

Miscellaneous Tuning

Key Files, Inventory Files

Your important configuration files in SeisComP3 are in

ls seiscomp3/etc

```
defaults    global.cfg  inventory  key  
scmaster.cfg  descriptions  init      kernel.cfg  
scautopick.cfg  scrttpv.cfg
```

User profiles and logs are hidden in .seiscomp3

Duplicate Configuration

On original system

```
cd ~
```

```
tar czf original_etc.tgz seiscomp3/etc
```

On new system

```
cd ~
```

```
tar czf new_system_etc_backup20170824.tgz  
seiscomp3/etc
```

```
tar xzf original_etc.tgz
```

```
#THIS WILL ERASE EXISTING FILES IN  
seiscomp3/etc and replace them which is why  
we made the backup first!
```

Keep Only Streams We Want

Edit your seedlink profile

Delete the IRIS “chain” and add 4 new chains

IRIS_HH__ use selector: ??HH?.D

IRIS_HH use selector: HH?.D

IRIS_BH__ use selector: ??BH?.D

IRIS_BH use selector: BH?.D

All connect to rtserve.iris.washington.edu

Example:

The screenshot shows a configuration window for a source named "IRIS_HH_ : chain". The window has a title bar with a close button. Below the title bar, there are four input fields arranged in a 2x2 grid, each with a lock icon to its right. The fields are:

- address**: Contains "rtserve.iris.washington.edu". Below it is the text "Hostname or IP of the Seedlink server."
- port**: Contains "18000". Below it is the text "Port of the Seedlink server"
- selectors**: Contains "??HH?.D". Below it is the text "List of stream selectors. If left empty all available ..."
- proc**: Is empty. Below it is the text "Name of the proc object (defined in streams.xml); used ..."

On the right side of the window, there is a file explorer showing a folder named "slink_IRI".

Add those; Save; System Tab: Run update-config just on seedlinnk, and then restart seedlink



Import MiniSEED

you can import into the slarchive/ arclink
directory archive with

```
seiscomp exec scart -I ~/filename.mseed ~/
seiscomp3/var/lib/archive
```



Tuning/ Playback

msrtsimul !

You need a multiplexed miniseed file. You can get this from your slarchive via the following command:

```
seiscomp exec scart -dsvE -t "2017-08-24  
14:55~ 2017-08-24 15:00" ~/seiscomp3/  
var/lib/archive/ > example.sorted.mseed
```

Tuning/ Playback

If you don't have it in your slarchive:

scmssort is a tool that can be used to sort data.

Download data

Concatenate it with the unix 'cat' command, pipe to scmssort.

```
cat f1.mseed f2.mseed f3.mseed | scmssort  
-v -t '2007-03-28 15:48~2007-03-28 16:18'  
> sorted.mseed
```


Database management

scdbstrip

Database clean-up of processing results.

Description

SeisComP3s processing is continuously writing to the database. This causes the database to grow and to occupy much space on the harddisc. scdbstrip taggles this problem and removed processed objects from the database older than a configurable time span.

This clean-up procedure is based on events. scdbstrip will remove all events with an origin time older than specified. It will also remove all associated objects such as picks, origins, arrivals, amplitudes and so on.

scdbstrip does not run as a daemon. To remove old objects continuously scdbstrip should be added to the list of cronjobs running every e.g. 30 minutes. The more often it runs the less objects it has to remove and the faster it will unlock the database again.



Inventory Management

We already mentioned this one, but it is useful particularly if you have metadata conflicts.

invextr

Extracts channels from inventory.

Description

invextr extracts or removes networks, stations and channels from an inventory XML file. It takes basically three important parameters:

1. channel ID list
2. input file
3. output file

whereas the output file defaults to stdout and the input file to stdin if not given.

A channel ID is a simple string that is matched against the final channel ID in the inventory. This final channel ID is constructed by joining the codes of all stages with a dot where the stages are network, station, location and channel.

Seiscomp Automatic Start

seiscomp print crontab

```
sysop@ubuntu14:~/seiscomp3$ seiscomp print crontab
*/3 * * * * /home/sysop/seiscomp3/bin/seiscomp check >/dev/null 2>&1
55 23 * * * /home/sysop/seiscomp3/var/lib/seedlink/backup_seqfiles >/dev/null 2>
&1
20 3 * * * /home/sysop/seiscomp3/var/lib/slarchive/purge_datafiles >/dev/null 2>
&1
sysop@ubuntu14:~/seiscomp3$
```

This seiscomp check will constantly monitor for stopped modules that are enabled, and try to start them. It will also start SeisComP3 on boot of your computer.

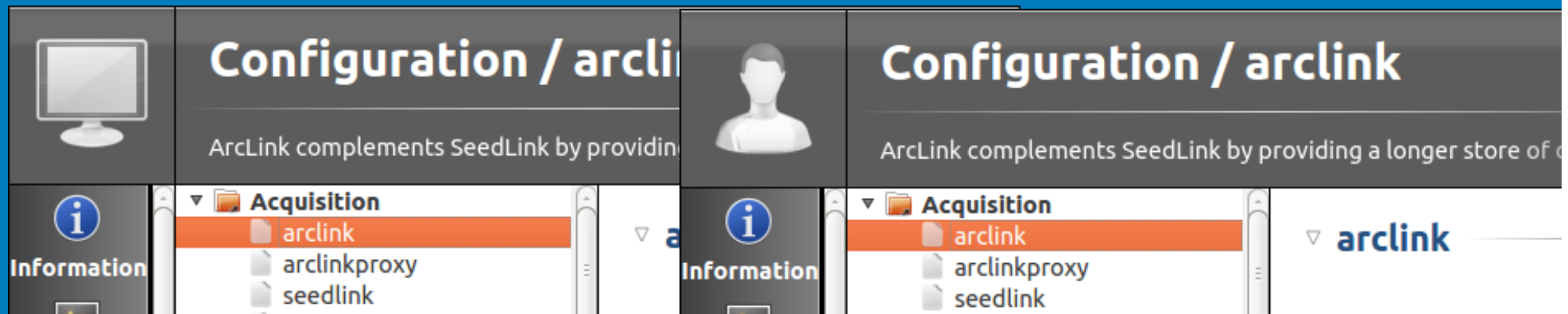
You'll need to install this in your crontab to run automatically; contact your system administrator to do this or google how; you'll often need to use the unfriendly vi editor, Ubuntu gives you nano as an easier option

User vs. System Changes

Click on the server computer icon in upper right hand corner you'll see a change to a person picture. The person indicates user-only changes to SeisComP3 which will go into the .seiscomp3 directory

Use user mode ONLY use for the GUI since that's all user specific.

For normal use, you should see the computer icon.



Local Tuning: Grid

seiscomp3/scautoloc/grid.conf

Look at autoloc docs. The ones installed on your SeisComp3 computer can be found here:

<file:///home/sysop/seiscomp3/share/doc/seiscomp3/html/apps/scautoloc.html#grid-file>



Local Tuning: Grid

you don't want more than 10,000 grid nodes.
Grid just gives you start solution, so it
doesn't have to be too accurate.

Example: in Ecuador - 5 pipelines, one for
each volcano. The could use coarse grid.
5 degrees



Setting Up Multiple Pipelines

- You can have SeisComP3 automatically detect local and teleseismic events with two pipelines.
- You can have SeisComP3 automatically detect local, regional and teleseismic events with three pipelines.
- Note that your station density must be enough to produce local solutions. A minimum of 4 GOOD signals coming from 4 stations is required to declare an event

Setting Up Multiple Pipelines

- The Wiki on seiscomp3.org has lots of good information, including step-by step instructions for installing the pipelines
- <https://www.seiscomp3.org/wiki/recipes/multiple-pick-loc-pipeline>



scbulletin

```
scbulletin -E eventId -d mysql://  
sysop:sysop@localhost/seiscomp3 -3
```

Where the *eventId* is the event's ID in SeisComP3. You can see this with `scolv` for example



scbulletin

Event:

Public ID eventID
Description
 region name: Eastern Honshu, Japan

3 Network magnitudes:

MLv 6.84 +/- 0.56 7
mb 6.70 +/- 1.08 4
M 6.80 7 preferred

Origin:

Date 2013-04-18
Time 09:04:14.0 +/- 0.4 s
Latitude 35.82 deg +/- 5 km
Longitude 137.84 deg +/- 4 km
Depth 78 km +/- 10 km
Agency ISTI
Mode automatic
Status NOT SET
Residual RMS 3.2 s
Azimuthal gap 112 deg

11 Phase arrivals:

sta	net	dist	azi	phase	time	res	wt
KNM	BO	0.5	260	P	09:04:28.9	-0.4	
A	1.0	KNM					
ONS	BO	1.0	70	P	09:04:33.2	0.4	A
A	1.0	ONS					
HJO	BO	3.2	149	P	09:05:01.1	-0.3	
A	1.0	HJO					
SAG	BO	3.7	278	P	09:05:08.2	-0.6	
A	1.0	SAG					
AOG	BO	3.7	154	P	09:05:17.3	-0.2	
AX	0.0	AOG					



scevtls – Find Events in the db

You can use scevtls to find event names to feed scbulletin.

```
scevtls -d mysql://sysop:sysop@localhost/  
seiscomp3 \  
  --begin "2012-01-01 00:00:00" \  
  --end "2013-01-01 00:00:00"
```



Automatic Event Emails

One Way To Do It

scalert detects new locations and solution revisions with a script you can create the content with scbulletin, and send an email with your Linux command-line mail command.



Velocity Models

Documentation here:

[https://www.seiscomp3.org/doc/jakarta/
current/apps/screloc.html](https://www.seiscomp3.org/doc/jakarta/current/apps/screloc.html)

velocity models are here:

seiscomp3/trunk/share/locsats/tables\$ ls

iasp91.LQ iasp91.Pn iasp91.ScS
tab.PKP tab.SKP tab.ms

iasp91.LR iasp91.Rg iasp91.Sg
tab.PKPab tab.SKS tab.pP_

iasp91.Lg iasp91.S iasp91.Sr



SeisComP3 Mailing List

Join the mailing list via the new Forum:

<https://forum.seiscomp3.org/>

