

Jump Quickly to Events Using Timestamps

This script lives in `/sdf/group/lcls/ds/ana/tutorials/psana1_examples/randomAccess.py` and demonstrates how to jump to a specific event using the three "timestamp" numbers that uniquely identify an LCLS event: seconds/nanoseconds/fiducials (fiducials increment at 360Hz and repeat every ~20 minutes, the other two numbers are a standard unix seconds/nanoseconds timestamp). In LCLS-II there will be only one number that will identify an event.

```
from psana import *
ds = DataSource('exp=xpptut15:run=54:smd')

# LCLS1 uses 3 numbers to define an event. In LCLS2 this is one number.
seconds = []
nanoseconds = []
fiducials = []

# get some times of events (these could come from a saved "small data" file, for example)
for nevent, evt in enumerate(ds.events()):
    evtId = evt.get(EventId)
    seconds.append(evtId.time()[0])
    nanoseconds.append(evtId.time()[1])
    fiducials.append(evtId.fiducials())
    if nevent==2: break

# now that we have the times, jump to the events in reverse order
ds = DataSource('exp=xpptut15:run=54:idx')
myrun = next(ds.runs())
for sec,nsec,fid in zip(reversed(seconds),reversed(nanoseconds),reversed(fiducials)):
    et = EventTime(int((sec<<32)|nsec),fid)
    evt = myrun.event(et)
    print(evt.get(EventId).fiducials())
```

This example shows how one uses "idx" (or "index") mode to iterate over events randomly and learn the total number of events in a run:

```
from psana import *
ds = DataSource('exp=xpptut15:run=54:idx')
run = next(ds.runs())
times = run.times()
print(f'Found {len(times)} events in run')
for nevt, t in enumerate(times):
    evt = run.event(t)
    if nevt==2:
        break
```

Note that for technical reasons the number of events in a run can be different in idx and the more-common smd mode.