

L-2. how to use the sensok table

1.0 Introduction

1.1 General Description

Occasionally automated data are known to be bad, yet the values are within a reasonable range and pass any automated QC checks. Examples include tipping buckets clogged with leaves or snow, floats frozen in stilling wells or hung on the frost floor, or erroneous bubbler gage readings due to ice. All the above result in sensor reports contrary to actual conditions. In the tipping bucket example, a clog from leaves or snow will result in an under-report or zero report during the precipitation event, and false reports of precipitation during dry days. Float gages frozen in stilling wells or hung on the frost floor will not properly track the rise and fall of the water surface in the river. Ice temporarily freezing over the bubbler orifice will cause increased back-pressure and a false high reading compared to the actual water surface. The user can utilize the *sensok* table whether or not the user is taking advantage of the reasonable and gross range checking that the *shef_decode_raw* application performs. The *sensok* table allows the user to log a sensor's readings as bad during a specified period. Information stored in the *sensok* table includes a start date/time, and a comment stating the reason for the entry. The information stored in this table is utilized by the level 1 processor applications.

1.1 Table Definition

The table can be divided into 3 types of information: data point description, set good/bad characteristics and agency notification (optional) columns. The table schema is shown in Attachment A.

2.0 Table Usage

The *sensok* table allows the user to manually specify periods of bad 'raw' data. This table is read by the level 1 processor applications. Every time a data point is processed, the level 1 processor application(s) check for entries in the *sensok* table that match the lid and SHEF parameter code. The search is ordered so that the most recent entry is retrieved first. The logic is as follows:

1. No entries found:
 - a. No bad data.
2. Entries found after end of processing period:
 - a. Not valid – no effect on data within processing period.
 - b. Continue search.
3. Entries found within the processing period:

- a. First valid entry:
 - i. If 'ok' is 'Y' – data from this point to end of processing period is good.
 - ii. If 'ok' is 'N' – data from this point to end of processing period is bad.
 - iii. Continue search.
 - b. Subsequent entries:
 - i. If 'ok' is 'Y' – data from this point to previous valid entry is good.
 - ii. If 'ok' is 'N' – data from this point to previous valid entry is bad.
 - iii. Continue search.
 - c. No more entries:
 - i. Data from beginning of processing period to previous found entry is good.
4. Entries at or before processing period begins:
- a. First valid entry:
 - i. If the 'ok' entry is 'Y' – no bad data within processing period.
 - ii. If the 'ok' entry is 'N' – all data bad within processing period.
 - iii. Stop search
 - b. Subsequent entries:
 - i. If the 'ok' entry is 'Y' – data from beginning of processing period to previous found entry is good.
 - ii. If the 'ok' entry is 'N' – data from beginning of processing period to previous found entry is bad.
 - iii. Stop search.

3.0 User How-To

At this time, the only way to make an entry in the *sensok* table is manually through the *adbpg.pl* database utility or through *psql*. The database utility program *adbpg.pl* can be run through *arcmenu* or at the command line in order to access table *sensok*. For information on how to use the database utility program, *adbpg.pl*, see section J-3.

Example Usage

While viewing data with the *XDAT* application, which accesses the *IHFS* database, the following data for station *SIDM8* was noticed (Figure 1). Similar type information could have just as easily been viewed for this station using the *RFC Archive DB/Files System datview* application.

SIDM8 SIDNEY MT								
ID	PE	DUR	TS	E	OBSTIME	PRODUCT	VALUE	CHANGE
SIDM8	HG	0	RG	Z	2003-12-20 08:30:00	KWOHRRSKRF	5.92	-0.07
SIDM8	HG	0	RG	Z	2003-12-20 08:00:00	KWOHRRSKRF	5.99	-34.68
SIDM8	HG	0	RG	Z	2003-12-20 07:00:00	KWOHRRSKRF	40.67	0.04
SIDM8	HG	0	RG	Z	2003-12-20 06:00:00	KWOHRRSKRF	40.63	-0.07
SIDM8	HG	0	RG	Z	2003-12-20 05:00:00	KWOHRRSKRF	40.70	0.02
SIDM8	HG	0	RG	Z	2003-12-20 04:00:00	KWOHRRSKRF	40.68	4.13
SIDM8	HG	0	RG	Z	2003-12-20 03:00:00	KWOHRRSKRF	36.55	5.26
SIDM8	HG	0	RG	Z	2003-12-20 02:00:00	KWOHRRSKRF	31.29	5.21
SIDM8	HG	0	RG	Z	2003-12-20 01:00:00	KWOHRRSKRF	26.08	1.19
SIDM8	HG	0	RG	Z	2003-12-20 00:45:00	KWOHRRSKRF	24.89	1.34
SIDM8	HG	0	RG	Z	2003-12-20 00:30:00	KWOHRRSKRF	23.55	2.51
SIDM8	HG	0	RG	Z	2003-12-20 00:00:00	KWOHRRSKRF	21.04	4.94
SIDM8	HG	0	RG	Z	2003-12-19 23:00:00	KWOHRRSKRF	16.10	4.76
SIDM8	HG	0	RG	Z	2003-12-19 22:00:00	KWOHRRSKRF	11.34	4.68
SIDM8	HG	0	RG	Z	2003-12-19 21:00:00	KWOHRRSKRF	6.66	1.69
SIDM8	HG	0	RG	Z	2003-12-19 20:00:00	KWOHRRSKRF	4.97	-0.06
SIDM8	HG	0	RG	Z	2003-12-19 19:00:00	KWOHRRSKRF	5.03	-0.06
SIDM8	HG	0	RG	Z	2003-12-19 18:00:00	KWOHRRSKRF	5.09	-0.03
SIDM8	HG	0	RG	Z	2003-12-19 17:00:00	KWOHRRSKRF	5.12	0.00
SIDM8	HG	0	RG	Z	2003-12-19 16:45:00	KWOHRRSKRF	5.12	0.00
SIDM8	HG	0	RG	Z	2003-12-19 16:30:00	KWOHRRSKRF	5.12	0.00
SIDM8	HG	0	RG	Z	2003-12-19 16:00:00	KWOHRRSKRF	5.12	0.07
SIDM8	HG	0	RZ	Z	2003-12-19 15:10:00	KGGWRR1GGW	5.05	-0.05
SIDM8	HG	0	RG	Z	2003-12-19 15:00:00	KWOHRRSKRF	5.10	0.01
SIDM8	HG	0	RG	Z	2003-12-19 14:00:00	KWOHRRSKRF	5.09	0.02

Figure 1.

A conversation with WFO-GGW verifies the gage was ice affected for a time and that the values reported appear to be “bad” although the values passed both gross and reasonable range checks. The user could then use the previously mentioned *sensok* table and make the following entries.

**Example 1: Entry to indicate data
probably “bad” due to ice**

lid: SIDM8
pe1: H
pe2: G
dur: I
idur: 0
t: R
s: G
e: Z
p: Z
oktime: 2003-12-19 21:00:00
ok: N
init: jam
reason: ice affected
agcode:
agloc:
comment:

**Example 2: Entry to indicate data
good again**

lid: SIDM8
pe1: H
pe2: G
dur: I
idur: 0
t: R
s: G
e: Z
p: Z
oktime: 2003-12-20 08:30:00
ok: Y
init: jam
reason: looks ok
agcode:
agloc:
Comment:

Note the user in both examples chose to leave the last three fields empty.

4.0 Troubleshooting Information

Be sure to use the same lid and SHEF parameter code that is in the ‘raw’ data table, not the ‘processed’ table. The only valid values for the ‘ok’ column are “Y” and “N”. If the user continues to have problems with the table, contact the RFC Support Group.

Attachment A

column name	data type	description
<i>Data Point Descriptor Columns</i>		
lid	varchar(8)	location identifier
pe1	char(1)	1 st character of SHEF physical element code
pe2	char(1)	2 nd character of SHEF physical element code
dur	char(1)	SHEF duration code
idur	smallint	numeric duration code
t	char(1)	SHEF type code
s	char(1)	SHEF source code
e	char(1)	SHEF extremum code
p	char(1)	SHEF probability code
<i>Set bad/good characteristics columns</i>		
oktime	Timestamp without time zone	date and time of change in status
ok	char(1)	"N"– bad, "Y" - good
init	char(3)	initials of person making the entry
reason	char(80)	reason for change in status (optional)
<i>Agency notification columns</i>		
agcode	varchar(6)	abbreviation for notified agency (optional)
agloc	varchar(3)	location of notified agency (optional)
comment	varchar(30)	notification comment (optional)