

Breaking your database can be a necessary thing

Dealing with stations changing network codes

Jennifer Eakins
16 September 2019
AUG - Calgary

Scenarios

Both networks controlled within your data center:

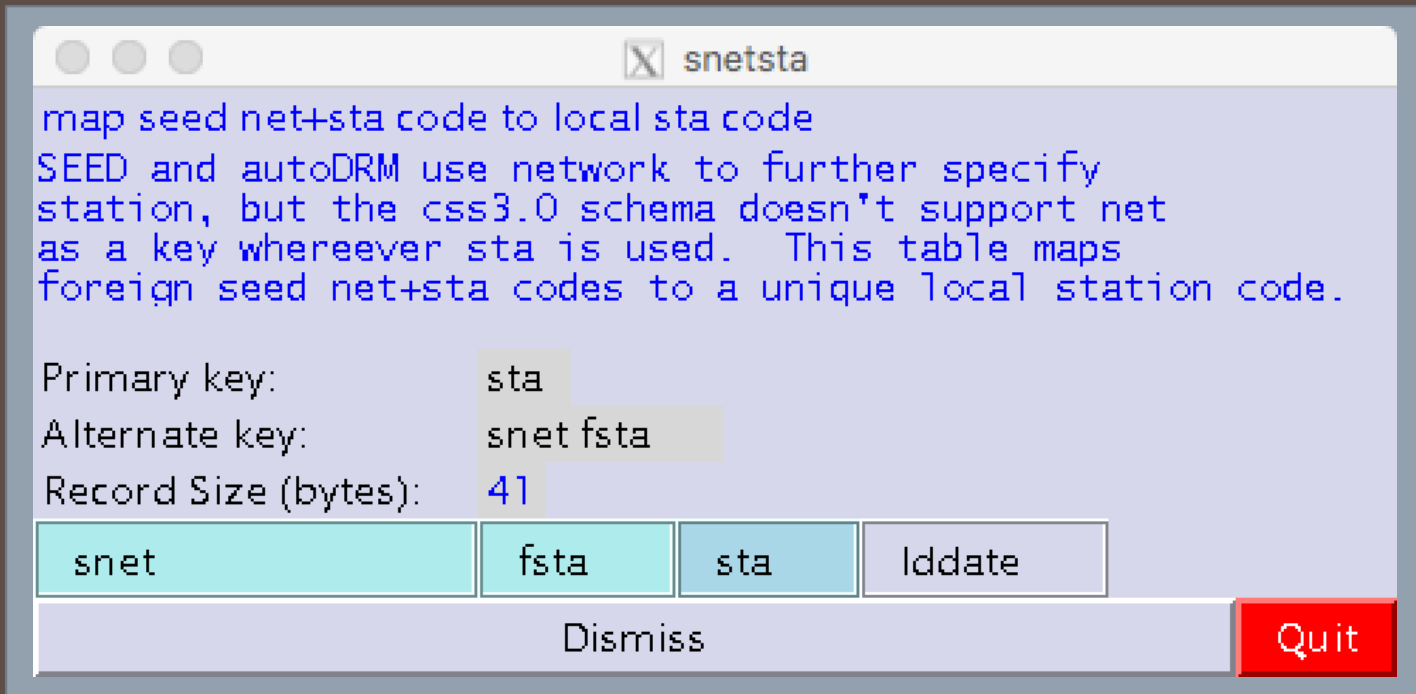
- ❖ Temporary network station moves to permanent
- ❖ Network code is being deprecated

Control moves outside of your data center:

- ❖ Transfer of operations to a new network
 - New station name along with new net code
 - Station name remains the same, but net code changes

snetsta table

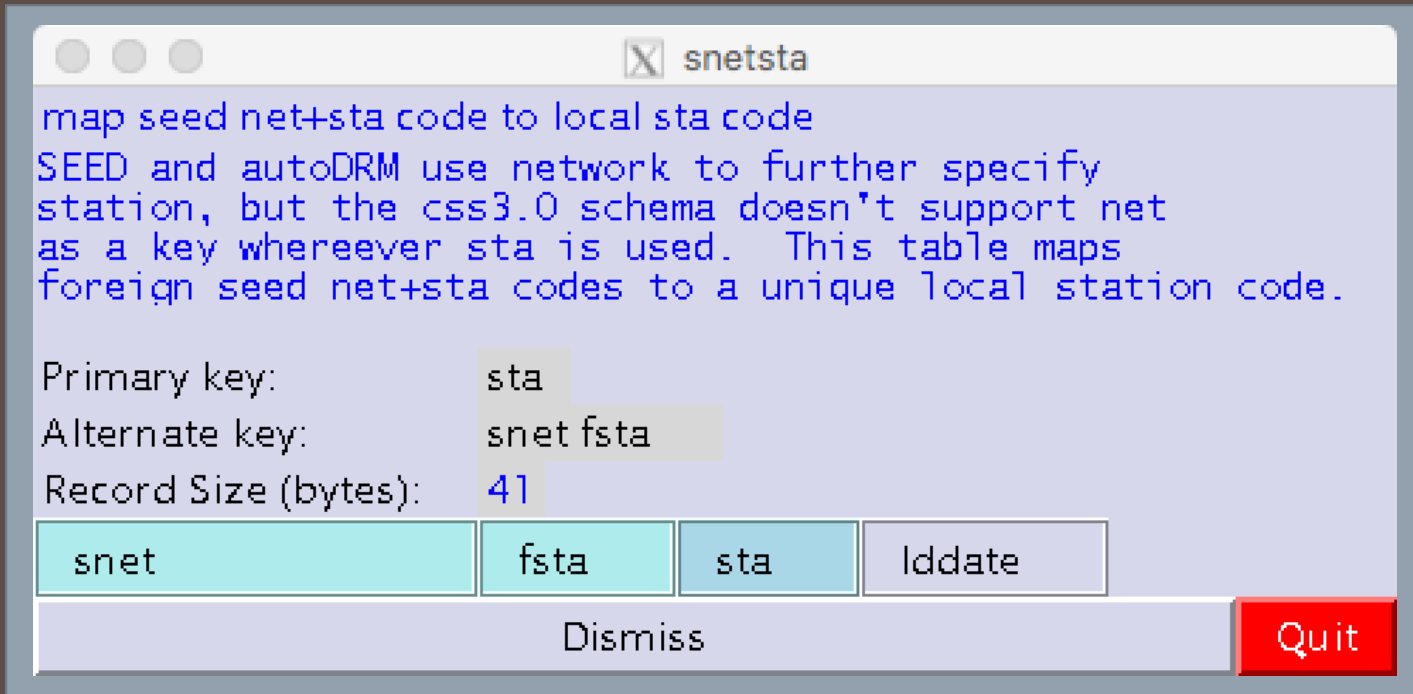
Purpose



- ❖ Only sta and chan uniquely identify stream in css3.0 and css3.1
- ❖ SEED uses net, sta, chan, and loc
- ❖ snetsta allows for mapping between them
- ❖ Programs like seed2db, orbdetect, orb2db, orb2wf, etc. consult this table for proper mapping

snetsta table

Description and limitations



- ❖ Primary key of snetsta is sta:
 - a clean database will not have two rows with the same sta field
- ❖ There is no start/end time field in snetsta

So how do you account for the new net_sta operationally?

Example

- ❖ orb contains same chan, different network
- ❖ snetsta built via metadata has only one netcode

Sources

```
Srcname
AZ_PFO/MGENC/M40
II_PFO_BH1_00/SEED
II_PFO_BH1_10/SEED
II_PFO_BH2_00/SEED
II_PFO_BH2_10/SEED
II_PFO_BHZ_00/SEED
II_PFO_BHZ_10/SEED
```

db: db/db

Tables [dropdown] New Window

lastid network schanloc site snetsta wfdisc

	snet	fsta	sta
0	AZ	PFO	PFO
1	II	PFO	PFO_II

2 Rows Table 'snetsta' updated: 9/10/19 (253) 16:41:14.68409 UTC

- ❖ snetsta table updates with new record for II net code
- ❖ netcode is appended to sta
- ❖ Waveform files will use value in sta for naming

Example

- ❖ orb contains same chan, different network
- ❖ snetsta has only one netcode

db: db/db

File Edit View Options Graphics Help

lastid network schanloc site snetsta wfdisc

	sta	chan	time	endtime	dfile
0	PFO	HHZ	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.990000 UTC	i4.PFO.HHZ.2019253_0=
1	PFO	HHN	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.990000 UTC	i4.PFO.HHN.2019253_0=
2	PFO	HHE	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.990000 UTC	i4.PFO.HHE.2019253_0=
3	PFO	HNZ	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.990000 UTC	i4.PFO.HNZ.2019253_0=
4	PFO	HNN	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.990000 UTC	i4.PFO.HNN.2019253_0=
5	PFO	HNE	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.990000 UTC	i4.PFO.HNE.2019253_0=
6	PFO	BHZ	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.975000 UTC	i4.PFO.BHZ.2019253_0=
7	PFO	BHN	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.975000 UTC	i4.PFO.BHN.2019253_0=
8	PFO	BHE	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.975000 UTC	i4.PFO.BHE.2019253_0=
9	PFO	LHZ	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.000000 UTC	i4.PFO.LHZ.2019253_0=
10	PFO	LHN	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.000000 UTC	i4.PFO.LHN.2019253_0=
11	PFO	LHE	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.000000 UTC	i4.PFO.LHE.2019253_0=
12	PFO_II	BHZ_00	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.950000 UTC	i4.PFO_II.BHZ_00.2019253_0=
13	PFO_II	BHZ_00	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.950000 UTC	i4.PFO_II.BHZ_00.2019253_0=
14	PFO_II	BHZ_10	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.975000 UTC	i4.PFO_II.BHZ_10.2019253_0=
15	PFO_II	BH1_10	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.975000 UTC	i4.PFO_II.BH1_10.2019253_0=
16	PFO_II	BH2_10	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.975000 UTC	i4.PFO_II.BH2_10.2019253_0=
17	PFO_II	BH1_00	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.950000 UTC	i4.PFO_II.BH1_00.2019253_0=
18	PFO_II	LH1_10	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.000000 UTC	i4.PFO_II.LH1_10.2019253_0=
19	PFO_II	LHZ_00	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.000000 UTC	i4.PFO_II.LHZ_00.2019253_0=
20	PFO_II	LH1_00	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.000000 UTC	i4.PFO_II.LH1_00.2019253_0=
21	PFO_II	LH2_00	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.000000 UTC	i4.PFO_II.LH2_00.2019253_0=
22	PFO_II	LH2_10	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.000000 UTC	i4.PFO_II.LH2_10.2019253_0=
23	PFO_II	LHZ_10	9/10/19 (253) 00:00:00.000000 UTC	9/10/19 (253) 23:59:59.000000 UTC	i4.PFO_II.LHZ_10.2019253_0=

24 Rows

Table 'wfdisc' updated: 9/10/19 (253) 16:51:07.46609 UTC

- ❖ Ideally, output file for new netcode would look like:
i4.PFO.BHZ_00.2019253_0=

Corrupt the dbmaster snetsta

“nodupes” snetsta

Benefit:

- ❖ Data files, arrivals, maps of stas, will continue to show a single “PFO”
- ❖ dbjoins on current data will link current metadata to ongoing arrivals
- ❖ No random mismatches attaching old/closed net code to new data

Drawbacks:

- ❖ dbjoins for data prior to the new net code will show wrong metadata
- ❖ Strange behaviors when you run seed2db for old data

dbbuild- dbmerge maintenance

Metadata build gets VERY complex!!

- Simple case: start/end times of AZ_PFO and II_PFO do not overlap!!
 - ❖ AZ - Build original network data with dbbuild
 - ❖ AZ - Delete snetsta records for AZ PFO
 - ❖ II - Build new network data (likely with miniseed2db)
 - ❖ Merge AZ and II dbs with dbmerge
- Headaches begin: start/end times overlap
 - ❖ AZ snetsta needs AZ PFO removed
 - ❖ Edits to site startdate/enddate
 - ❖ Exclusions of snetsta in dbmerge
 - ❖ Cat-ting of snetsta tables to make something with nodupes