                       NUM_DAYS_THIS_PARM = 1
           THEN CALL SW_ANN_FREQ_COMPS;
         LAST_PASS = '0'B;
       END;
       ELSE DO;
         LAST_PASS = '1'B;
         CALL SW_ANN_FREQ_COMPS;
         CALL SW_PERIOD_OF_RECORD;
         LAST_PASS = '0'B;
         ITS_SW_DATA = '0'B;
       END;
  END;
ELSE IF WATSTORE_UV_RECORD.PARAMETER_CODE = 72019 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 81028 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 45 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 12 |
```

```
    WATSTORE_UV_RECORD.PARAMETER_CODE = 13 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 50 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 14 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 20 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 35 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 36 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 30 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 52 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 81026 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 82300 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 10 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 95 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 90095 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 300 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 400 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 70 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 76 |
    WATSTORE_UV_RECORD.PARAMETER_CODE = 80
THEN DO;
  IF ITS_SW_DATA
    THEN DO;
      LAST_PASS = '1'B;
      CALL SW_ANN_FREQ_COMPS;
      LAST_PASS = '0'B;
      ITS_SW_DATA = '0'B;
    END;
  CALL OTHER_DATA_TRANSACTIONS;
  PREV_ID = WATSTORE_UV_RECORD.STATION_IDENTIFIER;
  READ FILE(WATSTOR) INTO (WATSTORE_UV_RECORD);
  CALL WATSTORE_SCREEN;
  IF PREV_ID = WATSTORE_UV_RECORD.STATION_IDENTIFIER &
     WATSTORE_NOT_EOF
    THEN DO;
      IF OTHER_YEAR ^= WATSTORE_UV_RECORD.YEAR_NUMBER &
                       OTHER_DAY = 1
        THEN CALL OTHER_DATA_TRANSACTIONS;
      LAST_PASS = '0'B;
    END;
    ELSE DO;
      LAST_PASS = '1'B;
      CALL OTHER_DATA_TRANSACTIONS;
      LAST_PASS = '0'B;
    END;
END;
ELSE IF WATSTORE_UV_RECORD.PARAMETER_CODE = 74207
  THEN DO;
    CALL OTHER_DATA_AVAIL;
    READ FILE(WATSTOR)  INTO(WATSTORE_UV_RECORD);
    CALL WATSTORE_SCREEN;
  END;
  ELSE DO;
    READ FILE(WATSTOR) INTO(WATSTORE_UV_RECORD);
    CALL WATSTORE_SCREEN;
  END;
```

```
  END;
  CALL ASSIGN_SW_DB_IDENTIFIERS;
  ITS_SW_DATA2 = '0'B;
  PER_STATE_TOTAL_SITES_OUT = PER_STATE_TOTAL_SITES_OUT + 1;
  CALL OUTPUT_COUNTS;
  WRITE FILE(NEWMWDI) FROM(MWDI_OUTPUT);
  READ FILE(OLDMWDI) INTO(MWDI_INPUT);
  CALL INPUT_SCREENING;
END;
ELSE IF MWDI_INPUT.STATE_CODE > CURRENT_STATE |
        (MWDI_INPUT.STATE_CODE = CURRENT_STATE &
        MWDI_INPUT.AGENCY_CODE > WATSTORE_UV_RECORD.AGENCY_CODE)
        | (MWDI_INPUT.STATE_CODE = CURRENT_STATE
        & MWDI_INPUT.AGENCY_CODE = WATSTORE_UV_RECORD.AGENCY_CODE
        & MWDI_INPUT.AGENCY_STATION_NO >
        WATSTORE_UV_RECORD.STATION_IDENTIFIER) | ^MWDI_NOT_EOF
  THEN DO;
    PER_STATE_TOTAL_SITES_ADDED = PER_STATE_TOTAL_SITES_ADDED +1;
    CONDITION_FLAG=3;
    CALL INDEX_NEW_SITE;
    PREVIOUS_STATION_ID=WATSTORE_UV_RECORD.STATION_IDENTIFIER;
    NOT_COUNTED = '1'B;
    DO WHILE (PREVIOUS_STATION_ID =
      WATSTORE_UV_RECORD.STATION_IDENTIFIER & WATSTORE_NOT_EOF);
    IF WATSTORE_UV_RECORD.PARAMETER_CODE = 65 |
       WATSTORE_UV_RECORD.PARAMETER_CODE = 60 |
       WATSTORE_UV_RECORD.PARAMETER_CODE = 61 |
       WATSTORE_UV_RECORD.PARAMETER_CODE = 54 |
       WATSTORE_UV_RECORD.PARAMETER_CODE = 72021 |
       WATSTORE_UV_RECORD.PARAMETER_CODE = 72022 |
       WATSTORE_UV_RECORD.PARAMETER_CODE = 72023 |
       WATSTORE_UV_RECORD.PARAMETER_CODE = 72036 |
       WATSTORE_UV_RECORD.PARAMETER_CODE = 55 |
       WATSTORE_UV_RECORD.PARAMETER_CODE = 81904
      THEN DO;
        IF OTHER_DAY > 0
          THEN DO;
            LAST_PASS = '1'B;
            CALL OTHER_DATA_TRANSACTIONS;
            LAST_PASS = '0'B;
            OTHER_DAY = 0;
          END;
        ITS_SW_DATA = '1'B;
        ITS_SW_DATA2 = '1'B;
        IF NOT_COUNTED
          THEN DO;
            PER_STATE_SITES_WITH_FREQ_HIST =
              PER_STATE_SITES_WITH_FREQ_HIST + 1;
            NOT_COUNTED = '0'B;
          END;
        CALL SW_FREQUENCY_COMPS;
        CALL SW_ANN_FREQ_COMPS;
        CALL SW_PERIOD_OF_RECORD;
        CALL OTHER_SW_ACTIONS;
```

```
    IF NEW_RECORD = '1'B
      THEN CALL SW_ANN_FREQ_COMPS;
    PREV_ID = WATSTORE_UV_RECORD.STATION_IDENTIFIER;
    READ FILE(WATSTOR) INTO (WATSTORE_UV_RECORD);
    CALL WATSTORE_SCREEN;
    IF PREV_ID = WATSTORE_UV_RECORD.STATION_IDENTIFIER &
       WATSTORE_NOT_EOF
      THEN DO;
        IF SW_YEAR ^= WATSTORE_UV_RECORD.YEAR_NUMBER &
                      NUM_DAYS_THIS_PARM = 1
          THEN CALL SW_ANN_FREQ_COMPS;
        LAST_PASS = '0'B;
      END;
      ELSE DO;
        LAST_PASS = '1'B;
        CALL SW_ANN_FREQ_COMPS;
        CALL SW_PERIOD_OF_RECORD;
        LAST_PASS = '0'B;
        ITS_SW_DATA = '0'B;
      END;
  END;
ELSE IF WATSTORE_UV_RECORD.PARAMETER_CODE = 72019 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 81028 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 45 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 12 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 13 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 50 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 14 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 20 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 35 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 36 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 30 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 52 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 81026 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 82300 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 10 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 95 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 90095 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 300 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 400 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 70 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 76 |
   WATSTORE_UV_RECORD.PARAMETER_CODE = 80
 THEN DO;
   IF ITS_SW_DATA
     THEN DO;
       LAST_PASS = '1'B;
       CALL SW_ANN_FREQ_COMPS;
       LAST_PASS = '0'B;
       ITS_SW_DATA = '0'B;
     END;
   CALL OTHER_DATA_TRANSACTIONS;
   PREV_ID = WATSTORE_UV_RECORD.STATION_IDENTIFIER;
   READ FILE(WATSTOR) INTO (WATSTORE_UV_RECORD);
```

```
            CALL WATSTORE_SCREEN;
            IF PREV_ID = WATSTORE_UV_RECORD.STATION_IDENTIFIER &
               WATSTORE_NOT_EOF
             THEN DO;
               IF OTHER_YEAR ^= WATSTORE_UV_RECORD.YEAR_NUMBER &
                                OTHER_DAY = 1
                 THEN CALL OTHER_DATA_TRANSACTIONS;
               LAST_PASS = '0'B;
             END;
              ELSE DO;
                LAST_PASS = '1'B;
                CALL OTHER_DATA_TRANSACTIONS;
                LAST_PASS = '0'B;
              END;
          END;
         ELSE IF WATSTORE_UV_RECORD.PARAMETER_CODE = 74207
           THEN DO;
             CALL OTHER_DATA_AVAIL;
             READ FILE(WATSTOR) INTO(WATSTORE_UV_RECORD);
             CALL WATSTORE_SCREEN;
           END;
           ELSE DO;
             READ FILE(WATSTOR)  INTO(WATSTORE_UV_RECORD);
             CALL WATSTORE_SCREEN;
           END;
         END;
         CALL ASSIGN_SW_DB_IDENTIFIERS;
         ITS_SW_DATA2 = '0'B;
         PER_STATE_TOTAL_SITES_OUT = PER_STATE_TOTAL_SITES_OUT + 1;
         CALL OUTPUT_COUNTS;
         WRITE FILE(NEWMWDI) FROM(MWDI_OUTPUT);
       END;
       ELSE DO;
         PUT FILE(SYSPRINT) EDIT('FATAL ERROR -- UNKNOWN CAUSE')
                 (A(28));
         STOP;
       END;
END;
CALL PRINT_CONTROL_SUBS;
CALL PRINT_CONTROL_TOTALS;
CALL PRINT_VALIDATED_SW_DATA;
MOVE_ELEMENTS_TO_BE_KEPT:PROC;
 DECLARE I FIXED DEC;
 MWDI_OUTPUT.SITE_ID = MWDI_INPUT.SITE_ID;
 MWDI_OUTPUT.AGENCY_CODE = MWDI_INPUT.AGENCY_CODE;
 MWDI_OUTPUT.AGENCY_STATION_NO = MWDI_INPUT.AGENCY_STATION_NO;
 MWDI_OUTPUT.LOCAL_WELL_NO = MWDI_INPUT.LOCAL_WELL_NO;
 MWDI_OUTPUT.STATION_NAME = MWDI_INPUT.STATION_NAME;
 MWDI_OUTPUT.LATITUDE = MWDI_INPUT.LATITUDE;
 MWDI_OUTPUT.LONGITUDE = MWDI_INPUT.LONGITUDE;
 MWDI_OUTPUT.LAT_LONG_ACCURACY = MWDI_INPUT.LAT_LONG_ACCURACY;
 MWDI_OUTPUT.SITE_TYPE = MWDI_INPUT.SITE_TYPE;
 MWDI_OUTPUT.COUNTRY_CODE = MWDI_INPUT.COUNTRY_CODE;
 MWDI_OUTPUT.STATE_CODE=MWDI_INPUT.STATE_CODE;
```

```
MWDI_OUTPUT.COUNTY_CODE=MWDI_INPUT.COUNTY_CODE;
MWDI_OUTPUT.STATE_COUNTY_CODE = MWDI_INPUT.STATE_COUNTY_CODE;
MWDI_OUTPUT.DISTRICT_CODE = MWDI_INPUT.DISTRICT_CODE;
MWDI_OUTPUT.HYDRO_UNIT_CODE = MWDI_INPUT.HYDRO_UNIT_CODE;
MWDI_OUTPUT.OFFICE_CODE = MWDI_INPUT.OFFICE_CODE;
MWDI_OUTPUT.TOTAL_DRAINAGE_AREA = MWDI_INPUT.TOTAL_DRAINAGE_AREA;
MWDI_OUTPUT.CONTRIB_DRAINAGE_AREA = MWDI_INPUT.CONTRIB_DRAINAGE_AREA;
MWDI_OUTPUT.ALTITUDE_DATUM = MWDI_INPUT.ALTITUDE_DATUM;
MWDI_OUTPUT.ALTITUDE_METH_OF_MEAS = MWDI_INPUT.ALTITUDE_METH_OF_MEAS;
MWDI_OUTPUT.ALTITUDE_ACCURACY = MWDI_INPUT.ALTITUDE_ACCURACY;
MWDI_OUTPUT.RIVER_REACH_NO = MWDI_INPUT.RIVER_REACH_NO;
MWDI_OUTPUT.RIVER_REACH_SEG_FLAG = MWDI_INPUT.RIVER_REACH_SEG_FLAG;
MWDI_OUTPUT.DEPTH_OF_WELL = MWDI_INPUT.DEPTH_OF_WELL;
MWDI_OUTPUT.SOURCE_WELL_DEPTH_DATA =MWDI_INPUT.SOURCE_WELL_DEPTH_DATA;
MWDI_OUTPUT.DEPTH_OF_HOLE = MWDI_INPUT.DEPTH_OF_HOLE;
MWDI_OUTPUT.PRINCIPAL_AQUIFER_CODE =MWDI_INPUT.PRINCIPAL_AQUIFER_CODE;
MWDI_OUTPUT.AQUIFER_TYPE_CODE = MWDI_INPUT.AQUIFER_TYPE_CODE;
MWDI_OUTPUT.CONSOLIDATED_AQUIFER_CODE =
   MWDI_INPUT.CONSOLIDATED_AQUIFER_CODE;
DO I=1 TO 3;
  MWDI_OUTPUT.BASIN_DESCRIPTOR(I) = MWDI_INPUT.BASIN_DESCRIPTOR(I);
  MWDI_OUTPUT.USE_OF_WATER(I) = MWDI_INPUT.USE_OF_WATER(I);
  MWDI_OUTPUT.USE_OF_SITE(I) = MWDI_INPUT.USE_OF_SITE(I);
END;
DO I=1 TO 6;
  MWDI_OUTPUT.OTHER_DATA_AVAILABLE(I) =
     MWDI_INPUT.OTHER_DATA_AVAILABLE(I);
END;
MWDI_OUTPUT.SW_ACTIVE_CODE = MWDI_INPUT.SW_ACTIVE_CODE;
MWDI_OUTPUT.GW_ACTIVE_CODE = MWDI_INPUT.GW_ACTIVE_CODE;
MWDI_OUTPUT.QW_ACTIVE_CODE = MWDI_INPUT.QW_ACTIVE_CODE;
MWDI_OUTPUT.BIO_ACTIVE_CODE = MWDI_INPUT.BIO_ACTIVE_CODE;
MWDI_OUTPUT.PHY_ACTIVE_CODE = MWDI_INPUT.PHY_ACTIVE_CODE;
MWDI_OUTPUT.SED_ACTIVE_CODE = MWDI_INPUT.SED_ACTIVE_CODE;
MWDI_OUTPUT.CHM_ACTIVE_CODE = MWDI_INPUT.CHM_ACTIVE_CODE;
MWDI_OUTPUT.MET_ACTIVE_CODE = MWDI_INPUT.MET_ACTIVE_CODE;
MWDI_OUTPUT.DATE_OF_LAST_UPDATE = MWDI_INPUT.DATE_OF_LAST_UPDATE;
MWDI_OUTPUT.DIS_NODE_IDENT = MWDI_INPUT.DIS_NODE_IDENT;
MWDI_OUTPUT.SW_DATA_CODE = MWDI_INPUT.SW_DATA_CODE;
MWDI_OUTPUT.GW_DATA_CODE = MWDI_INPUT.GW_DATA_CODE;
MWDI_OUTPUT.QW_DATA_CODE = MWDI_INPUT.QW_DATA_CODE;
MWDI_OUTPUT.MET_DATA_CODE = MWDI_INPUT.MET_DATA_CODE;
MWDI_OUTPUT.OTHER_DATA_CODE = MWDI_INPUT.OTHER_DATA_CODE;
MWDI_OUTPUT.NODE_DATE_OF_LAST_UPDATE =
   MWDI_INPUT.NODE_DATE_OF_LAST_UPDATE;
IF MWDI_INPUT.SW_BEGIN_YEAR >= '1850' &
   MWDI_INPUT.SW_BEGIN_YEAR <= '1986'
   THEN MWDI_OUTPUT.SW_BEGIN_YEAR = MWDI_INPUT.SW_BEGIN_YEAR;
   ELSE MWDI_OUTPUT.SW_BEGIN_YEAR = '0000';
IF MWDI_INPUT.SW_END_YEAR >= '1850' &
   MWDI_INPUT.SW_END_YEAR <= '1986'
   THEN MWDI_OUTPUT.SW_END_YEAR = MWDI_INPUT.SW_END_YEAR;
   ELSE MWDI_OUTPUT.SW_END_YEAR = '0000';
MWDI_OUTPUT.SW_INTERRUPTED_CODE = MWDI_INPUT.SW_INTERRUPTED_CODE;
```

```
MWDI_OUTPUT.COMPLETE_STAGE = MWDI_INPUT.COMPLETE_STAGE;
MWDI_OUTPUT.PEAK_STAGE = MWDI_INPUT.PEAK_STAGE;
MWDI_OUTPUT.LOW_STAGE = MWDI_INPUT.LOW_STAGE;
MWDI_OUTPUT.STAGE_STORAGE_MEDIA = MWDI_INPUT.STAGE_STORAGE_MEDIA;
MWDI_OUTPUT.COMPLETE_FLOW = MWDI_INPUT.COMPLETE_FLOW;
MWDI_OUTPUT.PEAK_FLOW = MWDI_INPUT.PEAK_FLOW;
MWDI_OUTPUT.LOW_FLOW = MWDI_INPUT.LOW_FLOW;
MWDI_OUTPUT.MISC_FLOW_MEAS = MWDI_INPUT.MISC_FLOW_MEAS;
MWDI_OUTPUT.FLOW_STORAGE_MEDIA = MWDI_INPUT.FLOW_STORAGE_MEDIA;
MWDI_OUTPUT.VOLUME = MWDI_INPUT.VOLUME;
MWDI_OUTPUT.VOLUME_CHANGE = MWDI_INPUT.VOLUME_CHANGE;
MWDI_OUTPUT.VOLUME_STORAGE_MEDIA = MWDI_INPUT.VOLUME_STORAGE_MEDIA;
IF MWDI_INPUT.GW_BEGIN_YEAR >= '1850' &
   MWDI_INPUT.GW_BEGIN_YEAR <= '1986'
   THEN MWDI_OUTPUT.GW_BEGIN_YEAR = MWDI_INPUT.GW_BEGIN_YEAR;
   ELSE MWDI_OUTPUT.GW_BEGIN_YEAR = '0000';
IF MWDI_INPUT.GW_END_YEAR >= '1850' &
   MWDI_INPUT.GW_END_YEAR <= '1986'
   THEN MWDI_OUTPUT.GW_END_YEAR = MWDI_INPUT.GW_END_YEAR;
   ELSE MWDI_OUTPUT.GW_END_YEAR = '0000';
MWDI_OUTPUT.GW_INTERRUPTED_CODE = MWDI_INPUT.GW_INTERRUPTED_CODE;
MWDI_OUTPUT.WATER_LEVEL_FREQ = MWDI_INPUT.WATER_LEVEL_FREQ;
MWDI_OUTPUT.WATER_LEVEL_STORAGE_MEDIA =
  MWDI_INPUT.WATER_LEVEL_STORAGE_MEDIA;
MWDI_OUTPUT.DISCHARGE_FREQUENCY = MWDI_INPUT.DISCHARGE_FREQUENCY;
MWDI_OUTPUT.DISCHARGE_STORAGE_MEDIA =
  MWDI_INPUT.DISCHARGE_STORAGE_MEDIA;
MWDI_OUTPUT.SUBSIDENCE_FREQ = MWDI_INPUT.SUBSIDENCE_FREQ;
MWDI_OUTPUT.SUBSIDENCE_STORAGE_MEDIA =
  MWDI_INPUT.SUBSIDENCE_STORAGE_MEDIA;
MWDI_OUTPUT.GW_OTHER_DATA_AVAILABLE =
  MWDI_INPUT.GW_OTHER_DATA_AVAILABLE;
MWDI_OUTPUT.MAJOR_VARIATIONS_CODE = MWDI_INPUT.MAJOR_VARIATIONS_CODE;
IF MWDI_INPUT.GW_TELEMETRY_CODE >= '0' &
   MWDI_INPUT.GW_TELEMETRY_CODE < '9'
   THEN MWDI_OUTPUT.GW_TELEMETRY_CODE = MWDI_INPUT.GW_TELEMETRY_CODE;
   ELSE MWDI_OUTPUT.GW_TELEMETRY_CODE = '';
MWDI_OUTPUT.GW_DATE_OF_LAST_UPDATE= MWDI_INPUT.GW_DATE_OF_LAST_UPDATE;
MWDI_OUTPUT.GW_PURPOSE_CODE = MWDI_INPUT.GW_PURPOSE_CODE;
MWDI_OUTPUT.GW_RECORDER_TYPE_CODE = MWDI_INPUT.GW_RECORDER_TYPE_CODE;
MWDI_OUTPUT.GW_RECORDER_FREQ_CODE = MWDI_INPUT.GW_RECORDER_FREQ_CODE;
IF MWDI_INPUT.QW_BEGIN_YEAR >= '1850' &
   MWDI_INPUT.QW_BEGIN_YEAR <= '1986'
   THEN MWDI_OUTPUT.QW_BEGIN_YEAR = MWDI_INPUT.QW_BEGIN_YEAR;
   ELSE MWDI_OUTPUT.QW_BEGIN_YEAR = '0000';
IF MWDI_INPUT.QW_END_YEAR >= '1850' &
   MWDI_INPUT.QW_END_YEAR <= '1986'
   THEN MWDI_OUTPUT.QW_END_YEAR = MWDI_INPUT.QW_END_YEAR;
   ELSE MWDI_OUTPUT.QW_END_YEAR = '0000';
MWDI_OUTPUT.QW_INTERRUPTED_CODE = MWDI_INPUT.QW_INTERRUPTED_CODE;
MWDI_OUTPUT.QW_TELEMETRY_CODE = MWDI_INPUT.QW_TELEMETRY_CODE;
MWDI_OUTPUT.QW_DATE_OF_LAST_UPDATE= MWDI_INPUT.QW_DATE_OF_LAST_UPDATE;
MWDI_OUTPUT.QW_PURPOSE_CODE = MWDI_INPUT.QW_PURPOSE_CODE;
MWDI_OUTPUT.QW_RECORDER_TYPE_CODE = MWDI_INPUT.QW_RECORDER_TYPE_CODE;
```

```
MWDI_OUTPUT.QW_RECORDER_FREQ_CODE = MWDI_INPUT.QW_RECORDER_FREQ_CODE;
MWDI_OUTPUT.STORET_POINTER = MWDI_INPUT.STORET_POINTER;
MWDI_OUTPUT.ENTERIC_BACTERIA = MWDI_INPUT.ENTERIC_BACTERIA;
MWDI_OUTPUT.NATIVE_BACTERIA = MWDI_INPUT.NATIVE_BACTERIA;
MWDI_OUTPUT.PHYTOPLANKTON = MWDI_INPUT.PHYTOPLANKTON;
MWDI_OUTPUT.ZOOPLANKTON = MWDI_INPUT.ZOOPLANKTON;
MWDI_OUTPUT.PERIPHYTON = MWDI_INPUT.PERIPHYTON;
MWDI_OUTPUT.MACROPHYTON = MWDI_INPUT.MACROPHYTON;
MWDI_OUTPUT.MICROINVERTEBRATES = MWDI_INPUT.MICROINVERTEBRATES;
MWDI_OUTPUT.MACROINVERTEBRATES = MWDI_INPUT.MACROINVERTEBRATES;
MWDI_OUTPUT.VERTEBRATES = MWDI_INPUT.VERTEBRATES;
MWDI_OUTPUT.FUNGI = MWDI_INPUT.FUNGI;
MWDI_OUTPUT.VIRUSES = MWDI_INPUT.VIRUSES;
MWDI_OUTPUT.BIO_BEGIN_YEAR = MWDI_INPUT.BIO_BEGIN_YEAR;
MWDI_OUTPUT.BIO_END_YEAR = MWDI_INPUT.BIO_END_YEAR;
MWDI_OUTPUT.BIO_DATE_OF_LAST_UPDATE =
  MWDI_INPUT.BIO_DATE_OF_LAST_UPDATE;
MWDI_OUTPUT.BIO_STORAGE_MEDIA = MWDI_INPUT.BIO_STORAGE_MEDIA;
MWDI_OUTPUT.PRIMARY_PRODUCTIVITY = MWDI_INPUT.PRIMARY_PRODUCTIVITY;
MWDI_OUTPUT.SECONDARY_PRODUCTIVITY= MWDI_INPUT.SECONDARY_PRODUCTIVITY;
MWDI_OUTPUT.CHEMOSYNTHETIC_ACTIVITY =
  MWDI_INPUT.CHEMOSYNTHETIC_ACTIVITY;
MWDI_OUTPUT.BIOSTIMULATORY_TEST = MWDI_INPUT.BIOSTIMULATORY_TEST;
MWDI_OUTPUT.TOXICITY_TEST = MWDI_INPUT.TOXICITY_TEST;
MWDI_OUTPUT.OTHER_BIO_ASSAY_TEST = MWDI_INPUT.OTHER_BIO_ASSAY_TEST;
MWDI_OUTPUT.CHEMICAL_TISSUE_ANALYSIS =
  MWDI_INPUT.CHEMICAL_TISSUE_ANALYSIS;
MWDI_OUTPUT.HISTOPATHIC_ANALYSIS = MWDI_INPUT.HISTOPATHIC_ANALYSIS;
MWDI_OUTPUT.OTHER_TISSUE_ANALYSES = MWDI_INPUT.OTHER_TISSUE_ANALYSES;
MWDI_OUTPUT.TEMPERATURE = MWDI_INPUT.TEMPERATURE;
MWDI_OUTPUT.SPECIFIC_CONDUCTANCE = MWDI_INPUT.SPECIFIC_CONDUCTANCE;
MWDI_OUTPUT.TURBIDITY = MWDI_INPUT.TURBIDITY;
MWDI_OUTPUT.COLOR = MWDI_INPUT.COLOR;
MWDI_OUTPUT.ODOR = MWDI_INPUT.ODOR;
MWDI_OUTPUT.PH = MWDI_INPUT.PH;
MWDI_OUTPUT.SUSPENDED_SOLIDS = MWDI_INPUT.SUSPENDED_SOLIDS;
MWDI_OUTPUT.PHY_BEGIN_YEAR = MWDI_INPUT.PHY_BEGIN_YEAR;
MWDI_OUTPUT.PHY_END_YEAR = MWDI_INPUT.PHY_END_YEAR;
MWDI_OUTPUT.PHY_DATE_OF_LAST_UPDATE =
  MWDI_INPUT.PHY_DATE_OF_LAST_UPDATE;
MWDI_OUTPUT.PHY_STORAGE_MEDIA = MWDI_INPUT.PHY_STORAGE_MEDIA;
MWDI_OUTPUT.BED_LOAD = MWDI_INPUT.BED_LOAD;
MWDI_OUTPUT.SUSPENDED_CONC = MWDI_INPUT.SUSPENDED_CONC;
MWDI_OUTPUT.TOTAL_CONC = MWDI_INPUT.TOTAL_CONC;
MWDI_OUTPUT.SUSPENDED_PARTICLE_SIZE =
  MWDI_INPUT.SUSPENDED_PARTICLE_SIZE;
MWDI_OUTPUT.BED_PARTICLE_SIZE = MWDI_INPUT.BED_PARTICLE_SIZE;
MWDI_OUTPUT.SUSPENDED_SEDIMENT_DISCH =
  MWDI_INPUT.SUSPENDED_SEDIMENT_DISCH;
MWDI_OUTPUT.TOTAL_SEDIMENT_DISCH = MWDI_INPUT.TOTAL_SEDIMENT_DISCH;
MWDI_OUTPUT.SED_BEGIN_YEAR = MWDI_INPUT.SED_BEGIN_YEAR;
MWDI_OUTPUT.SED_END_YEAR = MWDI_INPUT.SED_END_YEAR;
MWDI_OUTPUT.SED_DATE_OF_LAST_UPDATE =
  MWDI_INPUT.SED_DATE_OF_LAST_UPDATE;
```

```
MWDI_OUTPUT.SED_STORAGE_MEDIA = MWDI_INPUT.SED_STORAGE_MEDIA;
MWDI_OUTPUT.DISSOLVED_SOLIDS = MWDI_INPUT.DISSOLVED_SOLIDS;
MWDI_OUTPUT.MAJOR_IONS = MWDI_INPUT.MAJOR_IONS;
MWDI_OUTPUT.HARDNESS = MWDI_INPUT.HARDNESS;
MWDI_OUTPUT.SILICA = MWDI_INPUT.SILICA;
MWDI_OUTPUT.PHOSPHORUS = MWDI_INPUT.PHOSPHORUS;
MWDI_OUTPUT.PHOSPHORUS_SPECIES = MWDI_INPUT.PHOSPHORUS_SPECIES;
MWDI_OUTPUT.NITROGEN = MWDI_INPUT.NITROGEN;
MWDI_OUTPUT.NITROGEN_SPECIES = MWDI_INPUT.NITROGEN_SPECIES;
MWDI_OUTPUT.DETERGENTS = MWDI_INPUT.DETERGENTS;
MWDI_OUTPUT.OTHER_MINOR_CONSTITUENTS =
  MWDI_INPUT.OTHER_MINOR_CONSTITUENTS;
MWDI_OUTPUT.RADIOACTIVITY = MWDI_INPUT.RADIOACTIVITY;
MWDI_OUTPUT.RADIOCHEMICAL_SPECIES = MWDI_INPUT.RADIOCHEMICAL_SPECIES;
MWDI_OUTPUT.CARBON = MWDI_INPUT.CARBON;
MWDI_OUTPUT.ORGANIC_GROUPS = MWDI_INPUT.ORGANIC_GROUPS;
MWDI_OUTPUT.PESTICIDE_SPECIES = MWDI_INPUT.PESTICIDE_SPECIES;
MWDI_OUTPUT.OTHER_ORGANIC_SPECIES = MWDI_INPUT.OTHER_ORGANIC_SPECIES;
MWDI_OUTPUT.BIOCHEMICAL_OXYGEN_DEMAND =
  MWDI_INPUT.BIOCHEMICAL_OXYGEN_DEMAND;
MWDI_OUTPUT.CHEMICAL_OXYGEN_DEMAND =
  MWDI_INPUT.CHEMICAL_OXYGEN_DEMAND;
MWDI_OUTPUT.DISSOLVED_OXYGEN = MWDI_INPUT.DISSOLVED_OXYGEN;
MWDI_OUTPUT.OTHER_DISSOLVED_GASES = MWDI_INPUT.OTHER_DISSOLVED_GASES;
MWDI_OUTPUT.CHM_BEGIN_YEAR = MWDI_INPUT.CHM_BEGIN_YEAR;
MWDI_OUTPUT.CHM_END_YEAR = MWDI_INPUT.CHM_END_YEAR;
MWDI_OUTPUT.CHM_DATE_OF_LAST_UPDATE =
  MWDI_INPUT.CHM_DATE_OF_LAST_UPDATE;
MWDI_OUTPUT.CHM_STORAGE_MEDIA = MWDI_INPUT.CHM_STORAGE_MEDIA;
MWDI_OUTPUT.MET_BEGIN_DATE = MWDI_INPUT.MET_BEGIN_DATE;
MWDI_OUTPUT.MET_END_DATE = MWDI_INPUT.MET_END_DATE;
MWDI_OUTPUT.MET_INTERRUPTED_CODE = MWDI_INPUT.MET_INTERRUPTED_CODE;
MWDI_OUTPUT.RAINFALL = MWDI_INPUT.RAINFALL;
MWDI_OUTPUT.UNIT_RAINFALL = MWDI_INPUT.UNIT_RAINFALL;
MWDI_OUTPUT.AIR_TEMPERATURE = MWDI_INPUT.AIR_TEMPERATURE;
MWDI_OUTPUT.WIND_VELOCITY = MWDI_INPUT.WIND_VELOCITY;
MWDI_OUTPUT.MET_OTHER_DATA_AVAILABLE =
  MWDI_INPUT.MET_OTHER_DATA_AVAILABLE;
MWDI_OUTPUT.MET_TELEMETRY_CODE = MWDI_INPUT.MET_TELEMETRY_CODE;
MWDI_OUTPUT.MET_DATE_OF_LAST_UPDATE =
  MWDI_INPUT.MET_DATE_OF_LAST_UPDATE;
MWDI_OUTPUT.MET_STORAGE_MEDIA = MWDI_INPUT.MET_STORAGE_MEDIA;
MWDI_OUTPUT.MET_RECORDER_TYPE_CODE =
  MWDI_INPUT.MET_RECORDER_TYPE_CODE;
MWDI_OUTPUT.MET_RECORDER_FREQ_CODE =
  MWDI_INPUT.MET_RECORDER_FREQ_CODE;
DO I=1 TO 10;
  MWDI_OUTPUT.GW_POINTER(I) = MWDI_INPUT.GW_POINTER(I);
  MWDI_OUTPUT.GW_MOD_FILE(I) = MWDI_INPUT.GW_MOD_FILE(I);
  MWDI_OUTPUT.QW_POINTER(I) = MWDI_INPUT.QW_POINTER(I);
  MWDI_OUTPUT.QW_MOD_FILE(I) = MWDI_INPUT.QW_MOD_FILE(I);
  MWDI_OUTPUT.BIO_POINTER(I) = MWDI_INPUT.BIO_POINTER(I);
  MWDI_OUTPUT.BIO_MOD_FILE(I) = MWDI_INPUT.BIO_MOD_FILE(I);
  MWDI_OUTPUT.PHY_POINTER(I) = MWDI_INPUT.PHY_POINTER(I);
```

```
    MWDI_OUTPUT.PHY_MOD_FILE(I) = MWDI_INPUT.PHY_MOD_FILE(I);
    MWDI_OUTPUT.SED_POINTER(I) = MWDI_INPUT.SED_POINTER(I);
    MWDI_OUTPUT.SED_MOD_FILE(I) = MWDI_INPUT.SED_MOD_FILE(I);
    MWDI_OUTPUT.CHM_POINTER(I) = MWDI_INPUT.CHM_POINTER(I);
    MWDI_OUTPUT.CHM_MOD_FILE(I) = MWDI_INPUT.CHM_MOD_FILE(I);
    MWDI_OUTPUT.WRD_PROJECT_NUMBER(I) =
      MWDI_INPUT.WRD_PROJECT_NUMBER(I);
    MWDI_OUTPUT.NETWORK_CODE(I) = MWDI_INPUT.NETWORK_CODE(I);
    MWDI_OUTPUT.OTHER_SOURCE_AGENCY(I) =
      MWDI_INPUT.OTHER_SOURCE_AGENCY(I);
    MWDI_OUTPUT.SOURCE_FILE_IDENTIFIER(I) =
      MWDI_INPUT.SOURCE_FILE_IDENTIFIER(I);
    MWDI_OUTPUT.SOURCE_FILE_AGENCY(I) =
      MWDI_INPUT.SOURCE_FILE_AGENCY(I);
    MWDI_OUTPUT.MET_POINTER(I) = MWDI_INPUT.MET_POINTER(I);
    MWDI_OUTPUT.MET_MOD_FILE(I) = MWDI_INPUT.MET_MOD_FILE(I);
  END;
END MOVE_ELEMENTS_TO_BE_KEPT;
SW_RELATED_ELEMENTS:PROC;
  DECLARE INTERMEDIATE_DATE FIXED BIN;
  DECLARE I FIXED BIN;
  DECLARE (INDEX,SUBSTR,LENGTH) BUILTIN;
  IF MWDI_INPUT.SW_TELEMETRY_CODE ^= 0
    THEN MWDI_OUTPUT.SW_TELEMETRY_CODE=MWDI_INPUT.SW_TELEMETRY_CODE;
    ELSE MWDI_OUTPUT.SW_TELEMETRY_CODE = '';
  MWDI_OUTPUT.SW_RECORDER_TYPE_CODE=MWDI_INPUT.SW_RECORDER_TYPE_CODE;
  DO I=1 TO 4;
    MWDI_OUTPUT.SW_DATA_PURPOSE_CODE(I) =
              SUBSTR(MWDI_INPUT.SW_PURPOSE_CODE,I,1);
  END;
  DO I=1 TO 12;
    MWDI_OUTPUT.OTHER_SW_RELATED_DATA(I) =
               SUBSTR(MWDI_INPUT.SW_OTHER_DATA_AVAIL,I,1);
  END;
  IF MWDI_INPUT.SW_DATE_OF_LAST_UPDATE = ''
    THEN INTERMEDIATE_DATE = 0;
    ELSE DO;
      INTERMEDIATE_DATE = MWDI_INPUT.SW_DATE_OF_LAST_UPDATE;
      MWDI_OUTPUT.SW_DATE_OF_LAST_UPDATE = 19000001 +
              INTERMEDIATE_DATE * 100;
    END;
END SW_RELATED_ELEMENTS;
CONVERT_UV_ELEMENTS:PROC;
  IF MWDI_INPUT.UNIT_FLOW >= 'A' & MWDI_INPUT.UNIT_FLOW <= '9'
    THEN DO;
      MWDI_OUTPUT.FLOW_TYPE = 'C ';
      MWDI_OUTPUT.FLOW_FREQ_CODE = 'C';
    END;
  IF MWDI_INPUT.UNIT_STAGE >= 'A' & MWDI_INPUT.UNIT_STAGE <= '9'
    THEN DO;
      MWDI_OUTPUT.STAGE_TYPE = 'C ';
      MWDI_OUTPUT.STAGE_FREQ_CODE = 'C';
    END;
  IF MWDI_INPUT.UNIT_VOLUME >= 'A' & MWDI_INPUT.UNIT_VOLUME <= '9'
```

```
   THEN DO;
     MWDI_OUTPUT.VOLUME_TYPE = 'C ';
     MWDI_OUTPUT.VOLUME_FREQ_CODE = 'C';
   END;
END CONVERT_UV_ELEMENTS;
ASSIGN_SW_DB_IDENTIFIERS:PROC;
 DECLARE (I,J,K) FIXED DEC;
 DECLARE (FULL,PASS,NOT_FOUND) BIT(1);
 IF CONDITION_FLAG = 1 | CONDITION_FLAG = 2
   THEN DO;
     J=0;
     DO I=1 TO 10;
       IF J >= 4
         THEN DO;
           DO K=1 TO 4;
             IF MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(K) = 'NWIS'
               THEN DO;
                 MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(K) = '    ';
                 MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(K) = '    ';
               END;
           END;
           PASS = '1'B;
           K=1;
           DO WHILE(PASS);
             IF MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(K) = '    '
               THEN DO;
                 MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(K) = 'NWIS';
                 PASS = '0'B;
               END;
               ELSE K=K+1;
           END;
           IF MWDI_INPUT.SW_MODIFIERS(I).SW_POINTER = '    '
             THEN;
             ELSE PUT SKIP(3) EDIT
               (MWDI_INPUT.SW_MODIFIERS(I).SW_POINTER,
               ' - DATA BASE IDENTIFIER COULD NOT BE ASSIGNED ON ',
               'OUTPUT FOR STATION:  ',MWDI_INPUT.AGENCY_STATION_NO)
               (A(4),A(49),A(21),A(15));
         END;
         ELSE DO;
           IF MWDI_INPUT.SW_MODIFIERS(I).SW_POINTER='STOR'
             THEN DO;
               J=J+1;
               MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(J)='STOR';
               MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(J)='    ';
               MWDI_OUTPUT.SW_DB_DATE_OF_LAST_UPDATE(J)=19000000 +
                  DATE();
             END;
           ELSE IF MWDI_INPUT.SW_MODIFIERS(I).SW_POINTER='TNRS'
             THEN DO;
               J=J+1;
               MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(J)='TNRS';
               MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(J)='    ';
               MWDI_OUTPUT.SW_DB_DATE_OF_LAST_UPDATE(J)=19000000 +
```

```
                DATE();
            END;
          ELSE IF MWDI_INPUT.SW_MODIFIERS(I).SW_POINTER='WATD'
            THEN DO;
              J=J+1;
              MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(J)='NWIS';
              MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(J)='DV  ';
              MWDI_OUTPUT.SW_DB_DATE_OF_LAST_UPDATE(J)=19000000 +
                 DATE();
            END;
          ELSE IF MWDI_INPUT.SW_MODIFIERS(I).SW_POINTER='WATG'
            THEN DO;
              J=J+1;
              MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(J)='NWIS';
              MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(J)='DESC';
              MWDI_OUTPUT.SW_DB_DATE_OF_LAST_UPDATE(J)=19000000 +
                 DATE();
            END;
          ELSE IF MWDI_INPUT.SW_MODIFIERS(I).SW_POINTER='WATP'
            THEN DO;
              J=J+1;
              MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(J)='NWIS';
              MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(J)='EE  ';
              MWDI_OUTPUT.SW_DB_DATE_OF_LAST_UPDATE(J)=19000000 +
                 DATE();
            END;
          ELSE IF MWDI_INPUT.SW_MODIFIERS(I).SW_POINTER='WATS'
            THEN DO;
              J=J+1;
              MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(J)='NWIS';
              MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(J)='    ';
              MWDI_OUTPUT.SW_DB_DATE_OF_LAST_UPDATE(J)=19000000 +
                 DATE();
            END;
          ELSE IF MWDI_INPUT.SW_MODIFIERS(I).SW_POINTER ^= '    '
            THEN PUT SKIP(3) EDIT
               (MWDI_INPUT.SW_MODIFIERS(I).SW_POINTER,
               ' - DATA BASE IDENTIFIER COULD NOT BE ASSIGNED ON ',
               'OUTPUT FOR STATION:  ',MWDI_INPUT.AGENCY_STATION_NO)
              (A(4),A(49),A(21),A(15));
        END;
    END;
  END;
IF (CONDITION_FLAG = 2 | CONDITION_FLAG = 3) & ITS_SW_DATA2
  THEN DO;
    I = 1;
    NOT_FOUND = '1'B;
    FULL = '1'B;
    DO WHILE (NOT_FOUND & I <= 4);
      IF MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(I) ^= 'NWIS' &
         MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(I) ^= 'STOR' &
         MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(I) ^= 'TNRS'
        THEN DO;
          MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(I) = 'NWIS';
```

```
          MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(I) = 'UV  ';
          MWDI_OUTPUT.SW_DB_DATE_OF_LAST_UPDATE(I) = 19000000 +
             DATE();
          FULL = '0'B;
          NOT_FOUND = '0'B;
        END;
        ELSE IF MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(I) = 'NWIS' &
          MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(I) ^= 'DV  ' &
          MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(I) ^= 'EE  ' &
          MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(I) ^= 'DESC'
          THEN DO;
            MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(I) = 'UV  ';
            MWDI_OUTPUT.SW_DB_DATE_OF_LAST_UPDATE(I) = 19000000 +
               DATE();
            FULL = '0'B;
            NOT_FOUND = '0'B;
          END;
      I = I + 1;
    END;
    IF FULL
      THEN DO I=1 TO 4;
        IF MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(I) = 'NWIS' &
           MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(I) = 'DV  '
          THEN MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(I) = 'UVDV';
          MWDI_OUTPUT.SW_DB_DATE_OF_LAST_UPDATE(I) = 19000000 +
             DATE();
      END;
  END;
END ASSIGN_SW_DB_IDENTIFIERS;
SW_FREQUENCY_COMPS:PROC;
 IF WATSTORE_UV_RECORD.PARAMETER_CODE = 65
   THEN DO;
     MWDI_OUTPUT.STAGE_TYPE = 'C ';
     MWDI_OUTPUT.STAGE_FREQ_CODE = 'C';
   END;
   ELSE IF WATSTORE_UV_RECORD.PARAMETER_CODE = 60 |
           WATSTORE_UV_RECORD.PARAMETER_CODE = 61
     THEN DO;
       MWDI_OUTPUT.FLOW_TYPE = 'C ';
       MWDI_OUTPUT.FLOW_FREQ_CODE = 'C';
     END;
     ELSE IF WATSTORE_UV_RECORD.PARAMETER_CODE = 54 |
             WATSTORE_UV_RECORD.PARAMETER_CODE = 72021 |
             WATSTORE_UV_RECORD.PARAMETER_CODE = 72022 |
             WATSTORE_UV_RECORD.PARAMETER_CODE = 72023 |
             WATSTORE_UV_RECORD.PARAMETER_CODE = 72036
       THEN DO;
         MWDI_OUTPUT.VOLUME_TYPE = 'C ';
         MWDI_OUTPUT.VOLUME_FREQ_CODE = 'C';
       END;
       ELSE IF WATSTORE_UV_RECORD.PARAMETER_CODE = 55 |
               WATSTORE_UV_RECORD.PARAMETER_CODE = 81904
         THEN DO;
           MWDI_OUTPUT.VELOCITY_TYPE = 'C ';
```

```
            MWDI_OUTPUT.VELOCITY_FREQ_CODE = 'C';
          END;
END SW_FREQUENCY_COMPS;
SW_ANN_FREQ_COMPS:PROC;
 DECLARE NUM_DAYS_IN_LAST_YEAR FIXED BIN(15) STATIC;
 IF NEW_RECORD
   THEN DO;
     SW_FREQ_HISTORY_OUTPUT.UNIQUE_SITE_ID = MWDI_OUTPUT.SITE_ID;
     SW_FREQ_HISTORY_OUTPUT.AGENCY_CODE = MWDI_OUTPUT.AGENCY_CODE;
     SW_FREQ_HISTORY_OUTPUT.AGENCY_STATION_NUMBER =
         MWDI_OUTPUT.AGENCY_STATION_NO;
     SW_FREQ_HISTORY_OUTPUT.SW_YEAR_OF_MEASUREMENT =
         WATSTORE_UV_RECORD.YEAR_NUMBER;
     SW_FREQ_HISTORY_OUTPUT.STAGE_TYPE_HISTORY = '  ';
     SW_FREQ_HISTORY_OUTPUT.STAGE_FREQ_HISTORY = ' ';
     SW_FREQ_HISTORY_OUTPUT.FLOW_TYPE_HISTORY = '  ';
     SW_FREQ_HISTORY_OUTPUT.FLOW_FREQ_HISTORY = ' ';
     SW_FREQ_HISTORY_OUTPUT.VOLUME_TYPE_HISTORY = '  ';
     SW_FREQ_HISTORY_OUTPUT.VOLUME_FREQ_HISTORY = ' ';
     SW_FREQ_HISTORY_OUTPUT.VELOCITY_TYPE_HISTORY = '  ';
     SW_FREQ_HISTORY_OUTPUT.VELOCITY_FREQ_HISTORY = ' ';
     SW_FREQ_HISTORY_OUTPUT.SW_HISTORY_DATE_OF_LAST_UPDATE = 19000000 +
        DATE();
     SW_YEAR = WATSTORE_UV_RECORD.YEAR_NUMBER;
     CURRENT_PARAMETER_CODE = WATSTORE_UV_RECORD.PARAMETER_CODE;
     LAST_PASS = '0'B;
     NEW_RECORD = '0'B;
     SW_DAY = 0;
     NUM_DAYS_IN_LAST_YEAR = 0;
     NUM_DAYS_THIS_PARM = 0;
   END;
 IF LAST_PASS
   THEN DO;
     IF NUM_DAYS_IN_LAST_YEAR > 269
       THEN MWDI_OUTPUT.SW_SEASONAL_CODE = 'Y';
       ELSE MWDI_OUTPUT.SW_SEASONAL_CODE = 'S';
     PER_STATE_NUM_FREQ_HIST_RECORDS =
        PER_STATE_NUM_FREQ_HIST_RECORDS + 1;
     WRITE FILE(SWFRQHT)  FROM(SW_FREQ_HISTORY_OUTPUT);
     NEW_RECORD = '1'B;
   END;
   ELSE DO;
     IF SW_YEAR = WATSTORE_UV_RECORD.YEAR_NUMBER
       THEN DO;
         IF CURRENT_PARAMETER_CODE = WATSTORE_UV_RECORD.PARAMETER_CODE
           THEN NUM_DAYS_THIS_PARM = NUM_DAYS_THIS_PARM + 1;
           ELSE DO;
             IF NUM_DAYS_THIS_PARM > SW_DAY
               THEN SW_DAY = NUM_DAYS_THIS_PARM;
             NUM_DAYS_THIS_PARM = 1;
             CURRENT_PARAMETER_CODE =
                 WATSTORE_UV_RECORD.PARAMETER_CODE;
           END;
       END;
```

```
  ELSE DO;
    PER_STATE_NUM_FREQ_HIST_RECORDS =
        PER_STATE_NUM_FREQ_HIST_RECORDS + 1;
    WRITE FILE(SWFRQHT)  FROM(SW_FREQ_HISTORY_OUTPUT);
    SW_FREQ_HISTORY_OUTPUT.UNIQUE_SITE_ID = MWDI_OUTPUT.SITE_ID;
    SW_FREQ_HISTORY_OUTPUT.AGENCY_CODE = MWDI_OUTPUT.AGENCY_CODE;
    SW_FREQ_HISTORY_OUTPUT.AGENCY_STATION_NUMBER =
        MWDI_OUTPUT.AGENCY_STATION_NO;
    SW_FREQ_HISTORY_OUTPUT.SW_YEAR_OF_MEASUREMENT =
        WATSTORE_UV_RECORD.YEAR_NUMBER;
    SW_FREQ_HISTORY_OUTPUT.STAGE_TYPE_HISTORY = '  ';
    SW_FREQ_HISTORY_OUTPUT.STAGE_FREQ_HISTORY = ' ';
    SW_FREQ_HISTORY_OUTPUT.FLOW_TYPE_HISTORY = '  ';
    SW_FREQ_HISTORY_OUTPUT.FLOW_FREQ_HISTORY = ' ';
    SW_FREQ_HISTORY_OUTPUT.VOLUME_TYPE_HISTORY = '  ';
    SW_FREQ_HISTORY_OUTPUT.VOLUME_FREQ_HISTORY = ' ';
    SW_FREQ_HISTORY_OUTPUT.VELOCITY_TYPE_HISTORY = '  ';
    SW_FREQ_HISTORY_OUTPUT.VELOCITY_FREQ_HISTORY = ' ';
    SW_FREQ_HISTORY_OUTPUT.SW_HISTORY_DATE_OF_LAST_UPDATE =
      19000000 + DATE();
    SW_YEAR = WATSTORE_UV_RECORD.YEAR_NUMBER;
    CURRENT_PARAMETER_CODE = WATSTORE_UV_RECORD.PARAMETER_CODE;
    IF NUM_DAYS_THIS_PARM > SW_DAY
      THEN NUM_DAYS_IN_LAST_YEAR = NUM_DAYS_THIS_PARM;
      ELSE NUM_DAYS_IN_LAST_YEAR = SW_DAY;
    SW_DAY = 1;
    NUM_DAYS_THIS_PARM = 1;
  END;
IF CURRENT_PARAMETER_CODE = 65
  THEN DO;
    SW_FREQ_HISTORY_OUTPUT.STAGE_TYPE_HISTORY = 'C ';
    SW_FREQ_HISTORY_OUTPUT.STAGE_FREQ_HISTORY = 'C';
  END;
  ELSE IF CURRENT_PARAMETER_CODE = 60 |
       CURRENT_PARAMETER_CODE = 61
    THEN DO;
      SW_FREQ_HISTORY_OUTPUT.FLOW_TYPE_HISTORY = 'C ';
      SW_FREQ_HISTORY_OUTPUT.FLOW_FREQ_HISTORY = 'C';
    END;
    ELSE IF CURRENT_PARAMETER_CODE = 54 | CURRENT_PARAMETER_CODE
          = 72021
      | CURRENT_PARAMETER_CODE = 72022 | CURRENT_PARAMETER_CODE
          = 72023
      | CURRENT_PARAMETER_CODE = 72036
      THEN DO;
        SW_FREQ_HISTORY_OUTPUT.VOLUME_TYPE_HISTORY = 'C ';
        SW_FREQ_HISTORY_OUTPUT.VOLUME_FREQ_HISTORY = 'C';
      END;
      ELSE IF CURRENT_PARAMETER_CODE = 55 |
           CURRENT_PARAMETER_CODE=81904
        THEN DO;
          SW_FREQ_HISTORY_OUTPUT.VELOCITY_TYPE_HISTORY = 'C ';
          SW_FREQ_HISTORY_OUTPUT.VELOCITY_FREQ_HISTORY = 'C';
        END;
```

```
  END;
END SW_ANN_FREQ_COMPS;
OTHER_DATA_TRANSACTIONS:PROC;
 IF OTHER_NEW_RECORD
   THEN DO;
     GW_QW_MET_TRANSACTION_OUTPUT.UNIQUE_SITE_ID = MWDI_OUTPUT.SITE_ID;
     GW_QW_MET_TRANSACTION_OUTPUT.AGENCY_CODE =
         MWDI_OUTPUT.AGENCY_CODE;
     GW_QW_MET_TRANSACTION_OUTPUT.AGENCY_STATION_NUMBER =
        MWDI_OUTPUT.AGENCY_STATION_NO;
     GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE =
        WATSTORE_UV_RECORD.PARAMETER_CODE;
     GW_QW_MET_TRANSACTION_OUTPUT.YEAR_OF_MEASUREMENT =
        WATSTORE_UV_RECORD.YEAR_NUMBER;
     GW_QW_MET_TRANSACTION_OUTPUT.FREQUENCY_CODE = 'C';
     GW_QW_MET_TRANSACTION_OUTPUT.DATA_BASE_IDENTIFIER = 'NWIS';
     GW_QW_MET_TRANSACTION_OUTPUT.DATA_BASE_SUBSET_IDENTIFIER = 'UV  ';
     OTHER_YEAR = WATSTORE_UV_RECORD.YEAR_NUMBER;
     CURRENT_PARAMETER_CODE = WATSTORE_UV_RECORD.PARAMETER_CODE;
     LAST_PASS = '0'B;
     OTHER_NEW_RECORD = '0'B;
     OTHER_RECORD_WRITTEN = '0'B;
     OTHER_DAY = 0;
   END;
 IF (CURRENT_PARAMETER_CODE = WATSTORE_UV_RECORD.PARAMETER_CODE &
    OTHER_YEAR = WATSTORE_UV_RECORD.YEAR_NUMBER)
    THEN OTHER_DAY = OTHER_DAY + 1;
    ELSE LAST_PASS = '1'B;
    IF LAST_PASS
      THEN DO;
      IF OTHER_DAY > 269
        THEN GW_QW_MET_TRANSACTION_OUTPUT.SEASONAL_CODE = 'Y';
        ELSE GW_QW_MET_TRANSACTION_OUTPUT.SEASONAL_CODE = 'S';
      PER_STATE_TOTAL_TRANSACTIONS = PER_STATE_TOTAL_TRANSACTIONS + 1;
      IF GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 72019 |
         GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 81028
        THEN PER_STATE_GW_TRANSACTIONS = PER_STATE_GW_TRANSACTIONS + 1;
        ELSE IF GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 12 |
                GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 13 |
                GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 14 |
                GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 20 |
                GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 30 |
                GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 35 |
                GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 36 |
                GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 45 |
                GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 50 |
                GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 52 |
                GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 81026 |
                GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 82300
          THEN PER_STATE_MET_TRANSACTIONS =
              PER_STATE_MET_TRANSACTIONS + 1;
          ELSE IF GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 10 |
                  GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 70 |
                  GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 76 |
```

```
                    GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 80 |
                    GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 95 |
                    GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 300 |
                    GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 400 |
                    GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE = 90095
              THEN PER_STATE_QW_TRANSACTIONS =
                   PER_STATE_QW_TRANSACTIONS + 1;
       WRITE FILE(OTRTRNS)  FROM(GW_QW_MET_TRANSACTION_OUTPUT);
       OTHER_NEW_RECORD = '1'B;
       OTHER_RECORD_WRITTEN = '1'B;
       OTHER_DAY = '0'B;
     END;
END OTHER_DATA_TRANSACTIONS;
SW_PERIOD_OF_RECORD:PROC;
 DECLARE (BEGIN_YEAR,END_YEAR,LAST_YEAR) STATIC FIXED BIN(15) INIT(0);
 IF FIRST_SW_VAL
   THEN DO;
     SW_VALIDATE_PARM = WATSTORE_UV_RECORD.PARAMETER_CODE;
     SW_VALIDATE_YEAR = WATSTORE_UV_RECORD.YEAR_NUMBER;
     FIRST_SW_VAL = '0'B;
     MWDI_OUTPUT.SW_INTERRUPTED_CODE = 'N';
     BEGIN_YEAR = MWDI_OUTPUT.SW_BEGIN_YEAR;
     IF BEGIN_YEAR > WATSTORE_UV_RECORD.YEAR_NUMBER | BEGIN_YEAR = 0
       THEN MWDI_OUTPUT.SW_BEGIN_YEAR = WATSTORE_UV_RECORD.YEAR_NUMBER;
     END_YEAR = MWDI_OUTPUT.SW_END_YEAR;
   END;
 IF WATSTORE_UV_RECORD.PARAMETER_CODE = SW_VALIDATE_PARM &
    WATSTORE_UV_RECORD.STATION_IDENTIFIER = VALIDATE_ID
   THEN IF WATSTORE_UV_RECORD.YEAR_NUMBER = SW_VALIDATE_YEAR
     THEN LAST_YEAR = SW_VALIDATE_YEAR;
     ELSE IF WATSTORE_UV_RECORD.YEAR_NUMBER = SW_VALIDATE_YEAR + 1
       THEN DO;
         LAST_YEAR = SW_VALIDATE_YEAR;
         SW_VALIDATE_YEAR = WATSTORE_UV_RECORD.YEAR_NUMBER;
       END;
       ELSE DO;
         MWDI_OUTPUT.SW_INTERRUPTED_CODE = 'Y';
         LAST_YEAR = SW_VALIDATE_YEAR;
         SW_VALIDATE_YEAR = WATSTORE_UV_RECORD.YEAR_NUMBER;
       END;
   ELSE DO;
     IF END_YEAR < LAST_YEAR
       THEN MWDI_OUTPUT.SW_END_YEAR = LAST_YEAR;
     IF MWDI_OUTPUT.SW_END_YEAR = '1984' |
        MWDI_OUTPUT.SW_END_YEAR = '1985'
       THEN DO;
         MWDI_OUTPUT.SW_END_YEAR = '0000';
         MWDI_OUTPUT.SW_ACTIVE_CODE = 'Y';
       END;
       ELSE MWDI_OUTPUT.SW_ACTIVE_CODE = 'N';
     SW_VALIDATE_YEAR = WATSTORE_UV_RECORD.YEAR_NUMBER;
     SW_VALIDATE_PARM = WATSTORE_UV_RECORD.PARAMETER_CODE;
     VALIDATE_ID = WATSTORE_UV_RECORD.STATION_IDENTIFIER;
   END;
```

```
END SW_PERIOD_OF_RECORD;
OTHER_SW_ACTIONS:PROC;
 IF MWDI_OUTPUT.SW_RECORDER_TYPE_CODE < 'A' |
    MWDI_OUTPUT.SW_RECORDER_TYPE_CODE > '9'
   THEN MWDI_OUTPUT.SW_RECORDER_TYPE_CODE = 'A';
 MWDI_OUTPUT.SW_DATA_STORAGE_MEDIA = 'C';
 MWDI_OUTPUT.SW_DATE_OF_LAST_UPDATE = 19000000+DATE();
 MWDI_OUTPUT.SW_DATA_CODE = 'Y';
END OTHER_SW_ACTIONS;
OTHER_DATA_AVAIL:PROC;
 DECLARE K FIXED DEC;
 DECLARE FOUND_5 BIT(1);
 PUT SKIP EDIT ('ENTER OTHER_DATA_AVAIL') (A(22));
 FOUND_5='0'B;
 DO K=1 TO 6;
   IF MWDI_INPUT.OTHER_DATA_AVAILABLE(K) = '5'
     THEN DO;
       K=6;
       FOUND_5 = '1'B;
     END;
 END;
 IF FOUND_5
   THEN;
   ELSE DO;
     DO K=1 TO 6;
       IF MWDI_INPUT.OTHER_DATA_AVAILABLE(K) = ' '
         THEN DO;
           MWDI_OUTPUT.OTHER_DATA_AVAILABLE(K) = '5';
           K=6;
         END;
     END;
   END;
END OTHER_DATA_AVAIL;
INDEX_NEW_SITE:PROC;
 UNIQUE_SITE_IDENTIFIER.STATE_CODE = WATSTORE_UV_RECORD.STATE_CODE;
 READ FILE(NXTAVAL) INTO (UNIQUE_SITE_IDENTIFIER)
       KEY (UNIQUE_SITE_IDENTIFIER.STATE_CODE);
      MWDI_OUTPUT.SITE_ID =
         UNIQUE_SITE_IDENTIFIER.NEXT_AVAIL_UNIQUE_SITE_ID;
      UNIQUE_SITE_IDENTIFIER.NEXT_AVAIL_UNIQUE_SITE_ID =
          UNIQUE_SITE_IDENTIFIER.NEXT_AVAIL_UNIQUE_SITE_ID + 1;
      MWDI_OUTPUT.AGENCY_CODE = WATSTORE_UV_RECORD.AGENCY_CODE;
      MWDI_OUTPUT.AGENCY_STATION_NO =
           WATSTORE_UV_RECORD.STATION_IDENTIFIER;
      MWDI_OUTPUT.SITE_TYPE = WATSTORE_UV_RECORD.SITE_CODE;
      MWDI_OUTPUT.STATE_CODE = CURRENT_STATE;
      MWDI_OUTPUT.COUNTY_CODE = WATSTORE_UV_RECORD.COUNTY_CODE;
      MWDI_OUTPUT.STATE_COUNTY_CODE = CURRENT_STATE *
            1000 + WATSTORE_UV_RECORD.COUNTY_CODE;
      MWDI_OUTPUT.DISTRICT_CODE = WATSTORE_UV_RECORD.DISTRICT_CODE;
      REWRITE FILE(NXTAVAL) FROM (UNIQUE_SITE_IDENTIFIER)
                 KEY (UNIQUE_SITE_IDENTIFIER.STATE_CODE);
END INDEX_NEW_SITE;
PRINT_CONTROL_SUBS:PROC;
```

```
DECLARE I FIXED DEC;
PUT SKIP(3) EDIT('STATE:  ',PREVIOUS_STATE) (A(8),F(2));
PUT SKIP(2) EDIT('TOTAL SITES IN:  ',PER_STATE_TOTAL_SITES_IN)
        (A(17),F(7));
PUT SKIP(2) EDIT('TOTAL SITES OUT:  ',PER_STATE_TOTAL_SITES_OUT)
                (A(18),F(7));
PUT SKIP(2) EDIT('TOTAL SITES ADDED:  ',PER_STATE_TOTAL_SITES_ADDED)
                (A(20),F(7));
PUT PAGE EDIT('DATA ELEMENT COUNTS:') (A(20));
PUT SKIP(2) EDIT('STATE:  ',PREVIOUS_STATE) (A(8),F(2));
PUT SKIP(2) EDIT('DATA ELEMENT','TOTAL IN','TOTAL OUT')
                (A(12),X(13),A(8),X(10),A(9));
PUT SKIP(2) EDIT('UNIQUE SITE IDENTIFIER',
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIQUE_SITE_IDENTIFIER,
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.UNIQUE_SITE_IDENTIFIER)
  (A(22),X(3),F(7),X(11),F(7));
PUT SKIP(2) EDIT('AGENCY STATION NUMBER',
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.AGENCY_STATION_NUMBER,
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.AGENCY_STATION_NUMBER)
  (A(21),X(4),F(7),X(11),F(7));
PUT SKIP(2) EDIT('SW ACTIVE CODE',
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_ACTIVE_CODE,
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_ACTIVE_CODE)
  (A(14),X(11),F(7),X(11),F(7));
PUT SKIP(2) EDIT('SW BEGIN YEAR',
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_BEGIN_YEAR,
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_BEGIN_YEAR)
  (A(13),X(12),F(7),X(11),F(7));
PUT SKIP(2) EDIT('SW END YEAR',
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_END_YEAR,
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_END_YEAR)
  (A(11),X(14),F(7),X(11),F(7));
PUT SKIP(2) EDIT('SW INTERRUPTED CODE',
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_INTERRUPTED_CODE,
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_INTERRUPTED_CODE)
  (A(19),X(6),F(7),X(11),F(7));
PUT SKIP(2) EDIT('UNIT FLOWS',
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_FLOWS)
  (A(10),X(15),F(7));
PUT SKIP(2) EDIT('UNIT STAGE',
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_STAGE)
  (A(10),X(15),F(7));
PUT SKIP(2) EDIT('UNIT VOLUMES',
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_VOLUME)
  (A(12),X(13),F(7));
PUT SKIP(2) EDIT('SW OTHER DATA AVAILABLE',
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_OTHER_DATA_AVAILABLE,
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.OTHER_SW_RELATED_DATA)
  (A(23),X(2),F(7),X(11),F(7));
PUT SKIP(2) EDIT('SW TELEMETRY CODE',
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_TELEMETRY_CODE,
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_TELEMETRY_CODE)
  (A(17),X(8),F(7),X(11),F(7));
PUT SKIP(2) EDIT('SW PURPOSE CODE',
```

```
    PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_PURPOSE_CODE)
    (A(15),X(10),F(7));
  PUT SKIP(2) EDIT('SW RECORDER TYPE CODE',
    PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_RECORDER_TYPE_CODE,
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_RECORDER_TYPE_CODE)
    (A(21),X(4),F(7),X(11),F(7));
  PUT SKIP(2) EDIT('STAGE TYPE',
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.STAGE_TYPE)
    (A(10),X(33),F(7));
  PUT SKIP(2) EDIT('STAGE FREQUENCY CODE',
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.STAGE_FREQ_CODE)
    (A(20),X(23),F(7));
  PUT SKIP(2) EDIT('FLOW TYPE',
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.FLOW_TYPE)
    (A(9),X(34),F(7));
  PUT SKIP(2) EDIT('FLOW FREQUENCY CODE',
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.FLOW_FREQ_CODE)
    (A(19),X(24),F(7));
  PUT SKIP(2) EDIT('VOLUME TYPE',
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_TYPE)
    (A(11),X(32),F(7));
  PUT SKIP(2) EDIT('VOLUME FREQUENCY CODE',
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_FREQ_CODE)
    (A(21),X(22),F(7));
  PUT SKIP(2) EDIT('VELOCITY TYPE',
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_TYPE)
    (A(13),X(30),F(7));
  PUT SKIP(2) EDIT('VELOCITY FREQUENCY CODE',
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_FREQ_CODE)
    (A(23),X(20),F(7));
  PUT SKIP(2) EDIT('SW DATA STORAGE MEDIA',
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_STORAGE_MEDIA)
    (A(21),X(22),F(7));
  PUT SKIP(2) EDIT('SW SEASONAL CODE',
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_SEASONAL_CODE)
    (A(16),X(27),F(7));
  PUT SKIP(2) EDIT('SW DATE OF LAST UPDATE',
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATE_OF_LAST_UPDATE)
    (A(22),X(21),F(7));
  PUT SKIP(2) EDIT('SW DATA CODE',
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_CODE)
      (A(12),X(31),F(7));
  DO I=1 TO 4;
    PUT SKIP(2) EDIT('SW DATA PURPOSE CODE(',I,')',
      PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_PURPOSE_CODE(I))
      (A(21),F(1),A(1),X(20),F(7));
  END;
  DO I=1 TO 4;
    PUT SKIP(2) EDIT('SW DATA BASE IDENTIFIER(',I,')',
      PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_IDENTIFIER(I))
      (A(24),F(1),A(1),X(17),F(7));
    PUT SKIP(2) EDIT('SW DATA BASE SUBSET IDENTIFIER(',I,')',
PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_SUBSET_IDENTIFIER(I))
      (A(31),F(1),A(1),X(10),F(7));
```

```
END;
DATA_ELEMENT_COUNTS.INPUT.UNIQUE_SITE_IDENTIFIER =
  DATA_ELEMENT_COUNTS.INPUT.UNIQUE_SITE_IDENTIFIER +
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIQUE_SITE_IDENTIFIER;
DATA_ELEMENT_COUNTS.INPUT.AGENCY_STATION_NUMBER =
  DATA_ELEMENT_COUNTS.INPUT.AGENCY_STATION_NUMBER +
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.AGENCY_STATION_NUMBER;
DATA_ELEMENT_COUNTS.INPUT.SW_ACTIVE_CODE =
  DATA_ELEMENT_COUNTS.INPUT.SW_ACTIVE_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_ACTIVE_CODE;
DATA_ELEMENT_COUNTS.INPUT.SW_BEGIN_YEAR =
  DATA_ELEMENT_COUNTS.INPUT.SW_BEGIN_YEAR +
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_BEGIN_YEAR;
DATA_ELEMENT_COUNTS.INPUT.SW_END_YEAR =
  DATA_ELEMENT_COUNTS.INPUT.SW_END_YEAR +
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_END_YEAR;
DATA_ELEMENT_COUNTS.INPUT.SW_INTERRUPTED_CODE =
  DATA_ELEMENT_COUNTS.INPUT.SW_INTERRUPTED_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_INTERRUPTED_CODE;
DATA_ELEMENT_COUNTS.INPUT.UNIT_FLOWS =
  DATA_ELEMENT_COUNTS.INPUT.UNIT_FLOWS +
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_FLOWS;
DATA_ELEMENT_COUNTS.INPUT.UNIT_STAGE =
  DATA_ELEMENT_COUNTS.INPUT.UNIT_STAGE +
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_STAGE;
DATA_ELEMENT_COUNTS.INPUT.UNIT_VOLUME =
  DATA_ELEMENT_COUNTS.INPUT.UNIT_VOLUME +
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_VOLUME;
DATA_ELEMENT_COUNTS.INPUT.SW_OTHER_DATA_AVAILABLE =
  DATA_ELEMENT_COUNTS.INPUT.SW_OTHER_DATA_AVAILABLE +
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_OTHER_DATA_AVAILABLE;
DATA_ELEMENT_COUNTS.INPUT.SW_TELEMETRY_CODE =
  DATA_ELEMENT_COUNTS.INPUT.SW_TELEMETRY_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_TELEMETRY_CODE;
DATA_ELEMENT_COUNTS.INPUT.SW_PURPOSE_CODE =
  DATA_ELEMENT_COUNTS.INPUT.SW_PURPOSE_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_PURPOSE_CODE;
DATA_ELEMENT_COUNTS.INPUT.SW_RECORDER_TYPE_CODE =
  DATA_ELEMENT_COUNTS.INPUT.SW_RECORDER_TYPE_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_RECORDER_TYPE_CODE;
DATA_ELEMENT_COUNTS.OUTPUT.UNIQUE_SITE_IDENTIFIER =
  DATA_ELEMENT_COUNTS.OUTPUT.UNIQUE_SITE_IDENTIFIER +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.UNIQUE_SITE_IDENTIFIER;
DATA_ELEMENT_COUNTS.OUTPUT.AGENCY_STATION_NUMBER =
  DATA_ELEMENT_COUNTS.OUTPUT.AGENCY_STATION_NUMBER +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.AGENCY_STATION_NUMBER;
DATA_ELEMENT_COUNTS.OUTPUT.SW_ACTIVE_CODE =
  DATA_ELEMENT_COUNTS.OUTPUT.SW_ACTIVE_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_ACTIVE_CODE;
DATA_ELEMENT_COUNTS.OUTPUT.SW_BEGIN_YEAR =
  DATA_ELEMENT_COUNTS.OUTPUT.SW_BEGIN_YEAR +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_BEGIN_YEAR;
DATA_ELEMENT_COUNTS.OUTPUT.SW_END_YEAR =
  DATA_ELEMENT_COUNTS.OUTPUT.SW_END_YEAR +
```

```
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_END_YEAR;
DATA_ELEMENT_COUNTS.OUTPUT.SW_INTERRUPTED_CODE =
  DATA_ELEMENT_COUNTS.OUTPUT.SW_INTERRUPTED_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_INTERRUPTED_CODE;
DATA_ELEMENT_COUNTS.OUTPUT.OTHER_SW_RELATED_DATA =
  DATA_ELEMENT_COUNTS.OUTPUT.OTHER_SW_RELATED_DATA +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.OTHER_SW_RELATED_DATA;
DATA_ELEMENT_COUNTS.OUTPUT.SW_TELEMETRY_CODE =
  DATA_ELEMENT_COUNTS.OUTPUT.SW_TELEMETRY_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_TELEMETRY_CODE;
DATA_ELEMENT_COUNTS.OUTPUT.SW_RECORDER_TYPE_CODE =
  DATA_ELEMENT_COUNTS.OUTPUT.SW_RECORDER_TYPE_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_RECORDER_TYPE_CODE;
DATA_ELEMENT_COUNTS.OUTPUT.STAGE_TYPE =
  DATA_ELEMENT_COUNTS.OUTPUT.STAGE_TYPE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.STAGE_TYPE;
DATA_ELEMENT_COUNTS.OUTPUT.STAGE_FREQ_CODE =
  DATA_ELEMENT_COUNTS.OUTPUT.STAGE_FREQ_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.STAGE_FREQ_CODE;
DATA_ELEMENT_COUNTS.OUTPUT.FLOW_TYPE =
  DATA_ELEMENT_COUNTS.OUTPUT.FLOW_TYPE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.FLOW_TYPE;
DATA_ELEMENT_COUNTS.OUTPUT.FLOW_FREQ_CODE =
  DATA_ELEMENT_COUNTS.OUTPUT.FLOW_FREQ_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.FLOW_FREQ_CODE;
DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_TYPE =
  DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_TYPE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_TYPE;
DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_FREQ_CODE =
  DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_FREQ_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_FREQ_CODE;
DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_TYPE =
  DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_TYPE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_TYPE;
DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_FREQ_CODE =
  DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_FREQ_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_FREQ_CODE;
DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_STORAGE_MEDIA =
  DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_STORAGE_MEDIA +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_STORAGE_MEDIA;
DATA_ELEMENT_COUNTS.OUTPUT.SW_SEASONAL_CODE =
  DATA_ELEMENT_COUNTS.OUTPUT.SW_SEASONAL_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_SEASONAL_CODE;
DATA_ELEMENT_COUNTS.OUTPUT.SW_DATE_OF_LAST_UPDATE =
  DATA_ELEMENT_COUNTS.OUTPUT.SW_DATE_OF_LAST_UPDATE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATE_OF_LAST_UPDATE;
DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_CODE =
  DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_CODE +
  PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_CODE;
DO I=1 TO 4;
  DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_PURPOSE_CODE(I) =
    DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_PURPOSE_CODE(I) +
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_PURPOSE_CODE(I);
  DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_IDENTIFIER(I) =
```

```
    DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_IDENTIFIER(I) +
    PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_IDENTIFIER(I);
  DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_SUBSET_IDENTIFIER(I) =
    DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_SUBSET_IDENTIFIER(I) +
PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_SUBSET_IDENTIFIER(I);
 END;
 PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIQUE_SITE_IDENTIFIER = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.INPUT.AGENCY_STATION_NUMBER = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_ACTIVE_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_BEGIN_YEAR = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_END_YEAR = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_INTERRUPTED_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_FLOWS = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_STAGE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_VOLUME = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_OTHER_DATA_AVAILABLE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_TELEMETRY_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_PURPOSE_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_RECORDER_TYPE_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.UNIQUE_SITE_IDENTIFIER = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.AGENCY_STATION_NUMBER = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_ACTIVE_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_BEGIN_YEAR = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_END_YEAR = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_INTERRUPTED_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.OTHER_SW_RELATED_DATA = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_TELEMETRY_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_RECORDER_TYPE_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.STAGE_TYPE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.STAGE_FREQ_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.FLOW_TYPE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.FLOW_FREQ_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_TYPE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_FREQ_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_TYPE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_FREQ_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_STORAGE_MEDIA = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_SEASONAL_CODE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATE_OF_LAST_UPDATE = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_CODE = 0;
 DO I=1 TO 4;
   PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_PURPOSE_CODE(I) = 0;
   PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_IDENTIFIER(I) = 0;
 PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_SUBSET_IDENTIFIER(I)
     = 0;
 END;
 PUT PAGE EDIT('STATE:  ',PREVIOUS_STATE) (A(8),F(2));
 PUT SKIP(2) EDIT('NUMBER OF SITES WITH FREQUENCY HISTORIES:  ',
   PER_STATE_SITES_WITH_FREQ_HIST) (A(43),F(7));
 PUT SKIP(2) EDIT('NUMBER OF FREQUENCY HISTORY RECORDS:  ',
   PER_STATE_NUM_FREQ_HIST_RECORDS) (A(38),F(7));
 PUT SKIP(5) EDIT('STATE:  ',PREVIOUS_STATE) (A(8),F(2));
 PUT SKIP(2) EDIT('NUMBER OF GROUND-WATER TRANSACTIONS:  ',
   PER_STATE_GW_TRANSACTIONS) (A(38),F(7));
```

```
 PUT SKIP(2) EDIT('NUMBER OF QUALITY-OF-WATER TRANSACTIONS:  ',
   PER_STATE_QW_TRANSACTIONS) (A(42),F(7));
 PUT SKIP(2) EDIT('NUMBER OF METEOROLOGICAL TRANSACTIONS:  ',
   PER_STATE_MET_TRANSACTIONS) (A(40),F(7));
 PUT SKIP(2) EDIT('TOTAL NUMBER OF TRANSACTIONS:  ',
   PER_STATE_TOTAL_TRANSACTIONS) (A(31),F(7));
 TOTAL_SITES_IN = TOTAL_SITES_IN + PER_STATE_TOTAL_SITES_IN;
 TOTAL_SITES_OUT = TOTAL_SITES_OUT + PER_STATE_TOTAL_SITES_OUT;
 TOTAL_SITES_ADDED = TOTAL_SITES_ADDED + PER_STATE_TOTAL_SITES_ADDED;
 SITES_WITH_FREQ_HIST = SITES_WITH_FREQ_HIST +
       PER_STATE_SITES_WITH_FREQ_HIST;
 NUM_FREQ_HIST_RECORDS = NUM_FREQ_HIST_RECORDS +
   PER_STATE_NUM_FREQ_HIST_RECORDS;
 GW_TRANSACTIONS = GW_TRANSACTIONS + PER_STATE_GW_TRANSACTIONS;
 QW_TRANSACTIONS = QW_TRANSACTIONS + PER_STATE_QW_TRANSACTIONS;
 MET_TRANSACTIONS = MET_TRANSACTIONS + PER_STATE_MET_TRANSACTIONS;
 TOTAL_TRANSACTIONS = TOTAL_TRANSACTIONS +
    PER_STATE_TOTAL_TRANSACTIONS;
 PER_STATE_TOTAL_SITES_IN = 0;
 PER_STATE_TOTAL_SITES_OUT = 0;
 PER_STATE_TOTAL_SITES_ADDED = 0;
 PER_STATE_SITES_WITH_FREQ_HIST = 0;
 PER_STATE_NUM_FREQ_HIST_RECORDS = 0;
 PER_STATE_GW_TRANSACTIONS = 0;
 PER_STATE_QW_TRANSACTIONS = 0;
 PER_STATE_MET_TRANSACTIONS = 0;
 PER_STATE_TOTAL_TRANSACTIONS = 0;
END PRINT_CONTROL_SUBS;
PRINT_CONTROL_TOTALS:PROC;
 DECLARE I FIXED DEC;
 PUT PAGE EDIT('PROCESSING COMPLETED') (A(20));
 PUT SKIP(2) EDIT('TOTAL SITES IN:  ',TOTAL_SITES_IN) (A(17),F(7));
 PUT SKIP(2) EDIT('TOTAL SITES OUT:  ',TOTAL_SITES_OUT)
                (A(18),F(7));
 PUT SKIP(2) EDIT('TOTAL SITES ADDED:  ',TOTAL_SITES_ADDED)
                (A(20),F(7));
 PUT PAGE EDIT ('PROCESSING COMPLETED') (A(20));
 PUT SKIP(2) EDIT('TOTAL DATA ELEMENT COUNTS:') (A(26));
 PUT SKIP(2) EDIT('DATA ELEMENT','TOTAL IN','TOTAL OUT')
                (A(12),X(13),A(8),X(10),A(9));
 PUT SKIP(2) EDIT('UNIQUE SITE IDENTIFIER',
   DATA_ELEMENT_COUNTS.INPUT.UNIQUE_SITE_IDENTIFIER,
   DATA_ELEMENT_COUNTS.OUTPUT.UNIQUE_SITE_IDENTIFIER)
   (A(22),X(3),F(7),X(11),F(7));
 PUT SKIP(2) EDIT('AGENCY STATION NUMBER',
   DATA_ELEMENT_COUNTS.INPUT.AGENCY_STATION_NUMBER,
   DATA_ELEMENT_COUNTS.OUTPUT.AGENCY_STATION_NUMBER)
   (A(21),X(4),F(7),X(11),F(7));
 PUT SKIP(2) EDIT('SW ACTIVE CODE',
   DATA_ELEMENT_COUNTS.INPUT.SW_ACTIVE_CODE,
   DATA_ELEMENT_COUNTS.OUTPUT.SW_ACTIVE_CODE)
   (A(14),X(11),F(7),X(11),F(7));
 PUT SKIP(2) EDIT('SW BEGIN YEAR',
   DATA_ELEMENT_COUNTS.INPUT.SW_BEGIN_YEAR,
```

```
  DATA_ELEMENT_COUNTS.OUTPUT.SW_BEGIN_YEAR)
  (A(13),X(12),F(7),X(11),F(7));
PUT SKIP(2) EDIT('SW END YEAR',
  DATA_ELEMENT_COUNTS.INPUT.SW_END_YEAR,
  DATA_ELEMENT_COUNTS.OUTPUT.SW_END_YEAR)
  (A(11),X(14),F(7),X(11),F(7));
PUT SKIP(2) EDIT('SW INTERRUPTED CODE',
  DATA_ELEMENT_COUNTS.INPUT.SW_INTERRUPTED_CODE,
  DATA_ELEMENT_COUNTS.OUTPUT.SW_INTERRUPTED_CODE)
  (A(19),X(6),F(7),X(11),F(7));
PUT SKIP(2) EDIT('UNIT FLOWS',
  DATA_ELEMENT_COUNTS.INPUT.UNIT_FLOWS)
  (A(10),X(15),F(7));
PUT SKIP(2) EDIT('UNIT STAGE',
  DATA_ELEMENT_COUNTS.INPUT.UNIT_STAGE)
  (A(10),X(15),F(7));
PUT SKIP(2) EDIT('UNIT VOLUMES',
  DATA_ELEMENT_COUNTS.INPUT.UNIT_VOLUME)
  (A(12),X(13),F(7));
PUT SKIP(2) EDIT('SW OTHER DATA AVAILABLE',
  DATA_ELEMENT_COUNTS.INPUT.SW_OTHER_DATA_AVAILABLE,
  DATA_ELEMENT_COUNTS.OUTPUT.OTHER_SW_RELATED_DATA)
  (A(23),X(2),F(7),X(11),F(7));
PUT SKIP(2) EDIT('SW TELEMETRY CODE',
  DATA_ELEMENT_COUNTS.INPUT.SW_TELEMETRY_CODE,
  DATA_ELEMENT_COUNTS.OUTPUT.SW_TELEMETRY_CODE)
  (A(17),X(8),F(7),X(11),F(7));
PUT SKIP(2) EDIT('SW PURPOSE CODE',
  DATA_ELEMENT_COUNTS.INPUT.SW_PURPOSE_CODE)
  (A(15),X(10),F(7));
PUT SKIP(2) EDIT('SW RECORDER TYPE CODE',
  DATA_ELEMENT_COUNTS.INPUT.SW_RECORDER_TYPE_CODE,
  DATA_ELEMENT_COUNTS.OUTPUT.SW_RECORDER_TYPE_CODE)
  (A(21),X(4),F(7),X(11),F(7));
PUT SKIP(2) EDIT('STAGE TYPE',
  DATA_ELEMENT_COUNTS.OUTPUT.STAGE_TYPE)
  (A(10),X(33),F(7));
PUT SKIP(2) EDIT('STAGE FREQUENCY CODE',
  DATA_ELEMENT_COUNTS.OUTPUT.STAGE_FREQ_CODE)
  (A(20),X(23),F(7));
PUT SKIP(2) EDIT('FLOW TYPE',
  DATA_ELEMENT_COUNTS.OUTPUT.FLOW_TYPE)
  (A(9),X(34),F(7));
PUT SKIP(2) EDIT('FLOW FREQUENCY CODE',
  DATA_ELEMENT_COUNTS.OUTPUT.FLOW_FREQ_CODE)
  (A(19),X(24),F(7));
PUT SKIP(2) EDIT('VOLUME TYPE',
  DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_TYPE)
  (A(11),X(32),F(7));
PUT SKIP(2) EDIT('VOLUME FREQUENCY CODE',
  DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_FREQ_CODE)
  (A(21),X(22),F(7));
PUT SKIP(2) EDIT('VELOCITY TYPE',
  DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_TYPE)
```

```
    (A(13),X(30),F(7));
  PUT SKIP(2) EDIT('VELOCITY FREQUENCY CODE',
    DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_FREQ_CODE)
    (A(23),X(20),F(7));
  PUT SKIP(2) EDIT('SW DATA STORAGE MEDIA',
    DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_STORAGE_MEDIA)
    (A(21),X(22),F(7));
  PUT SKIP(2) EDIT('SW SEASONAL CODE',
    DATA_ELEMENT_COUNTS.OUTPUT.SW_SEASONAL_CODE)
    (A(16),X(27),F(7));
  PUT SKIP(2) EDIT('SW DATE OF LAST UPDATE',
    DATA_ELEMENT_COUNTS.OUTPUT.SW_DATE_OF_LAST_UPDATE)
    (A(22),X(21),F(7));
  PUT SKIP(2) EDIT('SW DATA CODE',
    DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_CODE)
    (A(12),X(31),F(7));
  DO I=1 TO 4;
    PUT SKIP(2) EDIT ('SW DATA PURPOSE CODE(',I,')',
      DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_PURPOSE_CODE(I))
      (A(21),F(1),A(1),X(20),F(7));
  END;
  DO I=1 TO 4;
    PUT SKIP(2) EDIT('SW DATA BASE IDENTIFIER(',I,')',
      DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_IDENTIFIER(I))
      (A(24),F(1),A(1),X(17),F(7));
    PUT SKIP(2) EDIT('SW DATA BASE SUBSET IDENTIFIER(',I,')',
      DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_SUBSET_IDENTIFIER(I))
      (A(31),F(1),A(1),X(10),F(7));
  END;
  PUT PAGE EDIT ('PROCESSING COMPLETED') (A(20));
  PUT SKIP(2) EDIT('TOTAL NUMBER OF SITES WITH FREQUENCY HISTORIES:  ',
    SITES_WITH_FREQ_HIST) (A(49),F(7));
  PUT SKIP(2) EDIT('TOTAL NUMBER OF FREQUENCY HISTORY RECORDS:  ',
    NUM_FREQ_HIST_RECORDS) (A(44),F(7));
  PUT SKIP(5) EDIT('PROCESSING COMPLETED') (A(20));
  PUT SKIP(2) EDIT('TOTAL NUMBER OF GROUND-WATER TRANSACTIONS:  ',
    GW_TRANSACTIONS) (A(44),F(7));
  PUT SKIP(2) EDIT('TOTAL NUMBER OF QUALITY-OF-WATER TRANSACTIONS:  ',
    QW_TRANSACTIONS) (A(48),F(7));
  PUT SKIP(2) EDIT('TOTAL NUMBER OF METEOROLOGICAL TRANSACTIONS:  ',
    MET_TRANSACTIONS) (A(46),F(7));
  PUT SKIP(2) EDIT('TOTAL NUMBER OF TRANSACTIONS:  ',
    TOTAL_TRANSACTIONS) (A(31),F(7));
END PRINT_CONTROL_TOTALS;
PRINT_VALIDATED_SW_DATA:PROC;
  DECLARE (I,NUMBER_PRINTED) FIXED DEC;
  DECLARE (NEWMWDI_EOF,
           SWFRQHT_EOF,
           OTRTRNS_EOF) BIT(1);
  ON ENDFILE (NEWMWDI) NEWMWDI_EOF = '1'B;
  ON ENDFILE (SWFRQHT) SWFRQHT_EOF = '1'B;
  ON ENDFILE (OTRTRNS) OTRTRNS_EOF = '1'B;
  NEWMWDI_EOF = '0'B;
  SWFRQHT_EOF = '0'B;
```

```
OTRTRNS_EOF = '0'B;
CLOSE FILE(NEWMWDI);
OPEN FILE (NEWMWDI) INPUT RECORD SEQUENTIAL;
READ FILE(NEWMWDI) INTO(MWDI_OUTPUT);
NUMBER_PRINTED = 0;
DO WHILE (NUMBER_PRINTED < 100 & ^NEWMWDI_EOF);
  IF MWDI_OUTPUT.AGENCY_CODE ^= 'USGS'
    THEN DO;
      PUT PAGE EDIT('UNIQUE SITE IDENTIFIER:  ',MWDI_OUTPUT.SITE_ID)
                 (A(25),F(9));
      PUT SKIP(2) EDIT ('AGENCY CODE:  ',MWDI_OUTPUT.AGENCY_CODE)
                 (A(14),A(5));
      PUT SKIP(2) EDIT ('AGENCY STATION NUMBER:  ',
                 MWDI_OUTPUT.AGENCY_STATION_NO) (A(24),A(15));
      PUT SKIP(2) EDIT ('SITE TYPE:  ',MWDI_OUTPUT.SITE_TYPE)
           (A(12),A(2));
      PUT SKIP(2) EDIT ('STATE CODE:  ',MWDI_OUTPUT.STATE_CODE)
           (A(13),F(2));
      PUT SKIP(2) EDIT ('COUNTY CODE:  ',MWDI_OUTPUT.COUNTY_CODE)
                 (A(14),F(3));
      PUT SKIP(2) EDIT ('STATE-COUNTY CODE:  ',
           MWDI_OUTPUT.STATE_COUNTY_CODE) (A(20),F(5));
      PUT SKIP(2) EDIT ('DISTRICT CODE:  ',MWDI_OUTPUT.DISTRICT_CODE)
                 (A(16),A(2));
      PUT SKIP(2) EDIT ('OTHER DATA AVAILABLE:  ',
          MWDI_OUTPUT.OTHER_DATA_AVAILABLE(1),
          MWDI_OUTPUT.OTHER_DATA_AVAILABLE(2),
          MWDI_OUTPUT.OTHER_DATA_AVAILABLE(3),
          MWDI_OUTPUT.OTHER_DATA_AVAILABLE(4),
          MWDI_OUTPUT.OTHER_DATA_AVAILABLE(5),
          MWDI_OUTPUT.OTHER_DATA_AVAILABLE(6)) (A(23),6(A(1)));
      PUT SKIP(2) EDIT ('SW ACTIVE CODE:  ',
           MWDI_OUTPUT.SW_ACTIVE_CODE)
          (A(17),A(1));
      PUT SKIP(2) EDIT ('SW DATA CODE:  ',MWDI_OUTPUT.SW_DATA_CODE)
          (A(15),A(1));
      PUT SKIP(2) EDIT ('SW BEGIN YEAR:  ',MWDI_OUTPUT.SW_BEGIN_YEAR)
          (A(16),A(4));
      PUT SKIP(2) EDIT ('SW END YEAR:  ',MWDI_OUTPUT.SW_END_YEAR)
          (A(14),A(4));
      PUT SKIP(2) EDIT ('SW INTERRUPTED CODE:  ',
          MWDI_OUTPUT.SW_INTERRUPTED_CODE) (A(22),A(1));
      PUT SKIP(2) EDIT ('STAGE TYPE:  ',MWDI_OUTPUT.STAGE_TYPE)
           (A(13),A(2));
      PUT SKIP(2) EDIT ('STAGE FREQUENCY CODE:  ',
          MWDI_OUTPUT.STAGE_FREQ_CODE) (A(23),A(1));
      PUT SKIP(2) EDIT ('FLOW TYPE:  ',MWDI_OUTPUT.FLOW_TYPE)
            (A(12),A(2));
      PUT SKIP(2) EDIT ('FLOW FREQUENCY CODE:  ',
           MWDI_OUTPUT.FLOW_FREQ_CODE) (A(22),A(1));
      PUT SKIP(2) EDIT ('VOLUME TYPE:  ',MWDI_OUTPUT.VOLUME_TYPE)
          (A(14),A(2));
      PUT SKIP(2) EDIT ('VOLUME FREQUENCY CODE:  ',
          MWDI_OUTPUT.VOLUME_FREQ_CODE) (A(24),A(1));
```

```
      PUT SKIP(2) EDIT ('VELOCITY TYPE:  ',MWDI_OUTPUT.VELOCITY_TYPE)
          (A(16),A(2));
      PUT SKIP(2) EDIT ('VELOCITY FREQUENCY CODE:  ',
          MWDI_OUTPUT.VELOCITY_FREQ_CODE) (A(26),A(1));
      PUT SKIP(2) EDIT ('SW DATA STORAGE MEDIA:  ',
          MWDI_OUTPUT.SW_DATA_STORAGE_MEDIA) (A(24),A(1));
      PUT SKIP(2) EDIT ('SW TELEMETRY CODE:  ',
           MWDI_OUTPUT.SW_TELEMETRY_CODE) (A(20),A(1));
      PUT SKIP(2) EDIT ('SW RECORDER TYPE CODE:  ',
          MWDI_OUTPUT.SW_RECORDER_TYPE_CODE) (A(24),A(1));
      PUT SKIP(2) EDIT ('SW DATA PURPOSE CODE:  ',
          MWDI_OUTPUT.SW_DATA_PURPOSE_CODE(1),
          MWDI_OUTPUT.SW_DATA_PURPOSE_CODE(2),
          MWDI_OUTPUT.SW_DATA_PURPOSE_CODE(3),
          MWDI_OUTPUT.SW_DATA_PURPOSE_CODE(4))
          (A(23),A(1),A(1),A(1),A(1));
      PUT SKIP(2) EDIT ('OTHER SURFACE-WATER RELATED DATA:  ',
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(1),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(2),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(3),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(4),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(5),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(6),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(7),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(8),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(9),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(10),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(11),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(12))
          (A(35),12(A(1)));
      PUT SKIP(2) EDIT ('SW SEASONAL CODE:  ',
          MWDI_OUTPUT.SW_SEASONAL_CODE) (A(19),A(1));
      PUT SKIP(2) EDIT ('SW DATE OF LAST UPDATE:  ',
          MWDI_OUTPUT.SW_DATE_OF_LAST_UPDATE) (A(25),F(8));
      DO I=1 TO 4;
        PUT SKIP(2) EDIT ('SW DATA BASE IDENTIFIER(',I,'):  ',
            MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(I))
             (A(24),F(1),A(4),A(4));
        PUT SKIP(2) EDIT ('SW DATA BASE SUBSET IDENTIFIER(',I,'):  ',
            MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(I))
             (A(31),F(1),A(4),A(4));
        PUT SKIP(2) EDIT ('SW DATA BASE DATE OF LAST UPDATE(',
            I,'):  ',MWDI_OUTPUT.SW_DB_DATE_OF_LAST_UPDATE(I))
            (A(33),F(1),A(4),F(8));
      END;
      NUMBER_PRINTED=NUMBER_PRINTED + 1;
    END;
    READ FILE(NEWMWDI) INTO(MWDI_OUTPUT);
END;
CLOSE FILE(NEWMWDI);
NEWMWDI_EOF = '0'B;
OPEN FILE(NEWMWDI) INPUT RECORD SEQUENTIAL;
READ FILE(NEWMWDI) INTO(MWDI_OUTPUT);
NUMBER_PRINTED = 0;
```

```
DO WHILE (NUMBER_PRINTED < 100 & ^NEWMWDI_EOF);
  IF MWDI_OUTPUT.AGENCY_CODE = 'USGS'
    THEN DO;
      PUT PAGE EDIT('UNIQUE SITE IDENTIFIER:  ',MWDI_OUTPUT.SITE_ID)
                  (A(25),F(9));
      PUT SKIP(2) EDIT ('AGENCY CODE:  ',MWDI_OUTPUT.AGENCY_CODE)
                  (A(14),A(5));
      PUT SKIP(2) EDIT ('AGENCY STATION NUMBER:  ',
                  MWDI_OUTPUT.AGENCY_STATION_NO) (A(24),A(15));
      PUT SKIP(2) EDIT ('SITE TYPE:  ',MWDI_OUTPUT.SITE_TYPE)
          (A(12),A(2));
      PUT SKIP(2) EDIT ('STATE CODE:  ',MWDI_OUTPUT.STATE_CODE)
           (A(13),F(2));
      PUT SKIP(2) EDIT ('COUNTY CODE:  ',MWDI_OUTPUT.COUNTY_CODE)
                  (A(14),F(3));
      PUT SKIP(2) EDIT ('STATE-COUNTY CODE:  ',
          MWDI_OUTPUT.STATE_COUNTY_CODE) (A(20),F(5));
      PUT SKIP(2) EDIT ('DISTRICT CODE:  ',MWDI_OUTPUT.DISTRICT_CODE)
                  (A(16),A(2));
      PUT SKIP(2) EDIT ('OTHER DATA AVAILABLE:  ',
          MWDI_OUTPUT.OTHER_DATA_AVAILABLE(1),
          MWDI_OUTPUT.OTHER_DATA_AVAILABLE(2),
          MWDI_OUTPUT.OTHER_DATA_AVAILABLE(3),
          MWDI_OUTPUT.OTHER_DATA_AVAILABLE(4),
          MWDI_OUTPUT.OTHER_DATA_AVAILABLE(5),
          MWDI_OUTPUT.OTHER_DATA_AVAILABLE(6)) (A(23),6(A(1)));
      PUT SKIP(2) EDIT ('SW ACTIVE CODE:  ',
          MWDI_OUTPUT.SW_ACTIVE_CODE) (A(17),A(1));
      PUT SKIP(2) EDIT ('SW DATA CODE:  ',MWDI_OUTPUT.SW_DATA_CODE)
          (A(15),A(1));
      PUT SKIP(2) EDIT ('SW BEGIN YEAR:  ',MWDI_OUTPUT.SW_BEGIN_YEAR)
          (A(16),A(4));
      PUT SKIP(2) EDIT ('SW END YEAR:  ',MWDI_OUTPUT.SW_END_YEAR)
          (A(14),A(4));
      PUT SKIP(2) EDIT ('SW INTERRUPTED CODE:  ',
          MWDI_OUTPUT.SW_INTERRUPTED_CODE) (A(22),A(1));
      PUT SKIP(2) EDIT ('STAGE TYPE:  ',MWDI_OUTPUT.STAGE_TYPE)
          (A(13),A(2));
      PUT SKIP(2) EDIT ('STAGE FREQUENCY CODE:  ',
          MWDI_OUTPUT.STAGE_FREQ_CODE) (A(23),A(1));
      PUT SKIP(2) EDIT ('FLOW TYPE:  ',MWDI_OUTPUT.FLOW_TYPE)
          (A(12),A(2));
      PUT SKIP(2) EDIT ('FLOW FREQUENCY CODE:  ',
          MWDI_OUTPUT.FLOW_FREQ_CODE) (A(22),A(1));
      PUT SKIP(2) EDIT ('VOLUME TYPE:  ',MWDI_OUTPUT.VOLUME_TYPE)
          (A(14),A(2));
      PUT SKIP(2) EDIT ('VOLUME FREQUENCY CODE:  ',
          MWDI_OUTPUT.VOLUME_FREQ_CODE) (A(24),A(1));
      PUT SKIP(2) EDIT ('VELOCITY TYPE:  ',MWDI_OUTPUT.VELOCITY_TYPE)
          (A(16),A(2));
      PUT SKIP(2) EDIT ('VELOCITY FREQUENCY CODE:  ',
          MWDI_OUTPUT.VELOCITY_FREQ_CODE) (A(26),A(1));
      PUT SKIP(2) EDIT ('SW DATA STORAGE MEDIA:  ',
          MWDI_OUTPUT.SW_DATA_STORAGE_MEDIA) (A(24),A(1));
```

```
      PUT SKIP(2) EDIT ('SW TELEMETRY CODE:  ',
          MWDI_OUTPUT.SW_TELEMETRY_CODE) (A(20),A(1));
      PUT SKIP(2) EDIT ('SW RECORDER TYPE CODE:  ',
          MWDI_OUTPUT.SW_RECORDER_TYPE_CODE) (A(24),A(1));
      PUT SKIP(2) EDIT ('SW DATA PURPOSE CODE:  ',
          MWDI_OUTPUT.SW_DATA_PURPOSE_CODE(1),
          MWDI_OUTPUT.SW_DATA_PURPOSE_CODE(2),
          MWDI_OUTPUT.SW_DATA_PURPOSE_CODE(3),
          MWDI_OUTPUT.SW_DATA_PURPOSE_CODE(4))
          (A(23),A(1),A(1),A(1),A(1));
      PUT SKIP(2) EDIT ('OTHER SURFACE-WATER RELATED DATA:  ',
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(1),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(2),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(3),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(4),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(5),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(6),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(7),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(8),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(9),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(10),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(11),
          MWDI_OUTPUT.OTHER_SW_RELATED_DATA(12))
          (A(35),12(A(1)));
      PUT SKIP(2) EDIT ('SW SEASONAL CODE:  ',
          MWDI_OUTPUT.SW_SEASONAL_CODE) (A(19),A(1));
      PUT SKIP(2) EDIT ('SW DATE OF LAST UPDATE:  ',
          MWDI_OUTPUT.SW_DATE_OF_LAST_UPDATE) (A(25),F(8));
      DO I=1 TO 4;
        PUT SKIP(2) EDIT ('SW DATA BASE IDENTIFIER(',I,'):  ',
            MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(I))
            (A(24),F(1),A(4),A(4));
        PUT SKIP(2) EDIT ('SW DATA BASE SUBSET IDENTIFIER(',I,'):  ',
            MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(I))
            (A(31),F(1),A(4),A(4));
        PUT SKIP(2) EDIT ('SW DATA BASE DATE OF LAST UPDATE(',I,
            '):  ',
            MWDI_OUTPUT.SW_DB_DATE_OF_LAST_UPDATE(I))
               (A(33),F(1),A(4),F(8));
      END;
      NUMBER_PRINTED=NUMBER_PRINTED + 1;
    END;
    READ FILE(NEWMWDI) INTO(MWDI_OUTPUT);
END;
CLOSE FILE(NEWMWDI);
CLOSE FILE(SWFRQHT);
OPEN FILE(SWFRQHT) INPUT RECORD SEQUENTIAL;
I=0;
READ FILE(SWFRQHT)  INTO(SW_FREQ_HISTORY_OUTPUT);
DO WHILE (I <= 300 & ^SWFRQHT_EOF);
  PUT PAGE EDIT ('UNIQUE SITE IDENTIFIER:  ',
      SW_FREQ_HISTORY_OUTPUT.UNIQUE_SITE_ID) (A(25),F(9));
  PUT SKIP(2) EDIT ('AGENCY CODE:  ',
      SW_FREQ_HISTORY_OUTPUT.AGENCY_CODE) (A(14),A(5));
```

```
  PUT SKIP(2) EDIT ('AGENCY STATION NUMBER:  ',
      SW_FREQ_HISTORY_OUTPUT.AGENCY_STATION_NUMBER) (A(24),A(15));
  PUT SKIP(2) EDIT ('SW YEAR OF MEASUREMENT:  ',
      SW_FREQ_HISTORY_OUTPUT.SW_YEAR_OF_MEASUREMENT) (A(25),F(4));
  PUT SKIP(2) EDIT ('STAGE TYPE HISTORY:  ',
      SW_FREQ_HISTORY_OUTPUT.STAGE_TYPE_HISTORY) (A(21),A(2));
  PUT SKIP(2) EDIT ('STAGE FREQUENCY HISTORY:  ',
      SW_FREQ_HISTORY_OUTPUT.STAGE_FREQ_HISTORY) (A(26),A(1));
  PUT SKIP(2) EDIT ('FLOW TYPE HISTORY:  ',
      SW_FREQ_HISTORY_OUTPUT.FLOW_TYPE_HISTORY) (A(20),A(2));
  PUT SKIP(2) EDIT ('FLOW FREQUENCY HISTORY:  ',
      SW_FREQ_HISTORY_OUTPUT.FLOW_FREQ_HISTORY) (A(25),A(1));
  PUT SKIP(2) EDIT ('VOLUME TYPE HISTORY:  ',
      SW_FREQ_HISTORY_OUTPUT.VOLUME_TYPE_HISTORY) (A(22),A(2));
  PUT SKIP(2) EDIT ('VOLUME FREQUENCY HISTORY:  ',
      SW_FREQ_HISTORY_OUTPUT.VOLUME_FREQ_HISTORY) (A(27),A(1));
  PUT SKIP(2) EDIT ('VELOCITY TYPE HISTORY:  ',
      SW_FREQ_HISTORY_OUTPUT.VELOCITY_TYPE_HISTORY) (A(24),A(2));
  PUT SKIP(2) EDIT ('VELOCITY FREQUENCY HISTORY:  ',
      SW_FREQ_HISTORY_OUTPUT.VELOCITY_FREQ_HISTORY) (A(29),A(1));
  PUT SKIP(2) EDIT ('SW HISTORY DATE OF LAST UPDATE:  ',
      SW_FREQ_HISTORY_OUTPUT.SW_HISTORY_DATE_OF_LAST_UPDATE)
      (A(33),F(8));
  I=I+1;
  READ FILE(SWFRQHT)  INTO(SW_FREQ_HISTORY_OUTPUT);
END;
CLOSE FILE(SWFRQHT);
CLOSE FILE(OTRTRNS);
OPEN FILE(OTRTRNS) INPUT RECORD SEQUENTIAL;
I=0;
READ FILE(OTRTRNS)  INTO(GW_QW_MET_TRANSACTION_OUTPUT);
DO WHILE (I <= 200 & ^OTRTRNS_EOF);
  PUT PAGE EDIT ('UNIQUE SITE IDENTIFIER:  ',
      GW_QW_MET_TRANSACTION_OUTPUT.UNIQUE_SITE_ID) (A(25),F(9));
  PUT SKIP(2) EDIT ('AGENCY CODE:  ',
      GW_QW_MET_TRANSACTION_OUTPUT.AGENCY_CODE) (A(14),A(5));
  PUT SKIP(2) EDIT ('AGENCY STATION NUMBER:  ',
      GW_QW_MET_TRANSACTION_OUTPUT.AGENCY_STATION_NUMBER)
      (A(24),A(15));
  PUT SKIP(2) EDIT ('PARAMETER CODE:  ',
      GW_QW_MET_TRANSACTION_OUTPUT.PARAMETER_CODE) (A(17),F(5));
  PUT SKIP(2) EDIT ('YEAR OF MEASUREMENT:  ',
      GW_QW_MET_TRANSACTION_OUTPUT.YEAR_OF_MEASUREMENT) (A(22),F(4));
  PUT SKIP(2) EDIT ('FREQUENCY CODE:  ',
      GW_QW_MET_TRANSACTION_OUTPUT.FREQUENCY_CODE) (A(17),A(1));
  PUT SKIP(2) EDIT ('SEASONAL CODE:  ',
      GW_QW_MET_TRANSACTION_OUTPUT.SEASONAL_CODE) (A(16),A(1));
  PUT SKIP(2) EDIT ('DATA BASE IDENTIFIER:  ',
      GW_QW_MET_TRANSACTION_OUTPUT.DATA_BASE_IDENTIFIER) (A(23),A(4));
  PUT SKIP(2) EDIT ('DATA BASE SUBSET IDENTIFIER:  ',
      GW_QW_MET_TRANSACTION_OUTPUT.DATA_BASE_SUBSET_IDENTIFIER)
      (A(30),A(4));
  I=I+1;
  READ FILE(OTRTRNS)  INTO(GW_QW_MET_TRANSACTION_OUTPUT);
```

```
 END;
 CLOSE FILE(OTRTRNS);
END PRINT_VALIDATED_SW_DATA;
OUTPUT_COUNTS:PROC;
 DECLARE I FIXED DEC;
 DECLARE OTHER_SW_RELATED_DATA_EXISTS BIT(1);
 IF MWDI_OUTPUT.SITE_ID ^= 0
   THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.UNIQUE_SITE_IDENTIFIER =
     PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.UNIQUE_SITE_IDENTIFIER + 1;
 IF SUBSTR(MWDI_OUTPUT.AGENCY_STATION_NO,2) >= 'A' &
    SUBSTR(MWDI_OUTPUT.AGENCY_STATION_NO,2) <= '99999999999999'
   THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.AGENCY_STATION_NUMBER =
     PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.AGENCY_STATION_NUMBER + 1;
 IF MWDI_OUTPUT.SW_ACTIVE_CODE = 'Y' |
    MWDI_OUTPUT.SW_ACTIVE_CODE = 'N'
   THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_ACTIVE_CODE =
     PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_ACTIVE_CODE + 1;
 IF MWDI_OUTPUT.SW_BEGIN_YEAR ^= 0
   THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_BEGIN_YEAR =
     PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_BEGIN_YEAR + 1;
 IF MWDI_OUTPUT.SW_END_YEAR ^= 0
   THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_END_YEAR =
     PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_END_YEAR + 1;
 IF MWDI_OUTPUT.SW_INTERRUPTED_CODE = 'Y' |
    MWDI_OUTPUT.SW_INTERRUPTED_CODE = 'N'
   THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_INTERRUPTED_CODE =
     PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_INTERRUPTED_CODE + 1;
 OTHER_SW_RELATED_DATA_EXISTS = '0'B;
 DO I=1 TO 12;
   IF MWDI_OUTPUT.OTHER_SW_RELATED_DATA(I) >= 'A' &
      MWDI_OUTPUT.OTHER_SW_RELATED_DATA(I) <= '9'
     THEN OTHER_SW_RELATED_DATA_EXISTS = '1'B;
 END;
 IF (OTHER_SW_RELATED_DATA_EXISTS)
   THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.OTHER_SW_RELATED_DATA =
     PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.OTHER_SW_RELATED_DATA + 1;
 IF MWDI_OUTPUT.SW_TELEMETRY_CODE >= '0' &
    MWDI_OUTPUT.SW_TELEMETRY_CODE <= '9'
   THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_TELEMETRY_CODE =
     PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_TELEMETRY_CODE + 1;
 IF MWDI_OUTPUT.SW_RECORDER_TYPE_CODE >= 'A' &
    MWDI_OUTPUT.SW_RECORDER_TYPE_CODE <= '9'
   THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_RECORDER_TYPE_CODE =
     PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_RECORDER_TYPE_CODE + 1;
 IF MWDI_OUTPUT.STAGE_TYPE >= 'A' &
    MWDI_OUTPUT.STAGE_TYPE <= '99'
   THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.STAGE_TYPE =
     PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.STAGE_TYPE + 1;
 IF MWDI_OUTPUT.STAGE_FREQ_CODE >= 'A' &
    MWDI_OUTPUT.STAGE_FREQ_CODE <= '9'
   THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.STAGE_FREQ_CODE =
     PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.STAGE_FREQ_CODE + 1;
 IF MWDI_OUTPUT.FLOW_TYPE >= 'A' &
    MWDI_OUTPUT.FLOW_TYPE <= '99'
```

```
    THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.FLOW_TYPE =
      PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.FLOW_TYPE + 1;
  IF MWDI_OUTPUT.FLOW_FREQ_CODE >= 'A' &
     MWDI_OUTPUT.FLOW_FREQ_CODE <= '9'
    THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.FLOW_FREQ_CODE =
      PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.FLOW_FREQ_CODE + 1;
  IF MWDI_OUTPUT.VOLUME_TYPE >= 'A' &
     MWDI_OUTPUT.VOLUME_TYPE <= '99'
    THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_TYPE =
      PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_TYPE + 1;
  IF MWDI_OUTPUT.VOLUME_FREQ_CODE >= 'A' &
     MWDI_OUTPUT.VOLUME_FREQ_CODE <= '9'
    THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_FREQ_CODE =
      PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VOLUME_FREQ_CODE + 1;
  IF MWDI_OUTPUT.VELOCITY_TYPE >= 'A' &
     MWDI_OUTPUT.VELOCITY_TYPE <= '99'
    THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_TYPE =
      PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_TYPE + 1;
  IF MWDI_OUTPUT.VELOCITY_FREQ_CODE >= 'A' &
     MWDI_OUTPUT.VELOCITY_FREQ_CODE <= '9'
    THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_FREQ_CODE =
      PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.VELOCITY_FREQ_CODE + 1;
  IF MWDI_OUTPUT.SW_DATA_STORAGE_MEDIA >= 'A' &
     MWDI_OUTPUT.SW_DATA_STORAGE_MEDIA <= '9'
    THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_STORAGE_MEDIA =
      PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_STORAGE_MEDIA + 1;
  IF MWDI_OUTPUT.SW_SEASONAL_CODE >= 'A' &
     MWDI_OUTPUT.SW_SEASONAL_CODE <= '9'
    THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_SEASONAL_CODE =
      PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_SEASONAL_CODE + 1;
  IF MWDI_OUTPUT.SW_DATE_OF_LAST_UPDATE ^= 0
    THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATE_OF_LAST_UPDATE =
      PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATE_OF_LAST_UPDATE + 1;
  IF MWDI_OUTPUT.SW_DATA_CODE >= 'A' &
     MWDI_OUTPUT.SW_DATA_CODE <= '9'
    THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_CODE =
      PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_CODE + 1;
  DO I=1 TO 4;
    IF MWDI_OUTPUT.SW_DATA_PURPOSE_CODE(I) >= 'A' &
       MWDI_OUTPUT.SW_DATA_PURPOSE_CODE(I) <= '9'
     THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_PURPOSE_CODE(I) =
       PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_PURPOSE_CODE(I) + 1;
    IF MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(I) >= 'A' &
       MWDI_OUTPUT.SW_DATA_BASE_IDENTIFIER(I) <= '9999'
    THEN PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_IDENTIFIER(I)
      = PER_STATE_DATA_ELEMENT_COUNTS.OUTPUT.SW_DATA_BASE_IDENTIFIER(I)
      + 1;
    IF MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(I) >= 'A' &
       MWDI_OUTPUT.SW_DATA_BASE_SUBSET_IDENT(I) <= '9999'
    THEN PER_STATE_DATA_ELEMENT_COUNTS.SW_DATA_BASE_SUBSET_IDENTIFIER(I)
      = PER_STATE_DATA_ELEMENT_COUNTS.SW_DATA_BASE_SUBSET_IDENTIFIER(I)
      + 1;
  END;
END OUTPUT_COUNTS;
```

```
INPUT_COUNTS:PROC;
 DECLARE I FIXED DEC;
 IF MWDI_INPUT.SITE_ID ^= 0
   THEN PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIQUE_SITE_IDENTIFIER =
     PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIQUE_SITE_IDENTIFIER + 1;
 IF SUBSTR(MWDI_INPUT.AGENCY_STATION_NO,2) >= 'A' &
    SUBSTR(MWDI_INPUT.AGENCY_STATION_NO,2) <= '99999999999999'
   THEN PER_STATE_DATA_ELEMENT_COUNTS.INPUT.AGENCY_STATION_NUMBER =
     PER_STATE_DATA_ELEMENT_COUNTS.INPUT.AGENCY_STATION_NUMBER + 1;
 IF MWDI_INPUT.SW_ACTIVE_CODE ^= ''
   THEN PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_ACTIVE_CODE =
     PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_ACTIVE_CODE + 1;
 IF MWDI_INPUT.SW_BEGIN_YEAR ^= 0
   THEN PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_BEGIN_YEAR =
     PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_BEGIN_YEAR + 1;
 IF MWDI_INPUT.SW_END_YEAR ^= 0
   THEN PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_END_YEAR =
     PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_END_YEAR + 1;
 IF MWDI_INPUT.SW_INTERRUPTED_CODE ^= ''
   THEN PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_INTERRUPTED_CODE =
     PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_INTERRUPTED_CODE + 1;
 IF MWDI_INPUT.UNIT_FLOW ^= ''
   THEN PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_FLOWS =
     PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_FLOWS + 1;
 IF MWDI_INPUT.UNIT_STAGE ^= ''
   THEN PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_STAGE =
     PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_STAGE + 1;
 IF MWDI_INPUT.UNIT_VOLUME ^= ''
   THEN PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_VOLUME =
     PER_STATE_DATA_ELEMENT_COUNTS.INPUT.UNIT_VOLUME + 1;
 IF MWDI_INPUT.SW_OTHER_DATA_AVAIL ^= ''
   THEN PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_OTHER_DATA_AVAILABLE =
     PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_OTHER_DATA_AVAILABLE + 1;
 IF MWDI_INPUT.SW_TELEMETRY_CODE ^= 0
   THEN PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_TELEMETRY_CODE =
     PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_TELEMETRY_CODE + 1;
 IF MWDI_INPUT.SW_PURPOSE_CODE ^= ''
   THEN PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_PURPOSE_CODE =
     PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_PURPOSE_CODE + 1;
 IF MWDI_INPUT.SW_RECORDER_TYPE_CODE ^= ''
   THEN PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_RECORDER_TYPE_CODE =
     PER_STATE_DATA_ELEMENT_COUNTS.INPUT.SW_RECORDER_TYPE_CODE + 1;
END INPUT_COUNTS;
INPUT_SCREENING:PROC;
  DECLARE (ICOUNT,HOW_LONG) FIXED BIN(15);
  HOW_LONG = 5;
  CALL STRIP_BLANKS(MWDI_INPUT.AGENCY_CODE,HOW_LONG);
  IF MWDI_INPUT.AGENCY_CODE < 'A' | MWDI_INPUT.AGENCY_CODE > '99999'
    THEN MWDI_INPUT.AGENCY_CODE = '';
  HOW_LONG = 24;
  CALL STRIP_BLANKS(MWDI_INPUT.LOCAL_WELL_NO,HOW_LONG);
  IF MWDI_INPUT.LOCAL_WELL_NO < 'A' |
     MWDI_INPUT.LOCAL_WELL_NO > '999999999999999999999999'
    THEN MWDI_INPUT.LOCAL_WELL_NO = '';
```

```
HOW_LONG = 48;
CALL STRIP_BLANKS(MWDI_INPUT.STATION_NAME,HOW_LONG);
IF MWDI_INPUT.STATION_NAME < 'A' |
   MWDI_INPUT.STATION_NAME >
   '99999999999999999999999999999999999999999999999999'
  THEN MWDI_INPUT.STATION_NAME = '';
IF MWDI_INPUT.LAT_LONG_ACCURACY < 'A' |
   MWDI_INPUT.LAT_LONG_ACCURACY > '9'
  THEN MWDI_INPUT.LAT_LONG_ACCURACY = '';
HOW_LONG = 2;
CALL STRIP_BLANKS(MWDI_INPUT.SITE_TYPE,HOW_LONG);
IF MWDI_INPUT.SITE_TYPE < 'A' | MWDI_INPUT.SITE_TYPE > '99'
  THEN MWDI_INPUT.SITE_TYPE = '';
HOW_LONG = 2;
CALL STRIP_BLANKS(MWDI_INPUT.COUNTRY_CODE,HOW_LONG);
IF MWDI_INPUT.COUNTRY_CODE < 'A' | MWDI_INPUT.COUNTRY_CODE > '99'
  THEN MWDI_INPUT.COUNTRY_CODE = '';
HOW_LONG = 2;
CALL STRIP_BLANKS(MWDI_INPUT.DISTRICT_CODE,HOW_LONG);
IF MWDI_INPUT.DISTRICT_CODE < 'A' | MWDI_INPUT.DISTRICT_CODE > '99'
  THEN MWDI_INPUT.DISTRICT_CODE = '';
IF MWDI_INPUT.ALTITUDE_METH_OF_MEAS < 'A' |
   MWDI_INPUT.ALTITUDE_METH_OF_MEAS > '9'
  THEN MWDI_INPUT.ALTITUDE_METH_OF_MEAS = '';
IF MWDI_INPUT.SOURCE_WELL_DEPTH_DATA < 'A' |
   MWDI_INPUT.SOURCE_WELL_DEPTH_DATA > '9'
  THEN MWDI_INPUT.SOURCE_WELL_DEPTH_DATA = '';
HOW_LONG = 8;
CALL STRIP_BLANKS(MWDI_INPUT.PRINCIPAL_AQUIFER_CODE,HOW_LONG);
IF MWDI_INPUT.PRINCIPAL_AQUIFER_CODE < 'A' |
   MWDI_INPUT.PRINCIPAL_AQUIFER_CODE > '99999999'
  THEN MWDI_INPUT.PRINCIPAL_AQUIFER_CODE = '';
IF MWDI_INPUT.AQUIFER_TYPE_CODE < 'A' |
   MWDI_INPUT.AQUIFER_TYPE_CODE > '9'
  THEN MWDI_INPUT.AQUIFER_TYPE_CODE = '';
IF MWDI_INPUT.CONSOLIDATED_AQUIFER_CODE < 'A' |
   MWDI_INPUT.CONSOLIDATED_AQUIFER_CODE > '9'
  THEN MWDI_INPUT.CONSOLIDATED_AQUIFER_CODE = '';
DO ICOUNT = 1 TO 3;
  IF MWDI_INPUT.BASIN_DESCRIPTOR(ICOUNT) < 'A' |
     MWDI_INPUT.BASIN_DESCRIPTOR(ICOUNT) > '9'
    THEN MWDI_INPUT.BASIN_DESCRIPTOR(ICOUNT) = '';
  IF MWDI_INPUT.USE_OF_WATER(ICOUNT) < 'A' |
     MWDI_INPUT.USE_OF_WATER(ICOUNT) > '9'
    THEN MWDI_INPUT.USE_OF_WATER(ICOUNT) = '';
  IF MWDI_INPUT.USE_OF_SITE(ICOUNT) < 'A' |
     MWDI_INPUT.USE_OF_SITE(ICOUNT) > '9'
   THEN MWDI_INPUT.USE_OF_SITE(ICOUNT) = '';
END;
DO ICOUNT = 1 TO 6;
  IF MWDI_INPUT.OTHER_DATA_AVAILABLE(ICOUNT) < 'A' |
     MWDI_INPUT.OTHER_DATA_AVAILABLE(ICOUNT) > '9'
    THEN MWDI_INPUT.OTHER_DATA_AVAILABLE(ICOUNT) = '';
END;
```

```
 IF MWDI_INPUT.SW_ACTIVE_CODE < 'A' | MWDI_INPUT.SW_ACTIVE_CODE > '9'
   THEN MWDI_INPUT.SW_ACTIVE_CODE = '';
 IF MWDI_INPUT.GW_ACTIVE_CODE < 'A' | MWDI_INPUT.GW_ACTIVE_CODE > '9'
   THEN MWDI_INPUT.GW_ACTIVE_CODE = '';
 IF MWDI_INPUT.QW_ACTIVE_CODE < 'A' | MWDI_INPUT.QW_ACTIVE_CODE > '9'
   THEN MWDI_INPUT.QW_ACTIVE_CODE = '';
IF MWDI_INPUT.BIO_ACTIVE_CODE < 'A' | MWDI_INPUT.BIO_ACTIVE_CODE > '9'
   THEN MWDI_INPUT.BIO_ACTIVE_CODE = '';
IF MWDI_INPUT.PHY_ACTIVE_CODE < 'A' | MWDI_INPUT.PHY_ACTIVE_CODE > '9'
   THEN MWDI_INPUT.PHY_ACTIVE_CODE = '';
IF MWDI_INPUT.SED_ACTIVE_CODE < 'A' | MWDI_INPUT.SED_ACTIVE_CODE > '9'
   THEN MWDI_INPUT.SED_ACTIVE_CODE = '';
IF MWDI_INPUT.CHM_ACTIVE_CODE < 'A' | MWDI_INPUT.CHM_ACTIVE_CODE > '9'
   THEN MWDI_INPUT.CHM_ACTIVE_CODE = '';
IF MWDI_INPUT.MET_ACTIVE_CODE < 'A' | MWDI_INPUT.MET_ACTIVE_CODE > '9'
   THEN MWDI_INPUT.MET_ACTIVE_CODE = '';
 CALL STRIP_BLANKS(MWDI_INPUT.DIS_NODE_IDENT,6);
 IF MWDI_INPUT.DIS_NODE_IDENT < 'A' |
    MWDI_INPUT.DIS_NODE_IDENT > '999999'
   THEN MWDI_INPUT.DIS_NODE_IDENT = '';
 IF MWDI_INPUT.SW_DATA_CODE < 'A' | MWDI_INPUT.SW_DATA_CODE > '9'
   THEN MWDI_INPUT.SW_DATA_CODE = '';
 IF MWDI_INPUT.GW_DATA_CODE < 'A' | MWDI_INPUT.GW_DATA_CODE > '9'
   THEN MWDI_INPUT.GW_DATA_CODE = '';
 IF MWDI_INPUT.QW_DATA_CODE < 'A' | MWDI_INPUT.QW_DATA_CODE > '9'
   THEN MWDI_INPUT.QW_DATA_CODE = '';
 IF MWDI_INPUT.MET_DATA_CODE < 'A' | MWDI_INPUT.MET_DATA_CODE > '9'
   THEN MWDI_INPUT.MET_DATA_CODE = '';
IF MWDI_INPUT.OTHER_DATA_CODE < 'A' | MWDI_INPUT.OTHER_DATA_CODE > '9'
   THEN MWDI_INPUT.OTHER_DATA_CODE = '';
 IF MWDI_INPUT.SW_INTERRUPTED_CODE < 'A' |
    MWDI_INPUT.SW_INTERRUPTED_CODE > '9'
   THEN MWDI_INPUT.SW_INTERRUPTED_CODE = '';
 IF MWDI_INPUT.COMPLETE_STAGE < 'A' | MWDI_INPUT.COMPLETE_STAGE > '9'
   THEN MWDI_INPUT.COMPLETE_STAGE = '';
 IF MWDI_INPUT.PEAK_STAGE < 'A' | MWDI_INPUT.PEAK_STAGE > '9'
   THEN MWDI_INPUT.PEAK_STAGE = '';
 IF MWDI_INPUT.LOW_STAGE < 'A' | MWDI_INPUT.LOW_STAGE > '9'
   THEN MWDI_INPUT.LOW_STAGE = '';
 IF MWDI_INPUT.STAGE_STORAGE_MEDIA < 'A' |
    MWDI_INPUT.STAGE_STORAGE_MEDIA > '9'
   THEN MWDI_INPUT.STAGE_STORAGE_MEDIA = '';
 IF MWDI_INPUT.COMPLETE_FLOW < 'A' | MWDI_INPUT.COMPLETE_FLOW > '9'
   THEN MWDI_INPUT.COMPLETE_FLOW = '';
 IF MWDI_INPUT.PEAK_FLOW < 'A' | MWDI_INPUT.PEAK_FLOW > '9'
   THEN MWDI_INPUT.PEAK_FLOW = '';
 IF MWDI_INPUT.LOW_FLOW < 'A' | MWDI_INPUT.LOW_FLOW > '9'
   THEN MWDI_INPUT.LOW_FLOW = '';
 IF MWDI_INPUT.MISC_FLOW_MEAS < 'A' |
    MWDI_INPUT.MISC_FLOW_MEAS > '9'
   THEN MWDI_INPUT.MISC_FLOW_MEAS = '';
 IF MWDI_INPUT.FLOW_STORAGE_MEDIA < 'A' |
    MWDI_INPUT.FLOW_STORAGE_MEDIA > '9'
   THEN MWDI_INPUT.FLOW_STORAGE_MEDIA = '';
```

```
  IF MWDI_INPUT.VOLUME < 'A' | MWDI_INPUT.VOLUME > '9'
    THEN MWDI_INPUT.VOLUME = '';
  IF MWDI_INPUT.VOLUME_CHANGE < 'A' | MWDI_INPUT.VOLUME_CHANGE > '9'
    THEN MWDI_INPUT.VOLUME_CHANGE = '';
  IF MWDI_INPUT.VOLUME_STORAGE_MEDIA < 'A' |
     MWDI_INPUT.VOLUME_STORAGE_MEDIA > '9'
    THEN MWDI_INPUT.VOLUME_STORAGE_MEDIA = '';
  IF MWDI_INPUT.UNIT_FLOW < 'A' | MWDI_INPUT.UNIT_FLOW > '9'
     THEN MWDI_INPUT.UNIT_FLOW = '';
  IF MWDI_INPUT.UNIT_STAGE < 'A' | MWDI_INPUT.UNIT_STAGE > '9'
     THEN MWDI_INPUT.UNIT_STAGE = '';
  IF MWDI_INPUT.UNIT_VOLUME < 'A' | MWDI_INPUT.UNIT_VOLUME > '9'
     THEN MWDI_INPUT.UNIT_VOLUME = '';
  CALL STRIP_BLANKS(MWDI_INPUT.SW_OTHER_DATA_AVAIL,12);
  IF MWDI_INPUT.SW_OTHER_DATA_AVAIL < 'A' |
     MWDI_INPUT.SW_OTHER_DATA_AVAIL > '999999999999'
     THEN MWDI_INPUT.SW_OTHER_DATA_AVAIL = '';
  IF MWDI_INPUT.SW_TELEMETRY_CODE < '0' |
     MWDI_INPUT.SW_TELEMETRY_CODE > '9'
     THEN MWDI_INPUT.SW_TELEMETRY_CODE = '0';
  CALL STRIP_BLANKS(MWDI_INPUT.SW_DATE_OF_LAST_UPDATE,4);
  IF MWDI_INPUT.SW_DATE_OF_LAST_UPDATE < '0' |
     MWDI_INPUT.SW_DATE_OF_LAST_UPDATE > '9999'
     THEN MWDI_INPUT.SW_DATE_OF_LAST_UPDATE = '';
  CALL STRIP_BLANKS(MWDI_INPUT.SW_PURPOSE_CODE,9);
  IF MWDI_INPUT.SW_PURPOSE_CODE < 'A' |
     MWDI_INPUT.SW_PURPOSE_CODE > '999999999'
     THEN MWDI_INPUT.SW_PURPOSE_CODE = '';
  IF MWDI_INPUT.SW_RECORDER_TYPE_CODE < 'A' |
     MWDI_INPUT.SW_RECORDER_TYPE_CODE > '9'
     THEN MWDI_INPUT.SW_RECORDER_TYPE_CODE = '';
  IF MWDI_INPUT.SW_RECORDER_FREQ_CODE < 'A' |
     MWDI_INPUT.SW_RECORDER_FREQ_CODE > '9'
     THEN MWDI_INPUT.SW_RECORDER_FREQ_CODE = '';
  DO ICOUNT = 1 TO 10;
     CALL STRIP_BLANKS(MWDI_INPUT.SW_POINTER(ICOUNT),7);
     IF MWDI_INPUT.SW_POINTER(ICOUNT) < 'A' |
        MWDI_INPUT.SW_POINTER(ICOUNT) > '9999999'
        THEN MWDI_INPUT.SW_POINTER(ICOUNT) = '';
     IF MWDI_INPUT.SW_MOD_FILE(ICOUNT) < 'A' |
        MWDI_INPUT.SW_MOD_FILE(ICOUNT) > '9'
        THEN MWDI_INPUT.SW_MOD_FILE(ICOUNT) = '';
  END;
END INPUT_SCREENING;
STRIP_BLANKS:PROC (VARIABLE,HOW_LONG);
  DECLARE VARIABLE CHAR(*);
  DECLARE HOW_LONG FIXED BIN(15);
  DECLARE BLANKS BIT(1) INIT('1'B);
  DECLARE (ICOUNT,JCOUNT) FIXED BIN(15) INIT(1);
  DECLARE (INDEX,SUBSTR) BUILTIN;
  DO WHILE (BLANKS);
    ICOUNT = INDEX (VARIABLE,' ');
    IF ICOUNT = 1 & JCOUNT <= HOW_LONG
      THEN DO;
```

```
          VARIABLE = SUBSTR(VARIABLE,2);
          JCOUNT = JCOUNT + 1;
        END;
        ELSE BLANKS = '0'B;
    END;
  END STRIP_BLANKS;
  WATSTORE_SCREEN:PROC;
    CALL STRIP_BLANKS(WATSTORE_UV_RECORD.STATE_CODE,2);
    IF WATSTORE_UV_RECORD.STATE_CODE < '0' |
       WATSTORE_UV_RECORD.STATE_CODE > '99'
       THEN WATSTORE_UV_RECORD.STATE_CODE = '';
    CALL STRIP_BLANKS(WATSTORE_UV_RECORD.AGENCY_CODE,5);
    IF WATSTORE_UV_RECORD.AGENCY_CODE < 'A' |
       WATSTORE_UV_RECORD.AGENCY_CODE > '99999'
       THEN WATSTORE_UV_RECORD.AGENCY_CODE = '';
    CALL STRIP_BLANKS(WATSTORE_UV_RECORD.SITE_CODE,2);
    IF WATSTORE_UV_RECORD.SITE_CODE < 'A' |
       WATSTORE_UV_RECORD.SITE_CODE > '99'
       THEN WATSTORE_UV_RECORD.SITE_CODE = '';
    CALL STRIP_BLANKS(WATSTORE_UV_RECORD.DISTRICT_CODE,2);
    IF WATSTORE_UV_RECORD.DISTRICT_CODE < 'A' |
       WATSTORE_UV_RECORD.DISTRICT_CODE > '99'
       THEN WATSTORE_UV_RECORD.DISTRICT_CODE = '';
    CALL STRIP_BLANKS(WATSTORE_UV_RECORD.COUNTY_CODE,3);
    IF WATSTORE_UV_RECORD.COUNTY_CODE < '0' |
       WATSTORE_UV_RECORD.COUNTY_CODE > '999'
       THEN WATSTORE_UV_RECORD.COUNTY_CODE = '';
  END WATSTORE_SCREEN;
  END IXCVUV;
OK,
```

☆ U.S. GOVERNMENT PRINTING OFFICE: 1988 — 567-240/90001