



**Version 10**

## Chapter 6

## Appendices

**Support: 530-878-5013**  
[info@datawise.ws](mailto:info@datawise.ws)



## This Chapter's Contents

### Chapter 6

Ch. 6	<b>Contents</b>	<b>1</b>
Ch. 6	<a href="#"><u>Appendix A. Relational Database Tables</u></a>	<b>2</b>
Ch. 6	<a href="#"><u>Appendix B. Directory Structure for DataWise® Version 10 Applications</u></a>	<b>3</b>
Ch. 6	<a href="#"><u>Appendix C. Real-time Data Acquisition Programs in the Core Directory (Windows Version)</u></a>	<b>4</b>
Ch. 6	<a href="#"><u>Application Glossary</u></a>	<b>10</b>
Ch. 6	<b>Appendix D.</b>	<b>11-19</b>
Ch. 6	<a href="#"><u>Enabling the Reception of Legacy ALERT Data via an RS232 Port or an Incoming TCP/IP Socket Stream</u></a>	<b>11</b>
Ch. 6	<a href="#"><u>Enabling the Reception of Legacy ALERT Data via UDP Packets</u></a>	<b>12</b>
Ch. 6	<a href="#"><u>Enabling the Reception of Legacy ALERT Data via an Outgoing TCP/IP Socket Stream</u></a>	<b>13</b>
Ch. 6	<a href="#"><u>Enabling the Reception of ALERT2 Data</u></a>	<b>14</b>
Ch. 6	<a href="#"><u>Enabling Data Transfer From Versions of DataWise® Previous to Version 10</u></a>	<b>15</b>
Ch. 6	<a href="#"><u>Enabling GOES data collection using the DCP Data Service Protocol</u></a>	<b>16</b>
Ch. 6	<a href="#"><u>Enabling GOES data collection from an LRIT</u></a>	<b>17</b>
Ch. 6	<a href="#"><u>Enabling NTCIP Data Acquisition</u></a>	<b>18</b>
Ch. 6	<a href="#"><u>Enabling High Sierra IceSight2020E Data Acquisition</u></a>	<b>19</b>
Ch. 6	<a href="#"><u>Appendix E. Predefined Sensor Types</u></a>	<b>20-22</b>
Ch. 6	<a href="#"><u>Appendix F.</u></a>	<b>23</b>
Ch. 6	<a href="#"><u>Complete Table of Contents</u></a>	<b>24</b>
Ch. 6	<a href="#"><u>Contacts</u></a>	<b>25</b>

This manual is designed to provide the user with easy to access chapters with quick links to fulfill your information needs.



## Appendix A. Relational Database Tables

(names may be case sensitive for non-Windows installations)

Table Name	Table Contents
Alarmactions	Up to 31 alarm actions can be specified. This table contains the executable program for each action. For example, warning_email.exe, warning_control.exe, etc.
Alarms	Information on active and historical alarms. The type of alarm (e.g., water level above flood stage), the date / time of the alarm, etc.
Alarmsettings	The actual alarm settings for each sensor (e.g., max value, min value)
Availableactions	All executables that can be invoked when an alarm occurs
Categories	The categories of sensor types that can be displayed on a map (e.g., Rainfall, Water Level, Road Surface Conditions)
Cat_members	Used internally only
Currentdata	For each defined sensor, a timestamp of the most recent data, the most recent data in engineering units and, if data is received in raw form (e.g., Legacy ALERT and others), the corresponding raw data value.
DataBankCalib	Timeseries data for all defined sensors in engineering units for the most recent 45 days
DataBankCalibHist	Timeseries data for all defined sensors in engineering units for data older than 45 days
DataBankRaw	Timeseries data for all defined sensors in raw for the most recent 45 days if the data for a sensor is acquired in raw form.
DataBankRawHist	Timeseries data for all defined sensors in raw for data older than 45 days if the data for a sensor is acquired in raw form.
EmailInfo	Email / cell phone numbers to deliver alarms to.
GoesID	Meta-data about GOES DCP's to receive and decode.
ReportGroups	Sensor groupings for statistical analysis displays.
Sensors	Meta-data for each sensor being stored in the database.
SensorTypes	Meta-data for each type of sensor that can be defined
ShefGroups	SHEF product generation specifications
ShefSensors	SHEF-encoding information
SignificantInfo	Sensor benchmarks and associated colors to display on maps and charts and text when sensor readings are in this range.
SMTPInfo	SMTP server information for sending email alarms
Stations	Station ID's, types, names, and time of last communications with the station
StationType1 - StationType18	Meta-data about the various station types that can be communicated with
StatusTypes	Status bit definitions for each sensor defined as a status sensor
users	Username, password, and permission levels



## Appendix B. Directory Structure for DataWise® Version 10 Applications

All paths are relative to the install root directory.

Directory Name	Contents
AlarmIncoming	Repository for temporary files for detected alarm conditions.
AlarmReset	Alarm reset repository for temporary files for detected alarm reset conditions.
Alert2_Temp	Repository for ALERT2 pre-processed temporary files.
Config	Directory containing configuration files for this instance of DataWise.
Core	Directory containing data acquisition applications that run continuously.
DWErrors	Repository for holding errors that have occurred.
DWTemp	Directory to hold transient files that are generated by various applications. Any files older than 1 hour are deleted.
Log	Directory to hold logs of DataWise activities. Logs can be used for diagnostic purposes
LoggedMultiAlarms	Holds logs of generated alarms.
NTCIP_Incoming	This directory and the following directory hold temporary files associated with the NTCIP data acquisition modules
NTCIP_Outgoing	... see above ...
State	Used by the DataWise core applications to keep track of the state of the real-time activities.
SystemRunning	Also used by the DataWise core applications to keep track of the state of the real-time activities.
Utils	Directory containing transient data acquisition applications.
Temp	Directory to hold transient files that are generated by various applications. Any files older than 1 hour are deleted.



## Appendix C.

# Real-time Data Acquisition Programs in the Core Directory (Windows Version)

### A. DWService.exe

Function: The application that enables DataWise to run as a service. It is never executed directly but started and stopped from the Windows services.

### B. dwinstall.exe

Function: To install DWService.exe as a Windows service.

Usage: Executed from the Windows command line. Command line is:

```
dwinstall.exe -install [u=username] [p=password] [path=base] [copy=n]
```

Where:

username is the user name for the DWService.exe to login under when starting.

(Default = LocalSystem with no password)

password is the password associated with the username.

base is the base directory in which this instance of DataWise will reside.

n is the copy (or instance) number of DataWise being installed.

(Default is copy 1)

Example: Assume that the root directory of DataWise is D:\MyDirectory\MyDataWise, and it is desired to run DataWise under the LocalSystem account, the command-line would be

```
dwinstall path=D:\MyDirectory\MyDataWise
```

This will install a copy of DataWise as a service with a name of DataWiseCopy1.

When the DataWiseCopy1 is started as a service, DWService.exe will start from the D:\MyDirectory\MyDataWise\Core directory.

This function is performed automatically by the installation process.

### C. systemrunning.exe

Under Windows, this application should is never executed directly. When the instance of DataWise is started, DWService.exe will start systemrunning.exe. When the instance of DataWise is stopped, DWService.exe will stop the systemrunning.exe program.

Systemrunning.exe is responsible for starting, monitoring, and stopping all data acquisition processes. The processes that it will start are specified in file dwtask.cfg located in the Config sub-directory. The dwtask.cfg is documented below. If a process started by systemrunning.exe encounters an error and stops, systemrunning.exe will restart it. Users can write their own data acquisition processes and DataWise can manage them as it does all processes distributed by DataWise Environmental Monitoring, Inc., but applications should be written to gracefully exit upon encountering an error so that systemrunning.exe can restart it. Guidelines for writing 3rd party applications are also documented below.



#### **D. filer.exe**

This application is responsible for filing all data being received by any DataWise data acquisition process. There are no command line arguments required, however when started by systemrunning.exe, important parameters are passed which are documented in following sections. Data acquisition applications can bypass filer.exe and store data directly but at the expense of the DataWise quality control, some alarm checking, and various other internal functions used for updating web displays. Filer.exe monitors sub-directory Incoming for data to be filed. Files containing data must be formatted as follows:

If the data was received in uncalibrated format (e.g., legacy ALERT or some GOES data) sensorID idata time\_in\_seconds RAW (mm/dd/yy hh:mm:ss)

and if received already in engineering units,

sensorID rdata time\_in\_seconds CAL (mm/dd/yy hh:mm:ss)

where

- 1) sensorID is, clearly, the sensor ID (range from 1 to 2,147,483,647)?
- 2) idata is the integer data value (range from - 2,147,483,648 to 2,147,483,647)
- 3) rdata is the data in engineering units
- 4) RAW is a flag indicating the data was received in raw, uncalibrated form
- 5) CAL is a flag indicating the data was received in engineering units.

If the format of a data file encountered by filer.exe does not match either of the two formats it is discarded.

#### **E. multialarms.exe**

Multialarms.exe is responsible for detecting any sensors in alarm condition and executing the specified alarm delivery mechanism.

#### **F. alert\_operational.exe**

Alert\_operational.exe is the module used for receiving legacy ALERT data, either from a serial port or a TCP/IP socket stream or UDP datagram packet. If data is being received via UDP packets, alert\_operational.exe causes an instance of alert\_udp.exe to be executed and the two processes work in concert.

All required parameters must be read from appropriate files in the Config sub-directory and not from the MySQL database which means they must be entered with an appropriate text editor.



1) **To specify serial ports to receive legacy ALERT data from**, enter them in file alert\_ports.cfg. The format of the file is:

```
comport baudrate mappingfile validfile hsecontrols offset
```

e.g.:

```
com4 9600 No No Yes 0
```

Meaning:

comport = RS232 port on which to receive data

baudrate = baudrate

mapfile = "YES" or "NO". If "YES", info is in ..\Config\ml4.cfg (4 = com4)

validfile = "YES" or "NO". If "YES", info is in ..\Config\vl4.cfg (4 = com4)

hsecontrols = "YES" or "NO". If "YES", com port open for Read / Write, otherwise

Read offset = value to add to sensor ID

alert\_ports.cfg can contain as many lines as desired.

2) **To specify UDP ports to receive ALERT data on**, enter them in file alert\_udp\_ports.cfg as follows:

```
IP_Address Port# Label ctrlr=ControllerID
```

An example file is:

```
10.160.120.11 2101 alert ctrlr=None
10.160.120.12 2101 alert ctrlr=None
10.160.192.10 2101 alert ctrlr=None
10.160.192.11 2101 alert ctrlr=None
```

3) **To specify a TCP/IP socket stream to receive ALERT data on**, enter the information in file alert\_socket\_ports.cfg

Contents of file:

```
port mappingfile validfile offset prefix
```

eg:

```
2101 No No 0 No
```

Meaning:

port = TCP/IP Port number

mapfile = "YES" or "NO". If "YES", info is in ..Config\ms2101.cfg (2101 = port)

validfile = "YES" or "NO". If "YES", info is in ..Config\vs2101.cfg (2101 = port)

offset = value to add to sensor ID

prefix = for now, just set to "No"



### G. alert\_udp.exe

alert\_udp.exe is used to receive legacy ALERT data in the form of datagrams (UDP packets). It is never executed directly but when used is managed by alert\_operational.exe.

### H. alert2\_operational.exe

alert2\_operational.exe is used to receive and pre-process ALERT2 data. Pre-processed ALERT2 data is stored in temporary files in the Alert2\_Temp sub-directory where the final processing is performed by process\_alert2.exe. The data source must be specified on the command-line. Command-line arguments can be:

[c=comport] [b=baud] [udp=ip:port] [tcp=ip:port] [+A] [+C] [+N] [+all]

Where:

comport = the com port to receive data on

baud = optional baud rate specification (default is = 115200)

udp=ip:port specifies which IP address and port number to look for datagrams on

tcp=ip:port specifies which IP address and port number to create a socket connection on

+A instructs the software to decode ALERT2 type A messages

+C instructs the software to decode ALERT2 type C messages

+N instructs the software to decode ALERT2 type N messages

+all instructs the software to decode all ALERT2

One and only one type of data source MUST be specified. Multiple copies of alert2\_operational.exe can be started, however only one copy of process\_alert2.exe should be started.

At least one ALERT2 message type must be decoded.

### I. process\_alert2.exe

process\_alert2.exe is used for final processing of pre-processed ALERT2 data. Sub-directory Alert2\_Temp is continually monitored for incoming ALERT2 data.

### J. lrpt\_operational.exe

The purpose of this application is to connect to a computer running legacy DataWise via a socket and instruct the remote computer to forward all data that it receives.

command-line arguments are:

[[computer name] or [ip\_address]] [o=offset] [t=time\_offset]

either a computer name or ip\_address must be specified.

Where:

offset is a value to add to all sensor IDs received (default is 0)

time offset is the time (in minutes) to modify the received data by



#### **K. goes\_socket.exe**

goes\_socket.exe is used to connect to a computer that supports the DCP Data Service protocol and receive GOES DCP data, perform some initial processing and place the partially processed data files in sub-directory GOESIncoming where it is processed by process\_goes.exe.

command-line arguments are:

```
goes_socket.exe [[computer_name] or [ip_address]] port# user=username [-login] [copy=n]
```

Either "computer\_name" or "ip\_address" must be specified

port# is the TCP/IP port number

username is a valid username registered with NESDIS. Required in nearly all cases.

-login specifies that the specified computer or ip\_address does not require a username

Multiple copies of goes\_socket.exe can be started but a separate copy number must be specified for each.

#### **L. process\_goes.exe**

process\_goes.exe monitors sub-directory GOESIncoming and decodes partially processed GOES messages placed there by goes\_socket.exe or goes\_lrit.exe.

No command-line arguments are required.

#### **M. goes\_lrit.exe**

goes\_lrit.exe is used to receive data from an LRIT (Low-Rate Information Transmission System), perform some initial processing and place the partially processed data files in sub-directory GOESIncoming for continued processing by process\_goes.exe. This application is equivalent to goes\_socket.exe.

command-line arguments are:

```
goes_lrit.exe [[computer_name] or [ip_address]] port#
```

#### **N. ntcip\_direct.exe**

ntcip\_direct.exe manages traffic between DataWise and stations that use the NTCIP communication protocol.

Necessary parameters can be specified either on the command-line or in configuration files in the Config sub-directory. Usage is:

```
ntcip_direct.exe [p=comport_string] [f=comport_file] [ip=ip_string] [fip=ip_file]
```



Where:

comport\_string = ports on which data is received  
form = com1:9600,com2:19200,...  
comport\_file = file containing comports and baud rates  
form = com1:9600  
com2:19200  
ip\_string = ip address and port on which data is received  
form = 192.168.1.1:300,192.168.1.2:300,...  
ip\_file = file containing ip address and ports  
form = 192.168.1.1:300  
192.168.1.2:300

#### **O. ice\_sight2020.exe**

ice\_sight2020.exe is used to make a TCP/IP socket connection to a High Sierra Electronics IceSight2020E station and receive RWIS data sent out periodically by the unit.

Command-line usage is:

```
icesight_2020.exe s=s1,s2,..sn
```

Where:

s1,s2, ... are the station numbers to monitor  
stations must be of defined as type IceSight\_2020E

#### **P. scheduler.exe**

scheduler.exe is used to periodically execute an application that performs some pertinent function.

There are no command-line arguments. Upon startup, scheduler.exe reads file schedule.cfg in sub-directory Config.

The file contains one line for each scheduled function. The format of each line is"

```
"Name of Function to Execute" frequency offset ab1 ab2 ab3 ab4 ab5 ab6 ab7 ab8 ab9 ab10 ab11 ab12
```

Note that the name must be placed in quotes. The frequency is in minutes. The offset is in minutes past midnight. The ab1 through ab12 fields and absolute times (e.g., 0430) at which to execute a scheduled function. A value of -1 indicates a no-op.

Each line (sequentially numbered from 1) contains an batch file in the Utils sub-directory named schedule1.bat, schedule2.bat, etc.



## Application Glossary (from Specifying Data Acquisition Functionality)

A brief description of the functionality of each application follows. Further documentation is in [Appendix C](#).

<b>filer:</b>	Files incoming data into the database. Also checks for alarms based upon changes in status sensor bits.
<b>data_queue:</b>	Holds recently received data in shared memory.
<b>generate_web_data:</b>	Computes data values for map display.
<b>multialarms:</b>	Performs alarm checking and generates alarms for upper limit, lower limit, positive rate of change and negative rate of change alarms.
<b>scheduler:</b>	Executes programs at scheduled times.
<b>dwnetstat:</b>	Generates alarms when stations and sensors are not reporting at the expected frequency.
<b>qc_check:</b>	Performs quality control checks on data.

These applications alone will not acquire any data or issue commands. Additional lines must be added using a text editor to the file **dwtask.cfg** to specify the sources of data and how to acquire the data.



## Appendix D. Enabling the Reception of Legacy ALERT Data via an RS232 Port or an Incoming TCP/IP Socket Stream

Add the following line to dwtask.cfg  
D:\DW\_Flood\Core>alert\_operational.exe

Using this application, Legacy ALERT data can be received via an RS232 port or on incoming TCP/IP socket streams. The source of ALERT data is specified through files in the D:\DataWise\Config sub-directory.

**To specify serial ports on which to receive Legacy ALERT data**, enter them in file alert\_ports.cfg. The format of the file is:

```
comport baudrate mappingfile validfile hsecontrols offset
```

A sample line is:

```
com4 9600 No No Yes 0
```

where:

comport = RS232 port on which to receive data  
baudrate = baudrate  
mappingfile = "YES" or "NO". If "YES", info is in ..\Config\m14.cfg (4 = com4)  
validfile = "YES" or "NO". If "YES", info is in ..\Config\v14.cfg (4 = com4)  
hsecontrols = "YES" or "NO". If "YES", port open for RW, otherwise just R  
offset = value to add to sensor ID

The alert\_ports.cfg file can contain as many lines as required. Up to 255 comports can be specified.

**To specify an incoming TCP/IP socket stream to receive ALERT data on**, enter the information in file D:\DataWise\Config>alert\_socket\_ports.cfg. The format of the file is:

```
port mappingfile validfile offset prefix
```

An example is:

```
5001 No No 0 No
```

where:

port = TCP/IP Port number  
mapfile = "YES" or "NO". If "YES", info is in ..Config\ms2101.cfg (2101 = port)  
validfile = "YES" or "NO". If "YES", info is in ..Config\vs2101.cfg (2101 = port)  
offset = value to add to sensor ID  
prefix = for now, just set to "No"

As many lines as desired can be entered in order to receive data on multiple sockets.



## Enabling the Reception of Legacy ALERT Data via UDP Packets

Add the following line to dwtask.cfg

```
D:\DataWise\Core>alert_udp.exe
```

Using this application, Legacy ALERT data can be received via UDP packets. The source of ALERT data is specified through files in the D:\DataWise\Config sub-directory.

**To specify UDP ports to receive ALERT data on**, enter them in file alert\_udp\_ports.cfg. The format of the file is:

```
IP_Address  Port#  Label  ctrlr=ControllerID
```

An example file is:

```
10.160.120.11 2101 alert ctrlr=None
10.160.120.12 2101 alert ctrlr=None
10.160.192.10 2101 alert ctrlr=None
10.160.192.11 2101 alert ctrlr=None
```



## Enabling the Reception of Legacy ALERT Data via an Outgoing TCP/IP Socket Stream

Add the following line to dwtask.cfg  
D:\DataWise\Core>alert\_socket.exe

Using this application, Legacy ALERT data can be received via TCP/IP socket streams by specifying an IP address to connect to. IP addresses to connect to are specified through a file in the D:\DataWise\Config sub-directory.

To specify an outgoing TCP/IP socket stream to receive ALERT data on, enter the information in file D:\DataWise\Config>alert\_outgoing\_socket\_ports.cfg. The format of the file is:

```
port mappingfile validfile offset prefix
```

An example is:

```
5001 No No 0 No
```

where:

```
port      = TCP/IP Port number
mapfile   = "YES" or "NO". If "YES", info is in ..Config\ms2101.cfg (2101 = port)
validfile = "YES" or "NO". If "YES", info is in ..Config\vs2101.cfg (2101 = port)
offset    = value to add to sensor ID
prefix    = for now, just set to "No"
```



## Enabling the Reception of ALERT2 Data

Note that for data from an ALERT2 station to be decoded, that station must be properly configured. See the section on configuring ALERT2 stations. To receive ALERT2 data, add the following two lines to dwtask.cfg

```
D:\DataWise\Core>alert2_operational.exe  
D:\DataWise \Core\process_alert2.exe
```

To specify how alert2\_operational.exe should receive ALERT2 data, use the following command-line arguments:

```
[c=comport] [b=baud] [udp=ip:port] [tcp=ip:port] [+A] [+C] [+N] [+all] [copy=n]
```

Where:

comport = the com port to receive data on  
baud = optional baud rate specification (default is = 115200)  
udp = ip:port specifies which IP address and port number to look for UDP packets on  
tcp = ip:port specifies which IP address and port number to create a socket connection on  
n = copy (instance) number of the application (default is 0)

+A instructs the software to decode ALERT2 type A messages  
+C instructs the software to decode ALERT2 type C messages  
+N instructs the software to decode ALERT2 type N messages  
+all instructs the software to decode all ALERT2 messages

One and only one type of data source **MUST** be specified. Multiple copies of alert2\_operational.exe can be started, however only one copy of process\_alert2.exe should be started.

At least one ALERT2 message type must be decoded.



## Enabling Data Transfer from Versions of *DataWise*® Previous to Version 10

Add the following line to `dwtask.cfg`  
`D:\DataWise\Core\lrpt_operational.exe`  
and add the appropriate command-line argument.

Available arguments are:

`[[computer name] or [ip_address]] [o=offset] [t=time_offset]`

either a computer name or ip\_address must be specified.

Where:

offset is a value to add to all sensor IDs received (default is 0)

time offset is the time (in minutes) to modify the received data by

For example, to receive data from a computer at 192.168.1.1, add the following line to `dwtask.cfg`:

`D:\DataWise\Core\lrpt_operational.exe 192.168.1.1`



## Enabling GOES data collection using the DCP Data Service Protocol

Note that for data from a GOES DCP to be decoded, it must be properly configured. See the section on configuring GOES stations.

Add the following line to dwtask.cfg:  
D:\DataWise\Core\goes\_socket.exe  
and include the appropriate command-line arguments.

The possible command-line arguments are:

[[computer\_name] or [ip\_address]] port# user=username [-login] [copy=n]

Either "computer\_name" or "ip\_address" must be specified

port# is the TCP/IP port number

username is a valid username registered with NESDIS. Required in nearly all cases.

-login specifies that the specified computer or ip\_address does not require a username

Multiple copies of goes\_socket.exe can be started but a separate copy number must be specified for each.

If goes\_socket.exe is added to the dwtask.cfg file, the application process\_goes.exe must also be added. process\_goes.exe has no command-line arguments and is started simply by adding the line

D:\DataWise\Core\process\_goes.exe



## Enabling GOES data collection from an LRIT (Low-Rate Information Transmission System) Device

Note that for data from a GOES DCP to be decoded, it must be properly configured. See the section on configuring GOES stations.

Add the following line to dwtask.cfg

```
D:\DataWise\Core\goes_lrit.exe
```

followed by the appropriate command-line arguments:

```
[[computer_name] or [ip_address]] port#
```

As with goes\_socket.exe, if goes\_lrit.exe is added to the dwtask.cfg file, the application process\_goes.exe must also be added.

process\_goes.exe has no command-line arguments.



## Enabling NTCIP Data Acquisition

Note that for data from an NTCIP station to be acquired, it must be properly configured. See the section on configuring NTCIP station.

Add the following line to dwtask.cfg

```
D:\DataWise\Core\ntcip_direct.exe
```

ntcip\_direct.exe manages traffic between DataWise and RWIS (Road Weather Information Systems) stations that use the NTCIP communication protocol.

Necessary parameters can be specified either on the command-line or in configuration files in the Config sub-directory. Usage is:

```
[p=comport_string] [f=comport_file] [ip=ip_string] [fip=ip_file]
```

where:

```
comport_string = ports on which data is received
                 form = com1:9600,com2:19200,...
comport_file   = file containing comports and baud rates
                 form = com1:9600
                   com2:19200
ip_string      = ip address and port on which data is received
                 form = 192.168.1.1:300,192.168.1.2:300,...
ip_file        = file containing ip address and ports
                 form = 192.168.1.1:300
                   192.168.1.2:300
```

If communicating via a com port, each com port connects directly to an NTCIP station. If communicating using an IP address and port number, the IP address is associated with a station. Once properly configured, add the appropriate line to the dwtask.cfg file and start or restart DataWise.

To retrieve information from a station using the NTCIP protocol, an interrogation request must be sent to the unit. The application ntcip\_interrogate.exe is used to interrogate NTCIP stations. This application is in the Utils sub-directory. To periodically interrogate a station, use the scheduler application. The command-line to interrogate an NTCIP station is:

```
ntcip_interrogate.exe s=station# [tout=timeout]
```

Where:

station# is the number of the station to interrogate  
timeout is the time out (in seconds). Default is 5.

NTCIP stations must be properly configured before successful interrogations can occur. Read the section on station configuration.



## Enabling High Sierra IceSight2020E Data Acquisition

Note that for data from an IceSight2020E station to be decoded, it must be properly configured. See the section on configuring IceSight2020E stations.

Add the following line to dwtask.cfg

D:\DW\_Flood\Core\ice\_sight2020.exe

with the appropriate command-line parameters.

ice\_sight2020.exe is used to make a TCP/IP socket connection to a High Sierra Electronics IceSight2020E station and receive RWIS data sent out periodically by the unit.

Command-line usage is:

icesight\_2020.exe s=s1,s2,...sn

Where:

s1,s2, ... are the station numbers to monitor  
stations must be of defined as type IceSight\_2020E



## Appendix E. Predefined Sensor Types

### ALERT Sensor types (English units)

---

Water Level Float  
Water Level PT  
Tipping Bucket Pcp  
Period Precip  
Alert WindRun/Direction  
Relative Humidity  
Barometric Pressure  
Wind Speed  
Wind Direction  
Air Temperature  
Water Temperature  
Peak Wind Speed  
Battery Voltage  
Discharge  
Solar Radiation  
Sunshine Minutes  
Dew Point  
Fuel Moisture

### ALERT Sensor types (metric units)

---

Water Level Float  
Water Level PT  
Tipping Bucket Pcp  
Period Precip  
Alert WindRun/Direction  
Relative Humidity  
Barometric Pressure  
Wind Speed  
Wind Direction  
Air Temperature  
Water Temperature  
Peak Wind Speed  
Battery Voltage  
Discharge  
Solar Radiation  
Sunshine Minutes  
Dew Point  
Fuel Moisture



## High Sierra Low Water Crossing Sensor Types

---

HSE Controller  
HSE Master Status 1  
HSE Master Status 2  
HSE Remote Status 1  
HSE Remote Status 2

## High Sierra IceSight2020E Sensor Types

---

Ice Sight Displayed Condition Code  
Ice Sight Measured Condition Code  
Ice Sight Displayed Condition Mnemonic  
Ice Sight Measured Condition Mnemonic  
Ice Sight Displayed Friction Number  
Ice Sight Measured Friction Number  
Ice Sight Dirty Lens  
Ice Sight Relative Humidity  
Ice Sight Displayed Friction Code  
Ice Sight Measured Friction Code  
Ice Sight Grip Value

### Ice Sight Metric Units

---

Ice Sight Air Temperature (primary)  
Ice Sight Air Temperature (secondary)  
Ice Sight Air Temperature (tertiary)  
Ice Sight Sfc Temperature

### Ice Sight English Units

---

Ice Sight Air Temperature (primary)  
Ice Sight Air Temperature (secondary)  
Ice Sight Air Temperature (tertiary)  
Ice Sight Sfc Temperature



**NTCIP-compliant Protocol Sensor Types with both English and Metric Units**

NTCIP(e) Period Precip	NTCIP(e) SnowfallAccumRate
NTCIP(e) Surface Temp	NTCIP(e) Instant Radiation
NTCIP(e) Surface Status	NTCIP(e) SolarRadiation
NTCIP(e) RoadWetness	NTCIP(e) TotalSun
NTCIP(e) Precip Occurring	NTCIP(e) SpotWindDirection
NTCIP(e) Precip Rate	NTCIP(e) AirTemperature
NTCIP(e) Pcp Start Time	NTCIP(e) WetbulbTemp
NTCIP(e) Pavement Temp	NTCIP(e) DewpointTemp
NTCIP(e) Pcp End Time	NTCIP(e) MaxTemp
NTCIP(e) IceThickness	NTCIP(e) MinTemp
NTCIP(e) Visibility	NTCIP(e) WaterDepth
NTCIP(e) Visibility Situation	NTCIP(e) RoadwaySnowDepth
NTCIP(e) Precip Situation	NTCIP(e) SfcWaterDepth
NTCIP(e) Wind Situation	NTCIP(e) Sfc Salinity
NTCIP(e) Atmospheric Pressure	NTCIP(e) Sfc Conductivity
NTCIP(e) AvgWindDirection	NTCIP(e) Sfc Freeze Point
NTCIP(e) AvgWindSpeed	NTCIP(e) Sfc Black Ice
NTCIP(e) MaxWindGustSpeed	NTCIP(e) Sensor Error
NTCIP(e) MaxWindGustDir	NTCIP(e) SubSfc Temp
NTCIP(e) RelativeHumidity	NTCIP(e) SubSfc Moisture



## Appendix F.

### *Update in Progress*

Copyright c 2015 **DWEM (DataWise® Environmental Monitoring)**. All rights reserved.

Information in these materials is furnished for informational use only and is subject to change without notice. There is no transfer of title. **DataWise®** is a registered trademark and is subject to the contractual agreement with **DataWise® Environmental Monitoring**. Any unauthorized duplication or use of **DataWise®** in whole or in part, in print, or in any other storage and retrieval system is prohibited. No part of these materials may be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language in any form or by any means (electronic, mechanical, recording or otherwise) for any purpose other than the purchaser's personal or company use without the express written permission of **DataWise® Environmental Monitoring**. **DataWise® Environmental Monitoring** assumes no responsibility or liability for any errors or inaccuracies, which appear in these materials.



# Complete Table of Contents

<b>Chapters</b>		
<b>Ch. 1</b>	<b>Overview of <i>DataWise</i>® Version 10</b>	<b>Ch. 1 ~ 2</b>
<b>Ch. 1</b>	<b>Installation</b>	<b>Ch. 1 ~ 3-8</b>
<b>Ch. 2</b>	<b>Configure</b>	<b>Ch. 2 ~ 2-58</b>
<b>Ch. 3</b>	<b>Edit</b>	<b>Ch. 3 ~ 2-5</b>
<b>Ch. 3</b>	<b>Charts</b>	<b>Ch. 3 ~ 6-7</b>
<b>Ch. 4</b>	<b>Tubular Data</b>	<b>Ch. 4 ~ 2-9</b>
<b>Ch. 5</b>	<b>Controls</b>	<b>Ch. 5 ~ 2</b>
<b>Ch. 5</b>	<b>Hydrologic/Hydraulic Models</b>	<b>Ch. 5 ~ 3</b>
<b>Ch. 5</b>	<b>Dashboards</b>	<b>Ch. 5 ~ 4</b>
<b>Ch. 6</b>	<b>Appendices</b>	<b>Ch. 6 ~ 2-23</b>



**Support: 530-878-5013**  
[info@datawise.ws](mailto:info@datawise.ws)

**Brady R. Colton – Senior Software Developer**  
206-999-6458  
[brady@datawise.ws](mailto:brady@datawise.ws)

**Adriana Kahl – Senior Analyst**  
408-466-4257  
[adriana@datawise.ws](mailto:adriana@datawise.ws)

**Carol M. Lee**  
**CFO**  
[carol@datawise.ws](mailto:carol@datawise.ws)  
530-878-5013

**Donald E. Colton, Ph.D.**  
**Research Scientist & Sales**  
[don@datawise.ws](mailto:don@datawise.ws)  
530-277-4476

**Janice Cowden**  
**Customer Service**  
[janice@datawise.ws](mailto:janice@datawise.ws)

**International Sales**  
**Charles Luchessa**  
[charlie@datawise.ws](mailto:charlie@datawise.ws)  
530-615-1501

**Nikki Bonn**  
**Advertising**  
[nbonn@datawise.ws](mailto:nbonn@datawise.ws)

**Melody Connell**  
**Administrative Consultant**  
[melody@datawise.ws](mailto:melody@datawise.ws)

**Sally Roessler**  
**Administrative Consultant**  
[sally@datawise.ws](mailto:sally@datawise.ws)

**Website: [datawise.software](http://datawise.software)**

Filename: 6-Appendices.docx  
Folder: /Users/my\_mac/Desktop/DataWise stuff/DataWise Chapters/ready for pdf  
Template: /Users/my\_mac/Library/Group Containers/UBF8T346G9.Office/User  
Content.localized/Templates.localized/Normal.dotm  
Title:  
Subject:  
Author: Donald Colton  
Keywords:  
Comments:  
Creation Date: 5/16/16 2:42:00 PM  
Change Number: 2  
Last Saved On: 5/16/16 2:42:00 PM  
Last Saved By: Sally Roessler  
Total Editing Time: 0 Minutes  
Last Printed On: 5/16/16 2:42:00 PM  
As of Last Complete Printing  
Number of Pages: 26  
Number of Words: 4,536  
Number of Characters: 28,744 (approx.)