USGS
science for a changing world

PQLX: A Seismic Data Quality Control System Description, Applications, and Users Manual

By Daniel E. McNamara and Richard I. Boaz

Open-File Report 2010–1292

U.S. Department of the Interior
U.S. Geological Survey

U.S. Department of the Interior
KEN SALAZAR, Secretary

U.S. Geological Survey
Marcia K. McNutt, Director

U.S. Geological Survey, Reston, Virginia: 2011

For more information on the USGS—the Federal source for science about the Earth, its natural and living resources, natural hazards, and the environment, visit http://www.usgs.gov or call 1-888-ASK-USGS

For an overview of USGS information products, including maps, imagery, and publications, visit http://www.usgs.gov/pubprod

To order this and other USGS information products, visit http://store.usgs.gov

Suggested citation:
McNamara, D.E., and Boaz, R.I., 2011, PQLX: A seismic data quality control system description, applications, and users manual: U.S. Geological Survey Open-File Report 2010–1292, 41 p.

Contents

Figures

Conversion Factors

Inch/Pound to SI

Multiply	By	To obtain
	Length	
inch (in.)	2.54	centimeter (cm)
inch (in.)	25.4	millimeter (mm)
foot (ft)	0.3048	meter (m)
mile (mi)	1.609	kilometer (km)
mile, nautical (nmi)	1.852	kilometer (km)
yard (yd)	0.9144	meter (m)
	Area	
acre	4,047	square meter (m^2)
acre	0.4047	hectare (ha)
acre	0.4047	square hectometer (hm^2)
acre	0.004047	square kilometer (km^2)
square foot (ft^2)	929.0	square centimeter (cm^2)
square foot (ft^2)	0.09290	square meter (m^2)
square inch (in^2)	6.452	square centimeter (cm^2)
section (640 acres or 1 square mile)	259.0	square hectometer (hm^2)
square mile (mi^2)	259.0	hectare (ha)
square mile (mi^2)	2.590	square kilometer (km^2)
	Volume	
barrel (bbl), (petroleum, 1 barrel=42 gal)	0.1590	cubic meter (m^3)
ounce, fluid (fl. oz)	0.02957	liter (L)
pint (pt)	0.4732	liter (L)
quart (qt)	0.9464	liter (L)
gallon (gal)	3.785	liter (L)
gallon (gal)	0.003785	cubic meter (m^3)
gallon (gal)	3.785	cubic decimeter (dm^3)
million gallons (Mgal)	3,785	cubic meter (m^3)
cubic inch (in^3)	16.39	cubic centimeter (cm^3)
cubic inch (in^3)	0.01639	cubic decimeter (dm^3)
cubic inch (in^3)	0.01639	liter (L)

cubic foot (ft^3)	28.32	cubic decimeter (dm^3)
cubic foot (ft^3)	0.02832	cubic meter (m^3)
cubic yard (yd^3)	0.7646	cubic meter (m^3)
cubic mile (mi^3)	4.168	cubic kilometer (km^3)
acre-foot (acre-ft)	1,233	cubic meter (m^3)
acre-foot (acre-ft)	0.001233	cubic hectometer (hm^3)
	Flow rate	
acre-foot per day (acre-ft/d)	0.01427	cubic meter per second (m^3/s)
acre-foot per year (acre-ft/yr)	1,233	cubic meter per year (m^3/yr)
acre-foot per year (acre-ft/yr)	0.001233	cubic hectometer per year (hm^3/yr)
foot per second (ft/s)	0.3048	meter per second (m/s)
foot per minute (ft/min)	0.3048	meter per minute (m/min)
foot per hour (ft/hr)	0.3048	meter per hour (m/hr)
foot per day (ft/d)	0.3048	meter per day (m/d)
foot per year (ft/yr)	0.3048	meter per year (m/yr)
cubic foot per second (ft^3/s)	0.02832	cubic meter per second (m^3/s)
cubic foot per second per square mile [$(ft^3/s)/mi^2$]	0.01093	cubic meter per second per square kilometer [$(m^3/s)/km^2$]
cubic foot per day (ft^3/d)	0.02832	cubic meter per day (m^3/d)
gallon per minute (gal/min)	0.06309	liter per second (L/s)
gallon per day (gal/d)	0.003785	cubic meter per day (m^3/d)
gallon per day per square mile [$(gal/d)/mi^2$]	0.001461	cubic meter per day per square kilometer [$(m^3/d)/km^2$]
million gallons per day (Mgal/d)	0.04381	cubic meter per second (m^3/s)
million gallons per day per square mile [$(Mgal/d)/mi^2$]	1,461	cubic meter per day per square kilometer [$(m^3/d)/km^2$]
inch per hour (in/h)	0 .0254	meter per hour (m/h)
inch per year (in/yr)	25.4	millimeter per year (mm/yr)
mile per hour (mi/h)	1.609	kilometer per hour (km/h)

Mass		
ounce, avoirdupois (oz)	28.35	gram (g)
pound, avoirdupois (lb)	0.4536	kilogram (kg)
ton, short (2,000 lb)	0.9072	megagram (Mg)
ton, long (2,240 lb)	1.016	megagram (Mg)
ton per day (ton/d)	0.9072	metric ton per day
ton per day (ton/d)	0.9072	megagram per day (Mg/d)
ton per day per square mile [(ton/d)/mi^2]	0.3503	megagram per day per square kilometer [(Mg/d)/km^2]
ton per year (ton/yr)	0.9072	megagram per year (Mg/yr)
ton per year (ton/yr)	0.9072	metric ton per year
Pressure		
atmosphere, standard (atm)	101.3	kilopascal (kPa)
bar	100	kilopascal (kPa)
inch of mercury at 60ºF (in Hg)	3.377	kilopascal (kPa)
pound-force per square inch (lbf/in^2)	6.895	kilopascal (kPa)
pound per square foot (lb/ft^2)	0.04788	kilopascal (kPa)
pound per square inch (lb/in^2)	6.895	kilopascal (kPa)
Density		
pound per cubic foot (lb/ft^3)	16.02	kilogram per cubic meter (kg/m^3)
pound per cubic foot (lb/ft^3)	0.01602	gram per cubic centimeter (g/cm^3)
Energy		
kilowatthour (kWh)	3,600,000	joule (J)
Radioactivity		
picocurie per liter (pCi/L)	0.037	becquerel per liter (Bq/L)
Specific capacity		
gallon per minute per foot [(gal/min)/ft)]	0.2070	liter per second per meter [(L/s)/m]
Hydraulic conductivity		
foot per day (ft/d)	0.3048	meter per day (m/d)
Hydraulic gradient		
foot per mile (ft/mi)	0.1894	meter per kilometer

		(m/km)
	Transmissivity*	
foot squared per day (ft^2/d)	0.09290	meter squared per day (m^2/d)
	Application rate	
pounds per acre per year [(lb/acre)/yr]	1.121	kilograms per hectare per year [(kg/ha)/yr]
	Leakance	
foot per day per foot [(ft/d)/ft]	1	meter per day per meter
inch per year per foot [(in/yr)/ft]	83.33	millimeter per year per meter [(mm/yr)/m]

Temperature in degrees Celsius (°C) may be converted to degrees Fahrenheit (°F) as follows:

°F=(1.8×°C)+32

Temperature in degrees Fahrenheit (°F) may be converted to degrees Celsius (°C) as follows:

°C=(°F-32)/1.8

Vertical coordinate information is referenced to the insert datum name (and abbreviation) here for instance, "North American Vertical Datum of 1988 (NAVD 88)."

Horizontal coordinate information is referenced to the insert datum name (and abbreviation) here for instance, "North American Datum of 1983 (NAD 83)."

Altitude, as used in this report, refers to distance above the vertical datum.

*Transmissivity: The standard unit for transmissivity is cubic foot per day per square foot times foot of aquifer thickness [(ft^3/d)/ft^2]ft. In this report, the mathematically reduced form, foot squared per day (ft^2/d), is used for convenience.

Specific conductance is given in microsiemens per centimeter at 25 degrees Celsius (µS/cm at 25 °C).

Concentrations of chemical constituents in water are given either in milligrams per liter (mg/L) or micrograms per liter (µg/L).

NOTE TO USGS USERS: Use of hectare (ha) as an alternative name for square hectometer (hm^2) is restricted to the measurement of small land or water areas. Use of liter (L) as a special name for cubic decimeter (dm^3) is restricted to the measurement of liquids and gases. No prefix other than milli should be used with liter. Metric ton (t) as a name for megagram (Mg) should be restricted to commercial usage, and no prefixes should be used with it.

SI to Inch/Pound

Multiply	By	To obtain
	Length	
centimeter (cm)	0.3937	inch (in.)
millimeter (mm)	0.03937	inch (in.)
meter (m)	3.281	foot (ft)
kilometer (km)	0.6214	mile (mi)
kilometer (km)	0.5400	mile, nautical (nmi)
meter (m)	1.094	yard (yd)
	Area	
square meter (m^2)	0.0002471	acre
hectare (ha)	2.471	acre
square hectometer (hm^2)	2.471	acre
square kilometer (km^2)	247.1	acre
square centimeter (cm^2)	0.001076	square foot (ft^2)
square meter (m^2)	10.76	square foot (ft^2)
square centimeter (cm^2)	0.1550	square inch (ft^2)
square hectometer (hm^2)	0.003861	section (640 acres or 1 square mile)
hectare (ha)	0.003861	square mile (mi^2)
square kilometer (km^2)	0.3861	square mile (mi^2)
	Volume	
cubic meter (m^3)	6.290	barrel (petroleum, 1 barrel = 42 gal)
liter (L)	33.82	ounce, fluid (fl. oz)
liter (L)	2.113	pint (pt)
liter (L)	1.057	quart (qt)
liter (L)	0.2642	gallon (gal)
cubic meter (m^3)	264.2	gallon (gal)
cubic decimeter (dm^3)	0.2642	gallon (gal)
cubic meter (m^3)	0.0002642	million gallons (Mgal)
cubic centimeter (cm^3)	0.06102	cubic inch (in^3)
cubic decimeter (dm^3)	61.02	cubic inch (in^3)
liter (L)	61.02	cubic inch (in^3)
cubic decimeter (dm^3)	0.03531	cubic foot (ft^3)
cubic meter (m^3)	35.31	cubic foot (ft^3)
cubic meter (m^3)	1.308	cubic yard (yd^3)

cubic kilometer (km^3)	0.2399	cubic mile (mi^3)
cubic meter (m^3)	0.0008107	acre-foot (acre-ft)
cubic hectometer (hm^3)	810.7	acre-foot (acre-ft)
Flow rate		
cubic meter per second (m^3/s)	70.07	acre-foot per day (acre-ft/d)
cubic meter per year (m^3/yr)	0.000811	acre-foot per year (acre-ft/yr)
cubic hectometer per year (hm^3/yr)	811.03	acre-foot per year (acre-ft/yr)
meter per second (m/s)	3.281	foot per second (ft/s)
meter per minute (m/min)	3.281	foot per minute (ft/min)
meter per hour (m/hr)	3.281	foot per hour (ft/hr)
meter per day (m/d)	3.281	foot per day (ft/d)
meter per year (m/yr)	3.281	foot per year ft/yr)
cubic meter per second (m^3/s)	35.31	cubic foot per second (ft^3/s)
cubic meter per second per square kilometer [(m^3/s)/km^2]	91.49	cubic foot per second per square mile [(ft^3/s)/mi^2]
cubic meter per day (m^3/d)	35.31	cubic foot per day (ft^3/d)
liter per second (L/s)	15.85	gallon per minute (gal/min)
cubic meter per day (m^3/d)	264.2	gallon per day (gal/d)
cubic meter per day per square kilometer [(m^3/d)/km^2]	684.28	gallon per day per square mile [(gal/d)/mi^2]
cubic meter per second (m^3/s)	22.83	million gallons per day (Mgal/d)
cubic meter per day per square kilometer [(m^3/d)/km^2]	0.0006844	million gallons per day per square mile [(Mgal/d)/mi^2]
cubic meter per hour (m^3/h)	39.37	inch per hour (in/h)
millimeter per year (mm/yr)	0.03937	inch per year (in/yr)
kilometer per hour (km/h)	0.6214	mile per hour (mi/h)
Mass		
gram (g)	0.03527	ounce, avoirdupois (oz)
kilogram (kg)	2.205	pound avoirdupois (lb)
megagram (Mg)	1.102	ton, short (2,000 lb)
megagram (Mg)	0.9842	ton, long (2,240 lb)
metric ton per day	1.102	ton per day (ton/d)
megagram per day (Mg/d)	1.102	ton per day (ton/d)
megagram per day per square kilometer [(Mg/d)/km^2]	2.8547	ton per day per square mile [(ton/d)/mi^2]

megagram per year (Mg/yr)	1.102	ton per year (ton/yr)
metric ton per year	1.102	ton per year (ton/yr)
	Pressure	
kilopascal (kPa)	0.009869	atmosphere, standard (atm)
kilopascal (kPa)	0.01	bar
kilopascal (kPa)	0.2961	inch of mercury at 60°F (in Hg)
kilopascal (kPa)	0.1450	pound-force per inch (lbf/in)
kilopascal (kPa)	20.88	pound per square foot (lb/ft^2)
kilopascal (kPa)	0.1450	pound per square inch (lb/ft^2)
	Density	
kilogram per cubic meter (kg/m^3)	0.06242	pound per cubic foot (lb/ft^3)
gram per cubic centimeter (g/cm^3)	62.4220	pound per cubic foot (lb/ft^3)
	Energy	
joule (J)	0.0000002	kilowatthour (kWh)
	Radioactivity	
becquerel per liter (Bq/L)	27.027	picocurie per liter (pCi/L)
	Specific capacity	
liter per second per meter [(L/s)/m]	4.831	gallon per minute per foot [(gal/min)/ft]
	Hydraulic conductivity	
meter per day (m/d)	3.281	foot per day (ft/d)
	Hydraulic gradient	
meter per kilometer (m/km)	5.27983	foot per mile (ft/mi)
	Transmissivity*	
meter squared per day (m^2/d)	10.76	foot squared per day (ft^2/d)
	Application rate	
kilograms per hectare per year [(kg/ha)/yr]	0.8921	pounds per acre per year [(lb/acre)/yr]
	Leakance	
meter per day per meter [(m/d)/m]	1	foot per day per foot [(ft/d)/ft]
millimeter per year per meter [(mm/yr)/m]	0.012	inch per year per foot [(in/yr)/ft]

Temperature in degrees Celsius (°C) may be converted to degrees Fahrenheit (°F) as follows:

°F=(1.8×°C)+32

Temperature in degrees Fahrenheit (°F) may be converted to degrees Celsius (°C) as follows:

°C=(°F-32)/1.8

Vertical coordinate information is referenced to the insert datum name (and abbreviation) here, for instance, "North American Vertical Datum of 1988 (NAVD 88)"

Horizontal coordinate information is referenced to the insert datum name (and abbreviation) here, for instance, "North American Datum of 1983 (NAD 83)"

Altitude, as used in this report, refers to distance above the vertical datum.

*Transmissivity: The standard unit for transmissivity is cubic foot per day per square foot times foot of aquifer thickness [(ft^3/d)/ft^2]ft. In this report, the mathematically reduced form, foot squared per day (ft^2/d), is used for convenience.

Specific conductance is given in microsiemens per centimeter at 25 degrees Celsius (µS/cm at 25°C).

Concentrations of chemical constituents in water are given either in milligrams per liter (mg/L) or micrograms per liter (µg/L).

NOTE TO USGS USERS: Use of hectare (ha) as an alternative name for square hectometer (hm^2) is restricted to the measurement of small land or water areas. Use of liter (L) as a special name for cubic decimeter (dm^3) is restricted to the measurement of liquids and gases. No prefix other than milli should be used with liter. Metric ton (t) as a name for megagram (Mg) should be restricted to commercial usage, and no prefixes should be used with it.

PQLX: A Seismic Data Quality Control System Description, Applications, and Users Manual

By Daniel E. McNamara and Richard I. Boaz

Introduction

We present a detailed description and users manual for a new tool to evaluate seismic station performance and characteristics by providing quick and easy transitions between visualizations of the frequency and time domains. The software is based on the probability density functions (PDF) of power spectral densities (PSD) (McNamara and Buland, 2004) and builds on the original development of the PDF stand-alone software system (McNamara and Boaz, 2005) and the seismological data viewer application PQL (IRIS-PASSCAL Quick Look) and PQLII (available through the IRIS PASSCAL program: http://www.passcal.nmt.edu/content/pql-ii-program-viewing-data). With PQLX (PQL eXtended), computed PSDs are stored in a MySQL database, allowing a user to access specific time periods of PSDs (PDF subsets) and time series segments through a GUI-driven interface. The power of the method and software lies in the fact that there is no need to screen the data for system transients, earthquakes, or general data artifacts, because they map into a background probability level. In fact, examination of artifacts related to station operation and episodic cultural noise allow us to estimate both the overall station quality and a baseline level of Earth noise at each site. The output of this analysis tool is useful for both operational and scientific applications. Operationally, it is useful for characterizing the current and past performance of existing broadband stations, for conducting tests on potential new seismic station locations, for evaluating station baseline noise levels (McNamara and others, 2009), for detecting problems with the recording system or sensors, and for evaluating the overall quality of data and metadata. Scientifically, the tool allows for mining of PSDs for investigations on the evolution of seismic noise (for example, Aster and others, 2008; and Aster and others, 2010) and other phenomena. Currently, PQLX is operational at several organizations including the USGS National Earthquake Information Center (NEIC), the USGS Albuquerque Seismological Laboratory (ASL), and the Incorporated Research Institutions in Seismology (IRIS) Data Management Center (DMC) for

station monitoring and instrument response quality control. The PQLX system is available to the community at large through the U.S. Geological Survey (USGS) (http://ehpm-earthquake.wr.usgs.gov/research/software/pqlx.php) and IRIS (http://www.iris.edu/software/pqlx). Also provided is a fully searchable website for bug reporting and enhancement requests (http://wush.net/bugzilla/PQLX).

The first part of this document aims to describe and illustrate some of the features and capabilities of the software. The second part of this document is a detailed users manual that covers installation procedures, system requirements, operations, bug reporting, and software components (Appendix).

Server-Side Analysis

The PQLX server-analysis program processes data files and stores these results in a user-specified database. The server can be executed either directly from a command line or automated via cron. The system auto-detects and handles the following seismic data formats: Raw Reftek (RT100), mini-SEED, AH, SEGY, SAC, DR100, and NANO. The system is scalable from individual small installations (such as a temporary array) to very large and permanent datasets (multiple networks containing > 8,000 real-time channels).

All data channels "discovered" by the server process are processed with all header information (start time, length, sample rate, and so forth), and location of gaps and overlaps (mini-SEED format only) stored in the database. In addition, channels may be configured for PSD analysis using the method of McNamara and Buland (2004)

Deconvolution of the trace data for PSD analysis is computed using the RESP file output from the SEED format processing program rdseed (available from the IRIS DMC: http://www.iris.edu/software/downloads/rdseed_request.htm), used as input to evalresp() (http://www.iris.edu/manuals/evalresp.htm); deconvolution also includes any digital filters. Various PSD parameters are configurable by the user including window length, maximum period bound, and minimum and maximum power bounds. By a simple addition to a database table, the user has the ability to define, in generic terms, which channels (following a specific naming convention, for example, BH*) should be analyzed using a specific configuration. For example, a single database may define two different PSD configurations for BH

channels and LH channels. The system is configured to analyze the following channel groups: LH, BH, BL, HG, HN, HL, BH, BN, HH, SH, EH, and EP.

The server also allows the user to import (via XML) other types of data for use by the client GUI. This includes seismic event information (such as an event catalog), as well as metadata information for each channel (such as name, location, latitude, longitude, instrument type, and sensitivity).

The client GUI program, pqlx, provides a tool to view the processed data. It can connect to the database for visualization and client query of the analysis results. It is within the GUI application that most uses and features of the PQLX system are found.

Client GUI Application—PQLX

Building upon the framework of PQL and PQL II, the client-side GUI can connect to either a local database (located on the same machine) or a remote server, either LAN-based or WAN-based (including Internet-wide). This data visualization application is responsible for displaying all graphics of the PQLX system, and is separated into three tools: the Trace Viewer (the original PQL), the PDF Viewer, and the STN (station) Viewer. Each provides for different viewing capabilities of seismological data and contributes individually and collectively to the task of data quality control.

Trace Viewer

The Trace Viewer allows for display of waveform data read in by physical file. It provides for the magnification of traces, including multiple zooming options, spectral analysis of selected data, viewing of each of these data views in split screen mode simultaneously, as well as display of all header values (fig. 1). (Use of the Trace Viewer does not require a connection to a PQLX database. Indeed, a stand-alone version of PQL is delivered with the PQLX system.) Since all viewers are contained in the same application, pqlx, the Trace Viewer can be invoked from both PDF and STN viewers.

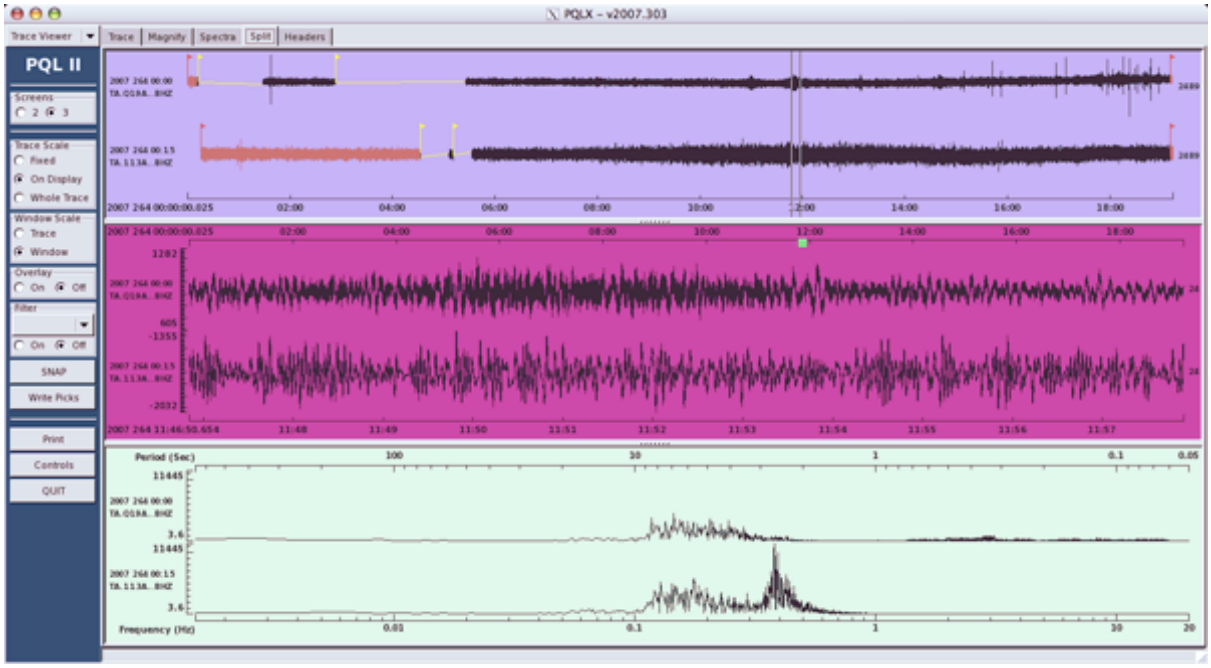


Figure 1. A sample Trace Viewer Split Screen Tab displaying three complete traces, a magnified portion, and the spectral transformation of this magnified portion.

PDF Viewer

After connecting to a database, the PDF Viewer allows for display of various types of PDFs based on the previously computed PSD data. Either System PDFs held on the database, or PDF's based on user-provided date and time parameters (a PDF of all PSDs for the month of June over all years of data held) can be requested for display. The main display tab has nine panels and renders System PDFs in various combinations: by station (three different PDFs for a chosen channel group of a single station), by PDF (three different stations for a chosen channel group of a chosen PDF), by both (three different PDFs for three different channels), or as a list (a list of chosen channels of a chosen PDF) (fig. 2).

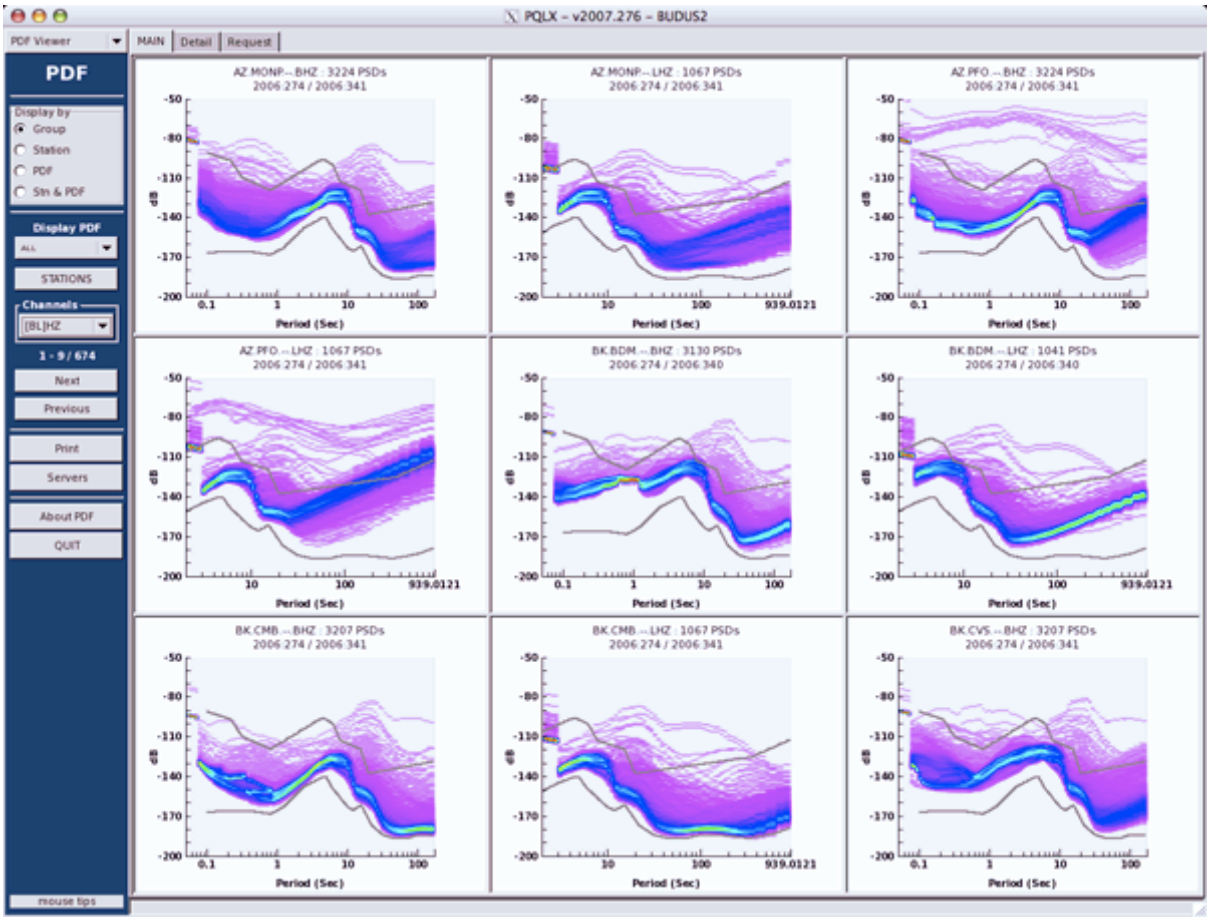


Figure 2. The nine-panel PDF Viewer Main Tab with the following selections defined by the sidebar controls: all PSDs held on the database (System PDF All) channels BHZ and LHZ for all stations in the database.

Clicking on any PDF on display in the Main Tab will take the PDF to the Detail Tab of the PDF Viewer for further analysis. This data view allows a user to select a specified portion of the PDF; this "sub-select" is defined as either a single point or bounding box. Once specified, three additional views are displayed: a PDF of all PSDs intersecting through the point or bounding box, a histogram displaying start and stop times of all PSDs (X-axis = day of year, Y-axis = hour of day), and a view of the first 15 PSD source traces (fig. 3). In addition, mouse-clicking on the lower left panel PDF makes this the new main Detail PDF for further sub-selection in the frequency domain. Mouse-clicking on the trace panel takes the user to the trace viewer for in-depth inspection of the PSD source traces in the time domain. The traces are retrieved from the disk location where data files have been saved, or if the files are not available on the local machine disk then they can be retrieved from a CWB

server where they have been stored, if the corresponding IP address and port have been provided and put in the "Detail" section of the "Controls" panel.

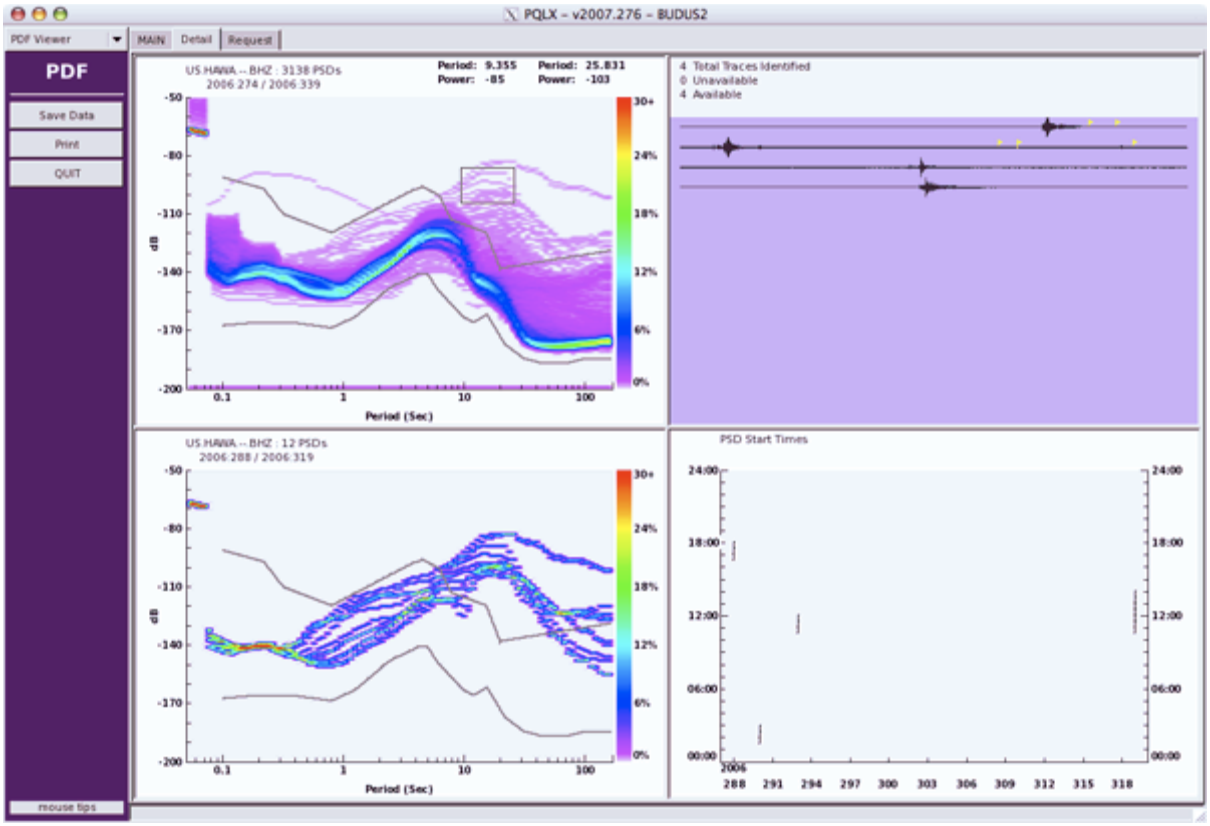


Figure 3. The PDF Detail Tab. The upper left panel displays System PDF All with a sub-select bounding box. The lower left panel displays the resultant PDF of PSDs intersecting the bounding box defined between periods of 9.3 to 25.8 seconds, and -85 to -103 dB (large earthquakes). The lower right panel is a histogram displaying the start times of the intersecting PSDs. The upper right panel displays the source traces for the intersecting PSDs, here containing the large earthquakes.

STN Viewer

The third display is the STN Viewer, organizing the display of trace data by station and channel. Fully configurable, it allows the user to specify which stations and channels should be viewed (traversed as a list), how many days of data should be rendered (between 1 and 60), how many lines to display per page, as well as whether the data should be drawn as actual data, or simply rendered as a horizontal line indicating that data exists. This

last option allows the user to check for the existence of data (as well as gaps and overlaps) via a connection to a database while not having access to the waveforms themselves (Fig. 4).

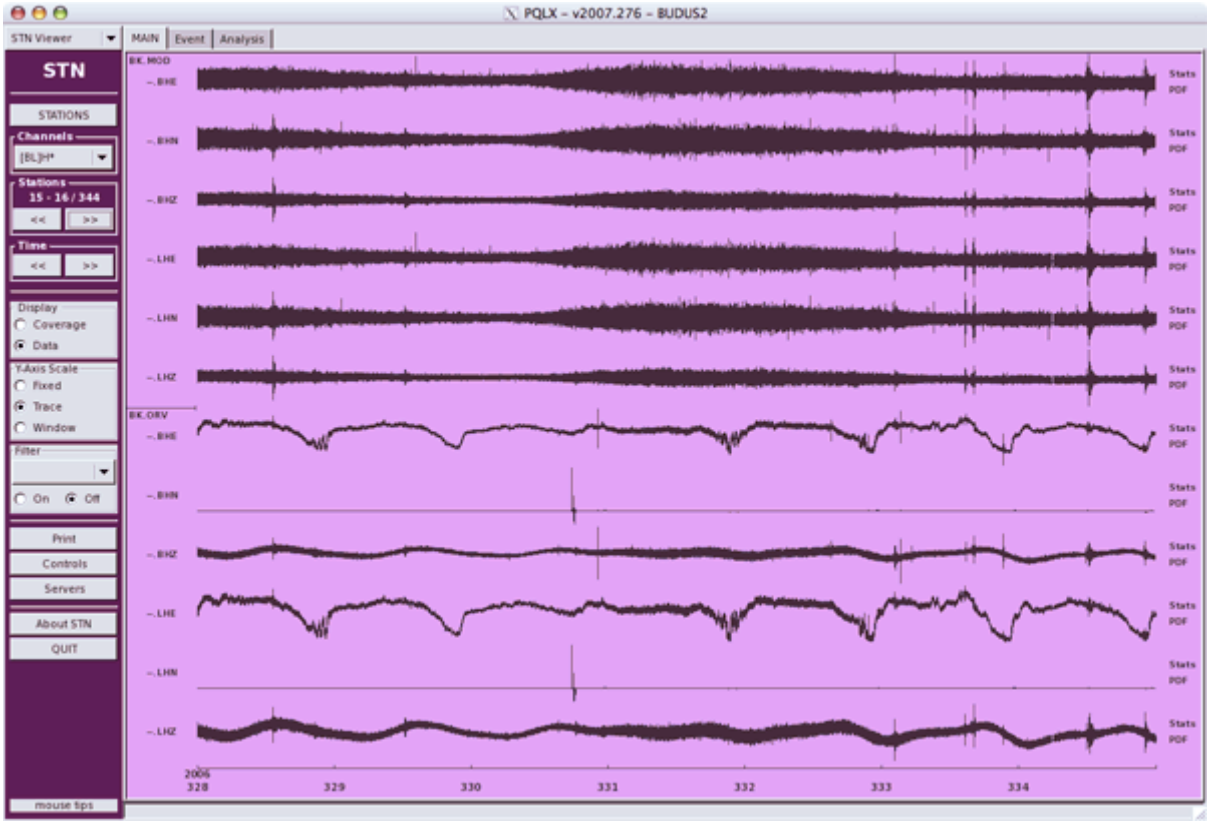


Figure 4*A*. The Main Tab of the STN Viewer displaying two stations of data, displayed channels filtered using "[BL]H*" (that is, all BH and LH channels), for one week.

Hovering over the channel label displays a pop-up providing channel metadata information previously imported by the server (fig. 4b).

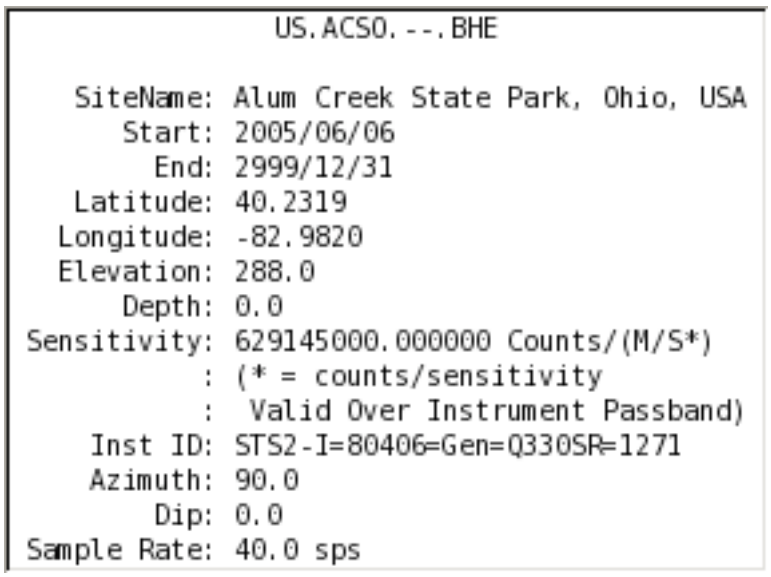


Figure 4*B*. Metadata information.

Hovering over the 'Stats' text for each channel displays a pop-up providing statistical quality control (QC) information such as start and end dates, total number of traces, total gaps and overlaps, and maximum and minimum values (fig. 4c).

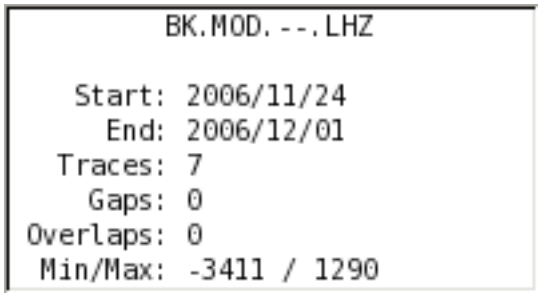


Figure 4*C*. Quality control information.

Hovering over the 'PDF' text displays a pop-up displaying the PDF of the data on display and with an additional click takes the user to the PDF Viewer Detail Tab for investigation (fig. 4d).

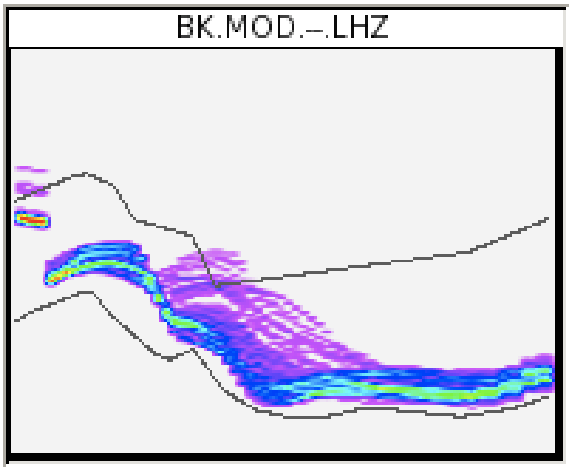


Figure 4*D*. PSD PDF of the data.

Quality Control Applications

PQLX can be used to explore problems with seismological data, including data gaps and overlaps, instrument-based anomalies, metadata quality, and local noise sources. Historical datasets can be assessed for overall quality, thus increasing the confidence of scientific results. Operators of real-time seismic networks can also benefit from the PQLX system as it allows near real-time analysis and response to data and telemetry problems.

Application at the USGS NEIC

The USGS NEIC receives over 4,500 channels of seismic data from over 500 global seismic stations in real time. For many contributed stations, calibration information is not well known. In addition, instrumentation upgrades or changes occur, making it difficult to maintain accurate and timely metadata. The use of real-time seismic data requires automated QC tools to ensure the accuracy of NEIC real-time earthquake products. Currently, the NEIC computes PSDs for all incoming channels using the PQLX software system on a daily basis. In order to identify out-of-nominal noise conditions, such as instrument response changes or systems transients, we visually compare the short-term PDFs against the long-term station noise baselines. This method is very sensitive to instrument response inaccuracies (fig. 5).

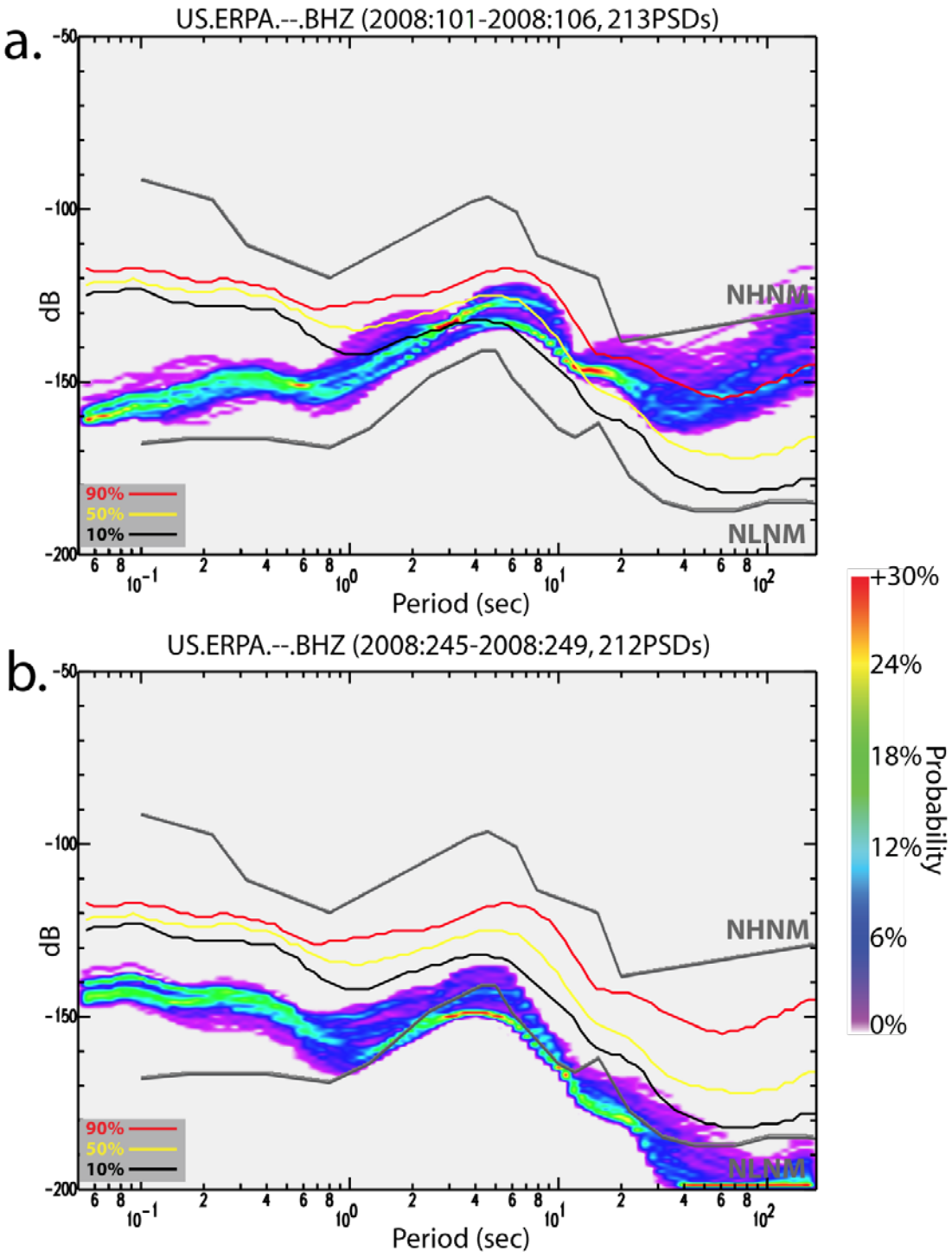


Figure 5. PDF examples using the Advanced National Seismic System (ANSS) backbone station US.ERPA.--.BHZ in Erie, Pennsylvania. (a) An incorrect instrument response has been applied to 213 hours of data, during 2008:101 to 2008:106, in order to demonstrate the sensitivity of this method to a possible error in instrument

response units. In this example, units of acceleration instead of velocity, expressed as an extra zero in the response file, results in counter-clockwise rotation of the PDF. This is clearly observed as low power at short-periods and high power at long-periods, relative to the long-term station baseline model. In contrast, when a zero is added to the instrument response it results in a counter-clockwise rotation of the PSDs. (b) When an incorrect sensitivity of several orders of magnitude too large is applied to US.ERPA.--.BHZ it results in an over-correction of the data and PSDs are well below the station baseline and NLNM.

Steps Toward Automation

In a system under development at the NEIC, hourly PSDs are compared to the long-term station baselines (McNamara and others, 2009). If the hourly PSD does not fall within the bounds of the station baseline, it is flagged as "out of nominal" and then compared to a set of station-specific noise source models. An effort has begun to characterize station-specific noise source models in an attempt to more precisely monitor a station's state of health. "Out of nominal" noise conditions currently monitored include (1) calibration pulses, (2) missing data, (3) spikes, and (4) mass re-centers. Once these noise models are defined, the server will be able to compare analysis results as they are created with the noise models and, if within (or without) the noise model bounds, will flag this portion of the trace as having breached the noise model in question. A message screen within the GUI will alert the user to the existence of this breach/flag (fig. 6).

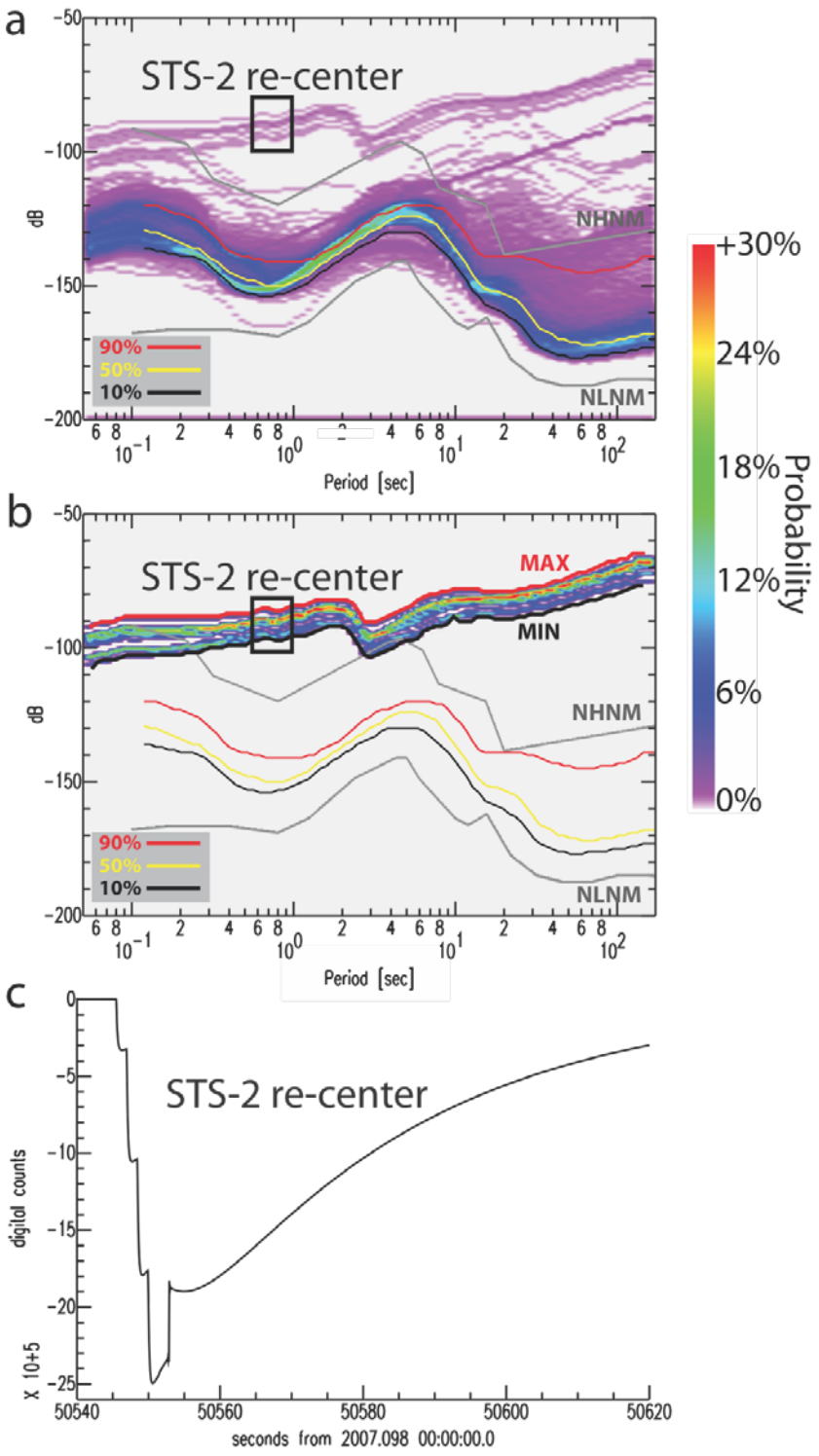

a
STS-2 re-center
dB
NHNM
NLNM
90%
50%
10%
Period [sec]
b
STS-2 re-center
MAX
MIN
NHNM
NLNM
90%
50%
10%
Period [sec]
+30%
24%
18%
12%
6%
0%
Probability
c
STS-2 re-center
digital counts
X 10+5
seconds from 2007.098 00:00:00.0

Figure 6. An example of a common excursion from the station baseline due to the re-centering of the Streckeisen STS2 seismometer at the ANSS backbone station US.LRAL. Using the PQLX software, an analyst can select groups of similar PSDs (fig. 6a), define a noise-source model by storing the characteristic maximum and minimum (fig. 6b), and also view the time series segments through the client interface in order to determine the source and characteristics of the noise excursion (fig. 6c). These steps allow the user to visually define the spectral characteristics of known system transients for comparison against the long-term station baselines. In the prototype-automated system, when an hourly PSD falls outside of the long-term station baseline, it is then compared to the set of known noise-source models for that channel. If a match of at least 75% occurs, a noise-source detection is declared, stored, and ultimately compiled in a report for further investigation by the analyst, system developers, and operations managers.

Future Development

PQLX continues to be a work in progress. Additional forthcoming development includes:

- Control of server execution, database and general system administration via the GUI client, pqlx
- Analysis of strong motion data
- Ability to make movies from a series of PDFs over time
- Automated report generation and notification based on user defined QC parameters
- Overlay additional statistics on PDF, for example 90^{th} and 10^{th} percentiles, median, mode, and so forth.
- User defined PSD processing parameters (window length, overlap).

Where To Get It

PQLX has been designed to be easily implemented. Fully open-source, easy to compile and implement, and self-configuring based on the existence of data, it allows the network operator and seismologist to concentrate their time and energies on what is important (what the data are trying to tell them).

Source code for PQLX is freely available for download via WWW from:

Download From: http://wush.net/svn/PQLX/tars

Login Username: PQLXtars

Login Password: getPQLX

This site can be accessed both through the USGS:

http://ehpm-earthquake.wr.usgs.gov/research/software/pqlx.php

and IRIS:

http://www.iris.edu/software/pqlx .

Also provided is a fully searchable website for bug reporting and enhancement requests; this can be found at http://wush.net/bugzilla/PQLX. For additional information including the PQLX installation and operations manual and PDF interpretation and method details see

http://geohazards.cr.usgs.gov/staffweb/mcnamara/Software/PQLX.html.

If you would like to be added to the PQLX mailing list, and be automatically informed of availability of future updates, please email Richard Boaz at ivor.boaz@gmail.com.

System Requirements

Compilation and installation of the entire system happens with a single command and has been demonstrated to work under the following operating systems: LINUX (multiple flavors, including 64-bit architecture), MAC OS X, and Solaris. Portability has been a primary concern, as such, porting to additional systems is likely possible with little effort.

Development Philosophy, History, and Acknowledgments

The PQLX software system has been (and continues to be) developed based on open-source software and is itself open-source (licensed under the GNU GPL, version 2). The overall design provides an architectural framework that is intended to be expandable in the future for inclusion of additional functionality as needed, for ease of use (both for end-user and technical maintainer), and for ease of software maintenance. The system comprises both contributed software and original development.

New development in the current release includes database design, server-side analysis program, client-side user interface, PDF image rendering program (to .png format), and data extraction API shell scripts, among others. All original development is provided by Richard Boaz with additional software contributions provided by:

- MySQL - the server-side database holding all analysis results
- GTK+ - the graphical user interface library used to render the client-side GUI, named pqlx.

- FFTW3 - the Fourier Transform library, provided by MIT (incorporated by Pete Lombard of the Berkely Seismological Laboratory)
- gdbm - the GNU database interface holding all client-side user information
- libmseed - a mini-SEED data format file reader library, provided by IRIS DMS
- evalresp - a SEED Response File reader, provided by ISTI
- vfbb - a DR100 data format file reader library, provided by the USGS
- Michael Aramini - High Resolution Bi-Cubic Spline Interpolator, used to render PDF images real-time

PQLX has been developed with major support and funding from the United States Geological Survey, IRIS, and the National Science Foundation. Finally, the authors would like to thank A. Ringler, S. Rastin and L. Gee for thoughtful comments that improved the presentation of this material.

References Cited

Aster, R., McNamara, D., Bromirski, P., 2008, Multi-decadal climate-induced oceanic microseism variability: Seismo. Res. Lett., 79, 2.

Aster, R.C., McNamara, D.E. and Bromirski, P., 2010, Global trends in extremal microseism intensity: Geophys. Res. Lett., 37, L14303, doi:10.1029/2010GL043472.

McNamara, D.E., Hutt, C.R., Gee, L.S., Benz, H.M., and Buland, R.P., 2009, A method to establish seismic noise baselines for automated station assessment: Seismo. Res. Lett., 80, 628-637.

McNamara, D. E., Boaz, R.I., 2005, Seismic noise analysis system, power spectral density probability density function: standalone softare package: United States Geological Survey Open File Report, 2005-1438, 30p.

McNamara, D.E., and Buland, R.P., 2004, Ambient noise levels in the continental United States: Bull. Seism. Soc. Am., 94, 4, 1517-1527.

Appendix

PQLX System Installation and Configuration

Introduction

This document describes all setup dependencies, requirements, and actions necessary to create a PQLX server instance, database(s), and the client-side GUI programs. Please read the entire document before proceeding with the installation as this will provide all the background necessary to get a system up and running in a successful manner.

The PQLX system will compile and execute on any of the following three platforms:
LINUX
Mac OS
Solaris

Setup of the PQLX server and system is generally defined to comprise the following actions:

- Server-side technical environment setup, performed once per server machine instance;
- PQLX software compilation and installation, executed once per platform the PQLX server and client will run on, for both server- and client-side programs; and
- PQLX database definition and creation, executed once for each database required.

Server-Side Technical Setup

Server-side technical environment setup comprises the following individual steps:

- Installation of MySQL database server

The database employed to support the PDF database is MySQL (at http://www.mysql.org); it is open-source and fully supported. Download the latest version of the MySQL Community Edition (being certain to download the version necessary for development) and install to the machine designated to be the PDF-PQLX server (typically downloadable from http://www.mysql.org/downloads/, minimum version = 5.0.18).

This may be installed on either a true server machine (on a network with clients connecting from other machines), a stand-alone installation (server and client to reside on the same machine), or both. From a technical standpoint as regards the MySQL database installation, there are no differences between these possible configurations.

Once the database has been installed, start the server (as per platform-specific instructions, using the mysqld_safe script form) and confirm success through execution of the mysql client access program. See installation and startup notes of MySQL for details. As well, consider automating the MySQL server execution on system startup (see platform-specific instructions with the MySQL documentation set), otherwise, the MySQL server will be required to be manually started each time the server machine is booted.

One common problem with the MySQL installation is that the default destination directory for MySQL databases may not be large enough. For reasonably large datasets, the required amount of disk space will also be correspondingly large. Be certain to confirm at time of installation that the directory MySQL will use for its databases is located on a file mount that is ultimately large enough for PQLX usage.

PQLX Software Installation, Compilation, and Production Setup

Installation and configuration of the PQLX system is composed of the following steps:

Satisfy all prerequisites
Compile the system
Install all executables to production directories
Initialize the server database for PQLX usage

Download the latest version of the available source code tar-ball. Decompress and de-tar to the destination directory of choice.

Compilation Pre-Requisites

MySQL

Once the installation of the MySQL server has completed, two environment variables must be set up prior to system compilation. The two entries contained in the file $PQLX/env/mysql.vars define the following MySQL-specifc environment variables:

MYSQLBIN - should be set to the bin directory of the MySQL installation; subsequently used during system execution.
(This will be the directory containing the MYSQL server named mysqld_safe.)
MYSQLINC - should be set to the include directory for MySQL; subsequently used only for system compilation.
(This will be the directory containing the MYSQL file named mysql.h.)

For Solaris implementations, the LD_RUN_PATH variable (set at beginning of env/makeVars file) must include both the PQL GTK+ libraries directory (defaults to PASSCAL installation, change if this path is not valid), and the directory containing the MySQL libraries, please modify accordingly.

It is intended that the modifications made to the file env/mysql.vars be saved in a separate location by the compiling user for subsequent use in future complications of the system. This way, when a new release of the PQLX software is made available, this file can be copied into the PQLX/env directory immediately prior to compilation, alleviating the need to make these modifications by hand each time.

GTK+

A local GTK+ installation must exist on the compiling machine; this can be achieved in one of two ways. It is possible that the minimum version of GTK+ is already installed on the machine. To check if this is the case, execute the following command:

```
sh> pkg-config --modversion gtk+-2.0
```

If this returns a version greater than or equal to 2.6.7, then the GTK+ requirement is already satisfied for this machine; you may skip to the next section Compilation.

If there is no version of GTK+ installed or the version installed does not satisfy the minimum requirement, a development version of GTK+ can be installed by downloading and installing the latest version of PQL II as this download also provides for a complete installation of GTK+. The download of PQL II is available from the PQL II Download Page.

Once GTK+ has been installed, verify that the program pkg-config exists in the $PATH environment variable of the compiling user.

Compilation

Compilation is carried out through the following steps:

Define the installation directory where the PQLX system will be held on disk.

Add the following two lines to the login .rc script (for example, ~/.bashrc) for the executing user of the compilation:

```
export PQLX=PQLX-install-directory-fullpath
source ${PQLX}/env/makeVars
```

Confirm the directories have been properly defined for the MySQL installation in the file $PQLX/env/mysql.vars detailed above.

Source the .rc file modified in step 2, or open a new window.

In the uppermost PQLX installation directory, compile all source code:

```
sh> cd $PQLX
sh> env/makeALL
```

The compilation and installation will verify success. If not, a message will indicate which programs are missing (failed) and the name of the log file containing the compilation output to be consulted for investigation.

If the verification fails, the named logfile contains all compiler and linker output messages. As the PQLX system comprises many diverse components, the relevant error message(s) will be located in the logfile in the specific section of each component. The verification message, when indicating failure, will indicate which specific components (libraries and/or executables) are missing. Use this message to position yourself correctly in the ouput logfile to determine the nature of the failure.

Sever-Only Compilation

In the case where it is not desired to compile the client GUI programs on the server (if, for example, the GTK+ libraries are not easily provided for the server platform and/or the client is simply not necessary to be compiled for the server platform), it is also possible to compile only server-side programs.

In this case, follow all procedures as laid out in the section Compilation, including an additional compilation argument to env/makeAll:

```
sh> env/makeALL srvr
```

Executed in this manner, no client GUI programs will be compiled, thus requiring no GTK+ graphics files (includes and libraries) be resident on the target server machine.

Installation

Upon successful compilation, the $PQLXPROD sub-directories will contain all files required for PDF server and PQLX system execution. Once compilation is complete, the $PQLXPROD directory may be copied to another directory designated for production execution purposes.

All users who wish to execute any portion of PQLX, either server- or client-side, will require their .bashrc file to source the file PQLXprodVars thus:

```
export PQLX=PQLX-production-directory-fullpath
source ${PQLX}/PQLXprodVars
```

The file PQLXprodVars defines the following environment variables, each required for execution of PQLX and its various components in a production environment:

```
export PQLXPROD=${PQLX}
export PQLXLOG=${PQLXPROD}/log
export PQLXBIN=${PQLXPROD}/bin/${PQLXARCH}
export PQLXDBDEF=${PQLXPROD}/dbdef
```

Once compilation and installation has been completed successfully, the MySQL database is ready for PQLX initialization and usage. Please see document PQLX-DB-Setup.pdf for details.

Subsequent updates/upgrades to the system can be made by compiling the release in question, followed by copying the PROD directory and subdirectories to the local production directory system. It is important that this be a copy operation since there are files generated and installed once the PQLXsystem is initialized with initPQLXdb.sh.

If ever the $PQLXPROD directory structure is overwritten, one file must be regenerated in order for the system to be able to do its work:

$PQLXBIN/mysqlRoot.pw – this file should contain the root password for the MYSQL installation supporting PQLX. If the database initialization specified a PQLX-only install, then the password is PqLXRooT (case sensitive!). Otherwise, this password is defined locally and must be provided to this file.

For example, to recreate using the PQLX-defined root password:

```
sh>  echo 'PqLXRooT' >$PQLXBIN/mysqlRoot.pw
```

Client/GUI Installation

As for the server, once compilation succeeds, the $PQLXBIN directory will contain all executables to execute the PQLX client GUI programs. To install this on a client machine, the following steps must be carried out for each machine (instructions for each are detailed above):

Verify if GTK+ is already installed and is minimum version required (2.6.7).
If not, install PQL from the PASSCAL download web page.
Install client programs pqlx, pql, and pqlxPNG to directory of choice, making certain this directory is included in the user's PATH environment variable.

PQLX System Files

The following tables detail the binary and script executables, as well as support files currently making up the PQLX system:

Server-Specific Files	Description
$(PQLXBIN)/initPQLXdb.sh	bash shell script - initializes MySQL for PQLX usage, executed once per MySQL

	server instance.
$(PQLXBIN)/re-initPQLXdb.sh	bash shell script - re-initializes MySQL for PQLX usage, executed only when wanting to delete ALL currently defined databases. Use this command with extreme caution, ALL PDF/PQLX databases are deleted by this command!
$(PQLXBIN)/makePQLXdb.sh	bash shell script - creates a PQLX database instance, executed once per PQLX database instance.
$(PQLXBIN)/delPQLXdb.sh	bash shell script - deletes a PQLX database instance, executing user must provide PDF database password provided in PQLX database definition File.
$(PQLXBIN)/pqlxSrvr	binary executable - PDF-PQLX server program
$(PQLXBIN)/pqlxSrvr_safe	bash shell script - executes pqlxSrvr, guarantees pqlxSrvr will complete all tasks in the event queue, that is, pqlxSrvr will restart if it crashes
$(PQLXBIN)/pqlxSrvr_quit	bash shell script - shuts down the pqlxSrvr process, both the executable and script
$(PQLXBIN)/importPDFdata	binary executable – imports PSDs computed as part of the the PDF Stand-Alone system, version 1.
$(PQLXBIN)/pqlxDBMaint	binary executable – perform database maintenance on a PQLX database.
$(PQLXBIN)/rePSD	bash shell script – re-execute PSD analyses for a given channel and time range.
$(PQLXBIN)/updatePQLXdb.sh	Re-read contents of $PQLXDBDEF (PQLX database definition) file and update to the PQLX database.
$(PQLXBIN)/addDataDirectory.sh	Add a data directory to be scanned to a PQLX database.
$(PQLXBIN)/delChannel.sh	Remove a specified channel completely from a PQLX database.

Client-Specific Files	Description
$(PQLXBIN)/pql	binary executable - PQL II
$(PQLXBIN)/pqlx	binary executable - PQL-eXtended - PQL II - Waveform Viewer PDF Viewer/Analyzer STN (station) Viewer/Analyzer
$(PQLXBIN)/pqlxPNG	binary executable – produce PNG versions of PDF plots
$(PQLXBIN)/exFREQS	bash shell executable – return list of PSD frequencies for a given channel
$(PQLXBIN)/exPDFfreq	bash shell executable – return a PDF for a

	given date range and frequency range
$(PQLXBIN)/exPDFhour	bash shell executable – return a PDF for a given channel and time and month range
$(PQLXBIN)/exPSDhour	bash shell executable – return PSDs for a given channel and date and time range

Support Files	Description
$(PQLXPROD)/dbdef/PQLXdb.template.pqlx	text file - template file used for defining a PQLX database
$(PQLXPROD)/dbdef/sql/initPQLXdb.sql	SQL referenced by initPQLXdb.sh script. Responsible for setting the PQLX database users in MySQL.
$(PQLXPROD)/dbdef/sql/makePQLXMETAdb.sql	SQL referenced by initPQLXdb.sh script. Responsible for creating the PQLXMETA database.
$(PQLXPROD)/dbdef/sql/makePQLXdb.sql	SQL referenced by makePQLXdb.sh. Responsible for creating a PQLX database instance.

PDF/PQLX Server Database Setup

Introduction

Before creating a new PQLX Server Database, some preparatory actions must be carried out:

Define which channels will be contained in a single PQLX Server Database
Define the location of all PQLX Server-related data
Define which channels, if any, should be imported from a previously executed PDF standalone system
Define the PQLX Server Definition File

Data Preparation

The PQLX system is driven off the existence of data. For each PQLX database instance, one or more directories are required to be set up containing a mix of either trace files or response files for the channels whose analysis is to be held by the database.

Decide for each PQLX database which channels (or stations) data should be grouped together in a single database. Subsquent client-side queries will be limited to comparisons of data contained within this single database during any given client connection with the server.

There are no requirements as to filenames or directory structures, configuration is left entirely up to the system administrator. All trace and response files may be located under a single directory structure or may be segregated between numerous directories; how this is managed is completely left to the implementor.

Waveform Data File Formats

The following list defines the seismological file formats supported and auto-detected by PQLX. Any PQLX data directory for any given data directory may contain any combination of the following seismological trace formats:

- mini-SEED
- SAC
- SEGY
- AH
- DR100
- NANO

Response File Format

Response files must be provided in a single format, namely, the format produced by the program rdseed, (also available via direct download from the IRIS website). (This format is used for all trace file formats, not just mini-SEED. For non-mini-SEED trace file formats, modify a mini-SEED response file, providing the appropriate values for the channel in question.)

Reponse file requirements are:

- Each response file must contain information for only a single channel.
- Each response file must contain response information for a single channel over all time. That is, a single channel may not use multiple response file instances to represent responses for different time periods.

If after executing the server, no PSD data is generated, the response file is the most likely culprit. Confirm that the response file adheres to the format described here and start again.

Data Set Preparation

Before creating and initializing a PQLX database, the following decisions must be made:

Decide how proposed channels should be segregated by database, that is, decide which channels should be contained within the same database instance.

For each PQLX database, set up one or more directories containing all PQLX-related files per database, trace files, and response files. There are no hierarchy or naming requirements for this directory or the files it holds; all directories and files may be named however desired, though planning a logical hierarchy may save headaches in the future. It is recommended that a separate directory be created to hold all Response Files for all channels; this will make maintenance easier.

If importing previously computed PSDs from a PDF standalone system, decide which channels should be imported to which database. If different channels are to be imported to different PQLX databases, create an input file (to be used with program importPDFdata) containing the channel directory of PDF statistics to import. If all channels are to be imported to the same database, an input file is not necessary (see importPDFdata documentation for further details.)

MySQL/PQLX Environment Initialization

Once the MySQL database has been successfully installed and tested and the PQLX system compiled and installed, the PQLX system is ready for initialization.

The PQLX server is initialized via the following command (located in $PQLXBIN):

login-sh> initPQLXdb.sh

This initialization step is required only once per PQLX server instance and is responsible for creating the PQLX database users as well as the PQLX meta database containing the definitions of all PQLX databases held on the server.

This initialization script will ask if the MySQL database instance is dedicated to PDF usage or not. If MySQL has been provided only for PQLX usage, answering 'Y' will assume the root password has not yet been set since MySQL installation and will be reset for PQLX usage. If the MySQL database instance is shared by other applications, answer 'N' and provide the root password. In both cases, the root password is saved in the file 'mysqlRoot.pw' for subsequent usage. If the MySQL root password ever changes, the contents of the file 'mysqlRoot.pw' must also be changed accordingly.

Furthermore, when specifying the root password, it is recommended to save a copy of the file mysqlRoot.pw. Thus, when installing subsequent versions of the PQLX system, this file must be copied into the $PQLXBIN directory by hand. Otherwise, the standard version of the file is delivered and installed, replacing the local version.

PQLX Database Initialization

To seed and initialize a PQLX database instance, the following actions must be carried out:

Define a database definition file (located in directory $PQLXDBDEF) for each PQLX/PDF database instance desired. The name of this file must adhere to the following file naming convention:

dbName.pqlx
where
dbName is the name of the PDF Server Database, with the suffix .pqlx

(See an example of this file contained in $PQLXDBDEF/samplePQLXdb.pqlx.
See file $PQLXDBDEF/PQLXdbTemplate.pqlx for a template of this file.)

Edit the database definition file and provide all required information:

Database description
Organization providing database
Technical administrator name
Technical administrator email address
Data administrator name
Data administrator email address
Database type (either PRODUCTION (permanent) or SCRATCH (temporary))
Database access (either PUBLIC (all clients able to read contents) or PRIVATE (only the database creator is able to read contents))
System administrator password
WWW directory holding PQLX system-defined PDFs (or NONE to produce no WWW plots)

All fields are required; any field missing or containing invalid values will result in PQLX database creation failure.

Create the PDF database using the script $PQLXBIN/makePQLXdb.sh, specifying all directories defined in Data Preparation Section, Item 2 above (execute with no arguments for Usage help).

If importing previous PDF data, execute program $PQLXBIN/importPDFdata, specifying either a directory or input file of directories to import (execute with no command line arguments for usage help). See server documentation PQLX-Server.pdf for a complete discussion.

Once a PQLX database instance has been created, the PQLX server is ready for execution. See server documentation PQLX-Server.pdf for details.

Other Database Helper Scripts

delPQLXdb.sh – To delete a previously created PQLX database.

updatePQLXdb.sh – To re-read the contents of the PQLX database definition file (created in step 1, section PQLX Database Initialization above), and update the database itself with the new information.

PQLXDataDirectory.sh – Allows a PQLX administrator to maintain the data directories defined for a particular database. Actions include listing all data directories currently defined, adding an additional data directory, deleting a data directory, and enabling and disabling the reading of a data directory. Execute without arguments for complete details. This can only be executed on the server machine itself.

listPSDchannels.sh – Provides a list of all PSD channels currently defined for a database.

listQErrors.sh – Outputs all errors currently held by the database. Errors include, but are not limited to, missing response files, trace files that do not overlap, and trace files not containing enough data for processing.

PDF/PQLX Server Execution

Introduction
Once a PQLX/PDF database has been initiated, the server is ready for execution.

The PQLX server process is executed to act on behalf of a single PQLX database instance and is responsible for carrying out the following actions:

Inspect all files in the directories defined with makePQLXdb.sh, identifying all new files since last execution (both trace files and response files);

Analyze all new files, writing all analysis statistics back to the PQLX database. The current analysis statistics being computed are:

- TRACE Statistics - trace file characteristics: Start/Stop times, number of samples, MIN, MAX, and MEAN amplitudes, and, for mini-SEED format files only, number and location of gaps and overlaps.
- PSD Statistics - computed according to algorithm laid out by D. McNamara and R. Buland (see discussion document as well as other references here), used for subsequent client-side PDF display.
- Additional global statistics for each channel over all time are also maintained.

Execution is carried out via the following command:

sh> $PQLXBIN/pqlxSrvr_safe [srvr:]PQLXdb-name numCPUs
where:

[srvr:] is the name of the server machine hosting the PQLX database, where the MySQL database server is running. No specification defaults to localhost, the MySQL database server and the pqlxSrvr are executed to run on the same machine.
PQLXdb-name is the PQLX database name to execute against.
numCPUs is the number of CPUs the server should execute on.

Executing the server with this script executes the program pqlxSrvr itself. The server program guarantees that the server will complete, automatically restarting itself if it crashes during the course of its work. If a crash occurs, a message will be output indicating this happened as well as the filename containing a list of the traces the server was processing when the crash occurred. In this manner, follow-up investigation is easy. Also, the list of trace files causing the crash may be used for any further script processing, to easily move all traces to a "side" directory for later investigation and/or to remove them from the data directory of valid or processable trace files. (On the other hand, once a file has been processed, it will not be reprocessed unless the time-stamp on the file changes.)

The pqlxSrvr program is event-driven, reading from the database the next action it is to carry out. The order of general operations is:

Traverse all PQLX-defined data directories identifying all new instances of data files and inserting this identity information into the database.

Fork numCPUs event processors, performing all currently defined data analyses on the newly identified trace files, and storing all results back to the database.

Traverse again all PQLX-defined data directories identifying all new files since completion of step 1.

This re-traversal is done so that the server can "catch up" to the current date of available/defined data files. Initial execution of the server (depending on the amount of data, speed of executing machine, number of used CPUs) may require days to complete. In the meantime, new data files may have been created, which now require processing. This guarantees that once the server completes the database, it will accurately reflect all currently existing data.

In addition, any trace files receiving real-time data transmission will be identified only one time per execution of the server on the same day. That is, they will not be "re-identified" as part of this step, thus avoiding a rather inconvenient infinite loop.

Loop over steps 2 and 3 until all identified files have been analyzed and no new files have been found.

Update channel statistics held on the database.

Update all System PDF plots for channels having had PSD calculations performed in step 2.

Currently, the following so-named channels are automatically tagged/identified to have PSD calculations performed: BH*, LH*, HH*, EP*, BL*, HG*, HL*, BG*, BN*, SH*, and EH*. Other channels can be added to this list very simply by adding an entry to the table PSDCFG. Insert a new entry into the table providing all definition information as for other entries.

If the dbWWWDIR entry is defined (as part of the dbName.pqlx database definition file), all updated system PDFs are also output to the defined directory as PNG plots.

pqlxSrvr_safe and Cron Execution

Setting up a cron job to execute the server is relatively straightforward. The execution of the server script pqlxSrvr_safe requires the environment variables defined in $PQLXPROD/PQLXprodVars to exist for execution (see the PQLX-Installation document for details on setting up a production-only environment). Make a bash script which executes at a minimum the following:

```
#!/bin/bash

export PQLX=/full/path/to/PQLX/PROD/directory
source $PQLX/PQLXprodVars
$PQLXBIN/pqlxSrvr_safe PQLX_DB numCPUs
```

Additionally, this script may contain any other commands required as part of an automatic execution of the PQLX server. For example, if data should be extracted from some data source before the PQLX server is executed, this could be provided within the same script. In such a way, if there ever exist dependencies whereby actions must be carried out in a specific order, these can be easily met by simply executing these actions serially within the same script. For example, a cron job script, named PQLXexecute, to extract data and execute the server on 4 CPUs on database PQLX_DB, might be:

```
#!/bin/bash

# extract yesterday's data (or all data since last extract)
# local command here to perform data extraction

# execute the PQLX server
export PQLX=/full/path/to/PQLX/PROD/directory
source $PQLX/PQLXprodVars
$PQLXBIN/pqlxSrvr_safe PQLX_DB 4

# perform any follow-up activities necessary
```

It would be the above script that would be specified for cron execution. For a daily update of a PQLX database, it is recommended that the server be executed locally after 00:00 GMT. In this manner, all trace file information is valid for the current day.

pqlxSrvr – Direct Execution

As noted, using the script pqlxSrvr_safe is intended for automatic production environment purposes and normal executions of the server. It is possible, however, to execute the server analysis program directly. This can be done in the case when wanting to re-analyze data files previously analyzed. If, for example, response file information has changed for data in the past that have already been analyzed (or, said otherwise: the PSD data are incorrect due to incorrect response information previously provided), an optional argument to the pqlxSrvr program allows these files to be defined for re-analysis.

Executing the pqlxSrvr program directly takes the following arguments (execute with no arguments for usage details):

```
--dbName=[SERVER:]pqlxDB  PQLX database name - required
--numCPU=#                Number of CPUs to use - optional (default=1)
--identFile=path-to-file  Filename listing traces to be Re-analyzed - optional
```

Providing a file of traces via the –identFile argument will force the server to reanalyze only the trace files listed in this file, replacing all previous analysis information. The format of this file is one trace file (full pathname) per line.

No scanning of the data directories is done as part of this execution. Once the re-analysis is complete, all channel statistics are updated and system PDFs recomputed.

Logfile Ouput

The following log files are created as part of the execution of pqlxSrvr:

- srvr.dbName.YYYY.JJJ.HH:MM.log – standard log file providing information related to the normal course of execution.
- srvr.dbName.YYYY.JJJ.HH:MM.err – error file holding any error messages generated during the course of execution. If pqlxSrvr is executed via pqlxSrvr_safe, this file is removed if empty once the server completes.
- srvr.dbName.YYYY.JJJ.HH:MM.crash – file containing, one per line, the full pathnames of all trace files that caused the server to crash in the course of its operations. For problem data, this allows for easy follow-up investigation. If pqlxSrvr is executed via pqlxSrvr_safe, this file is removed if empty once the server

completes.

It is recommended that these files be variously consulted to ensure that system operation is running smoothly and as expected/required.

Other Server-Side Programs

In additon to the main server analysis program, other server-side programs are provided:

pqlxPNG – generate PNG format plots of PDFs.
pqlxDBMaint – a database maintenance program
rePSD – re-execute PSD computations for a particular channel and time range
importPDFdata – import PDF data previously computed by the PDF Standalone system

pqlxPNG

This program generates PNG plots of PDFs. Execution may be either automatic (as part of a cron job), or manual (from the command line), and may be executed directly on a server machine or from a client machine.

Two types of executions are possible. Option 1 generates PNG plots for all System PDFs currently held by a PQLX database. Option 2 generates a single PNG plot for a user-provided input file.

Option 1

To generate PNG plots of all System PDFs, command-line usage is the following:

$(PQLXBIN)/pqlxPNG --systemPDF --dbName=[SERVER:]pqlxDB [--pngDir=/path/to]

Where –systemPDF and –dbName= arguments are both required. If executing from a client machine specify the server machine name or IP address (where the PQLX database resides) as part of the dbName parameter (--dbName=SrvrName:MYNETWORK).

The argument --pngDir= is optional. If not specified, pqlxPNG outputs the PNG plots to the directory specified by the dbWWWDIR entry in the PQLX database definition file. If this is not defined, then pqlxPNG will quit without doing anything. In this case, specifying a directory using the --pngDir argument will output the PNG plots to the directory specified.

Option 2

To generate a single PNG plot based on user-provided input, command-line usage is the following:

$(PQLXBIN)/pqlxPNG --inputPDF --pngName=title --pngDir=/path/to

where all arguments are required. In this case, an input file must be provided for PDF generation. Typically, this is done using the output of one of the PDF extract routines provided in $PQLXBIN and piping this to pqlxPNG. These extract routines are one of the following: exPDFfreq and exPDFhour (execute without arguments for complete usage details).

Alternatively, this file may be generated by the user. The format of this file is the first line containing PDF definition information followed by the PDF values. Create an output file using one of the PDF extract routines above and simply match the format.

Other Options

Other options affecting the output of pqlxPNG:

--width=
--height=
--noBorder

Where --width= and --height= indicate, in pixels, the exact width and height of the PNG plot to be generated, and --noBorder indicates that no border should be drawn around the plot itself.

Providing a width of less than 240 pixels also indicates that a thumbnail version of the PDF plot will be made. In this case, no Y-axis (dB scale) and no color bar are drawn. This is useful for creating small PDF plots to be used for a web page implementation.

pqlxDBMaint

Over time, a PQLX database, depending on exact usage, may become fragmented, causing the pqlxSrvr program to gradually require more and more time to execute. It is recommended that for normal installations, the pqlxDBMaint program be executed at least once per month. For installations that execute the pqlxSrvr program daily against trace files receiving real-time data, it is recommended the pqlxDBMaint program be executed once per week.

This should be an automatic execution, most likely part of a script executed via cron (see pqlxSrvr and Cron Execution above).

Usage:
$(PQLXBIN)/pqlxDBMaint [OPTION...] - Execute pqlxDBMaint
Application Options:
--dbName=[SERVER:]pqlxDB PQLX Database name – required

Example execution:

bash> $PQLXBIN/pqlxDBMaint –dbName=MYNETWORK

rePSD

One method of forcing a recomputation of all PSDs for a given channel and date range is via the shell script rePSD. This will delete all previously computed PSDs for the given channel and date range and create PQLX server events to reanalyze the corresponding trace files.

Usage:
$(PQLXBIN)/rePSD [HOST:]DBName NTW STN LOC CHN START-DATE END-DATE [numCPUs]

where:

START-DATE & END-DATE define the time period for which PSDs should
be recomputed (inclusive)
format: YYYY-MM-DD

numCPUs: optional parameter specifying the number of CPUs to use when

reinitiating the server.
Not specifying numCPUs will delete the PSDs and recreate pqlxSrvr events
for subsequent processing, but WILL NOT reinitiate the server.
This is useful when wanting to delete several ranges of PSDs
(in time or by channel) before reinitiating the server.

examples:

$(PQLXBIN)/rePSD micros IU ANMO -- BHE 1996-12-01 1998-02-01

to delete the PSDs for channel IU.ANMO.--.BHE between 01-DEC-96 and 01-FEB-98
and create PQLX events for recomputation, but DO NOT execute the server

$(PQLXBIN)//rePSD micros IU ANMO -- BHZ 1999-04-01 1999-06-01 4

to delete the PSDs for channel IU.ANMO.--.BHZ between 01-APR-99 and 01-JUN-99
and reinitiate the server to process all outstanding PQLX events, to execute on
4 CPUs

ImportPDFdata

Data previously computed using the former version of PQLX, the PDF standalone system, may be imported to a PQLX database using this program.

Usage:

$(PQLXBIN)/importPDFdata [options ...]
Application Options:
--dbName=[SERVER:]pqlxDB PQLX Database Name - required
--statsDir=path-to-directory PDF-SA Statistics Directory - (this or --dirsFile)
--dirsFile=path-to-file File of Directories to import - (this or --statsDir)

To specify that all PSD data should be imported from a PDF Stand-Alone installation, specify the STATS directory with –statsDir.

To specify that particular stations of PSD data should be imported from a PDF Stand-Alone installation, create a file listing, one per line, the STATS/STN directories to be imported. Execute importPDFdata specifying this file of directories with –dirsFile.

System PDFs are not computed as part of importPDFdata, system PDFs will be computed as part of the next execution of pqlxSrvr.

PQLX Client Execution

Once the server has completed its initial execution, the PQLX database is ready for reading by the client GUI, pqlx. The client may be executed as:

sh> $PQLXBIN/pqlx

Once started, different databases (and/or servers) may be connected. Please see the document PQLX-Client.pdf for a quick overview of the client program's functionality and usage.

PQLX Client Programs

Introduction

Once a server instance has been started, the PQLX system client application pqlx may be executed. This document is a quick usage guide and overview of the functionality provided by the following programs:

pqlx – GUI program to visualize database data
pqlxPNG – program to produce PNG plots of System and User PDFs

pqlx

The client program pqlx is the GUI interface used to read and visualize both the data stored in the PQLX database on the server as well as any trace files to be viewed. These trace files may be either files that have been analyzed by the server, or simply trace files that exist on disk that have not been analyzed; no requirement regarding: pqlxSrvr and trace files exists.

Startup

Execution of the client program is simply:

sh> $PQLXBIN/pqlx

start and render the GUI interface to the screen.

Available Data Display Systems

Via the pulldown menu in the upper left-hand corner of the screen, three data display systems are available:

- PQL II - view trace data in detail (this does not utilize a connection to a PQLX database)
- PDF Viewer - view PDF plots of PSD data stored in a PQLX database
- STN Viewer - view data and their availability by station and channel, utilizing the PQLX database connection

PQL II

A broad overview of the functionality provided by PQL II:

- Trace Tab - view trace data as individual trace files, data passed from the PDF viewer, or data passed from the STN viewer
- Magnify Tab - Zoom In and Out
- Spectra Tab - display and Fourier transform the selected data segments
- Split Tab - display multiple data views simultaneously
- Header Tab - display all data header information

More specific and detailed help and usage information can be found via the Help button found on the sidebar of the Controls Panel.

PDF Viewer

Upon initial execution, pqlx is not connected to any specific database. In order to view PDF plots, a connection to a database must be made. This is achieved via the Servers button available on the sidebar for the Main tab. The first time a server (not the localhost) is connected, the IP address or fully qualified hostname must be provided. Once a successful connection to the machine is made, double-clicking on the hostname in the left-hand display pane will list all available PQLX databases in the right-hand display pane.

(Connection requirements for pqlx and the server database: MYSQL communicates over port 3306. If you can ping the server while an attempt to connect to a database fails, confirm that the PQLX database server and client machines are both properly permissioned to allow open communications across port 3306.)

To connect to a specific PQLX database, double-click on the database name in the right-hand display pane. Once successfully connected, all required data will be downloaded and the GUI will build-up its various components for user interaction (all pull-down menus as well as internal data structures).

Main Tab

Display system PDFs that were previously computed during the most recent execution of the pqlxSrvr program.

Display By options:

Station - Display three user-selected system PDFs for a particular station and channel group
PDF - Display a system PDF for three individual stations and common channel group
Both - Display three system PDFs for each of three individual channels of specific stations

Available System PDFs:

ALL - PDF of all PSDs
Month - PDF of last month of available PSDs (previous 30 days)
Week - PDF of last week of available PSDs
L Month - PDF of previous month of available PSDs (previous 60 - 30 days)
L Week - PDF of previous month of available PSDs (previous 14 - 8 days)
Year - PDF of current year's PSDs
L Year - PDF of last year's PSDs

Mouse Functions:

HOVER - Lower right hand corner of PDF plot:
display Probability color Bar

CLICK - on PDF plot:
take PDF to Detail Tab

Detail Tab and Panes

NW Pane - Main PDF of Detail Tab

SW Pane - PDF of subselect of Main PDF. This PDF is made up of all PSDs intersecting the point or region defined by the NW Detail pane subselect

NE Pane - displays the first 15 traces representing the PSDs of the NW pane subselect

SE Pane - displays the start time and duration of all PSDs returned by the NW pane subselect

Mouse Functions:

NW Pane - Click:
define subselect point

NW Pane - Click+Drag
define subselect region

SW Pane - Click:
make subselect PDF the new Detail Main PDF, move to NW pane

NE Pane - Click:
Take all traces of subselect to PQL

Request Tab and Panes

Each display pane allows for display of a user-specified PDF based on date and time range parameters. For each of the display panes, selecting the corresponding button in the sidebar allows for the following types of PDF data requests to be rendered:

Range - a simple request allowing the user to specify start and stop values corresponding to the PSD values for year, day, and hour. Any field left blank will default to the boundary as defined in the database. Wrapping around boundaries is properly understood. For example, specifying a start day of 1 December and end day of 31 January will render a PDF of all PSDs for the months of December and January.

Frequency - a more complicated type of request allowing the user to specify the number of days for which PSDs should be extracted, followed by the number of days to be skipped. So, for example, it is possible to create a PDF of all weekdays, skipping weekends, or vice-versa. Rather more complicated requests are also possible, for example, a PDF of only the week before and week after a full moon, but only during the hours in which the moon is within 90 degrees above the station's position on Earth.

Ago - a request specifying a relative time range in terms of days ago. For example, a PDF of all PSDs occuring between 45 and 15 days ago.

Mouse Functions:

CLICK - on PDF plot:
take PDF to Detail Tab

STN Viewer

As with the PDF Viewer, once a database has been connected, either via the PDF viewer or the STN viewer, all GUI menus and internal data structures are appropriately populated.

The STN (station) viewer allows for display of data coverage or data organized by channel; each channel of data is displayed horizontally with the X-axis corresponding to individual days. Various settings via the Controls panel and sidebar provide for a display configuration corresponding to a user's precise requirements, for example, colours, maximum number of channels to display, number of days, and so forth.

Via the Display options available on the sidebar, two ways of displaying data are possible:

Coverage - Reading from the database, the coverage mode simply displays the existence of data as well as location and duration of gaps and overlaps. Small vertical ticks indicate the physical boundary of the actual data files. This is useful when the client does not have direct access to the trace files for display. Used in combination with the PDF display (described in Mouse Functions below), the user can still get a good indication of the quality of the data.

Data - In Data mode, actual data files are read from disk and displayed, much like in PQL. Unlike in PQL, however, all displayed data are internally held as a single trace. This is important when, for example, applying a filter; no effects of data discontinuity (physical data file boundaries) will be suffered.

Mouse Functions:

Stats Label:
HOVER:
Display statistics for channel for date range on display
Request PDF from database for channel

PDF Label:
HOVER:
Display PDF for channel of date range on display
CLICK:
Take PDF plot to PDF Viewer, Detail Tab

In Plot Region:
SHIFT+CLICK+DRAG
Take selection(s) to PQL, removing any previous selections in PQL
CTRL+CLICK+DRAG
Take selection(s) to PQL, adding to any previous selections already present in PQL
a-key+CLICK+DRAG
Take selection(s) to Analysis Tab, removing any previous selections
s-key+CLICK+DRAG
Take selection(s) to Analysis Tab, adding to any previous selections already present

The action of taking the selection(s) to PQL and the Analysis Tab happens on Key-UP of the selection keys. In this manner, multiple selections are possible.

On STN Label:
As for SHIFT and CTRL keys above:
Select entire station on display

On Channel Label:
As for SHIFT and CTRL keys above:
Select entire channel on display

Between Day Labels (X-axis region):
As for SHIFT and CTRL keys above:
Select entire day of data on display

pqlxPNG

The client program pqlxPNG provides the ability to create .png graphical format plots of both System PDFs and user-defined PDFs.

Using this program, it is very easy to create .png format output files of all System PDF plots, writing these to a directory that may be accessed via a web browser (LAN or WAN). This allows access to pqlxSrvr results to a broader category of end-user than simply those who have the GUI client pqlx installed:

those users who do not have the GUI client pqlx installed on their machine

those users not having access to the database itself, for example, external users who are not permissioned to access the LAN on which the server resides.

Also, in providing .png format output, it is very trivial to create graphical images that can be used for publication.

Important Note

Since the program makes use of GTK+ and graphics, execution of pqlxPNG must occur on a computer (and within a shell) that is aware of a visual display, a monitor. If the server machine, for example, does not have a display associated with it, this program will not work. Due to this restriction, it is highly recommended that this program be considered a client-side program and be executed only on a computer that has also demonstrated ability to execute the GUI client program pqlx.

Usage

pqlxPNG [OPTION...] - Make PNG Plot(s) of PDF Data

Application Options:

--width	Width of plot (in pixels, default = 550) - optional
--height	Height of plot (in pixels, default = 425) - optional
--noBorder	Draw no border around PDF plot - optional

--systemPDF	Make PNG plots of System PDFs...
--dbName=[SERVER:]pqlxDB	For database pqlxDB on SERVER - required
--pngDir=/path/to	Directory where to place PNG files - optional
--inputPDF	Make PNG plot of PDF provided via stdin.
--pngName=90TH	Name of PDF to provide in PNG filename - required
--pngDir=/path/to	Directory where to place PNG files – required

Options –width, --height, and –noBorder control the format aspects of the PDF plot and are self-explanatory.

Two types of PDF plots can be made with pqlxPNG:

All System PDF plots for a given database
PDF plot of data supplied as input via stdin by the user

System PDF Plot Generation

Specifying the option --systemPDF, all System PDFs defined for a given database will be generated. In this case, the option --dbName is required to specify for which database on which server the .png plots of System PDFs are to be generated.

If the optional argument --pngDir is omitted, the output will be put to the directory specified by the field DBWWWDir in $PQLXDBDEF/dbName.pqlx. If this field is defined to NONE, execution of pqlxPNG will fail. Define the field DBWWWDir and update this information to the database server using updatePQLXdb.sh and execute again. Alternatively, you may specify the option --pngDir to specify this output directory.

User-Supplied PDF Plot Generation

Specifying the option --inputPDF, a PDF plot will be created using the PDF information provided to pqlxPNG via stdin. This option is intended to be used in conjunction with the PDF data extract routines exPDFfreq and exPDFhour (see document PQLX-Extract-API.pdf for details).

For this usage, two additional options are required: --pngName and –pngDir. Option --pngName defines the name to be used when generating the .png output file name. Option --pngDir defines the directory where the .png files are to be output.

Example usages of this option include:

```
sh> exPDFfreq PQLXdb IU ANMO -- BHE 2007-01-01 2007-03-31 1.0 180.00 \
    -200 -50 -PNG | pqlxPNG --inputPDF --pngName=2007Q1 --pngDir=/tmp

sh> exPDFhour PQLXdb IU ANMO -- BHE 1 3 00:00 24:00 -PNG | pqlxPNG --inputPDF \
    --pngName=ALLQ1 --pngDir=/tmp
```

If producing your own PDF data for plotting, the required input format of PDF data for this option is the following:

```
NTW.STN.LOC.CHN SYYY SJ EYYY EJ numPSD
freq<TAB>power<TAB>number of hits
```

where the first line of the file is:

NTW.STN.LOC.CHN is the fully qualified channel name
SYYY – is the start year of the PDF
SJ – is the start julian day of the PDF
EYYY – is the ending year of the PDF
EJ – is the ending julian day of the PDF
numPSD – is the total number of PSDs represented by this PDF

and each succeeding line of the file is:

freq – the frequency
power – the power
number of hits – the number of occurrences of PSD data for this frequency and power combination.

An example of this format may be easily generated via execution of a data extract API script and specifying the -PNG option.

Output

For both output options, the directory structure and filenaming conventions are the same. All .png files are output to a subdirectory of pngDir. This subdirectory is named NTW.STN, where:

NTW and STN are the network and station of the channel being plotted

All .png files are named NTW.STN.LOC.CHN.name.png, where:

NTW.STN.LOC.CHN is as one would expect

name is the name of the PDF plot. For –systemPDF execution, this is the name of the System PDF. For --inputPDF execution, this is the name supplied by the argument –pngName.

pngcnt

A helper script, pngcnt, is provided to count the number of .png plot files existing in a png directory.

Usage

pngcnt directory

will return the total number of PNG plots existing in all subdirectories of directory.

PQLX PDF and PSD Data Extract Scripts

Introduction
In addition to PSD and PDF data visualization provided by the client GUI program pqlx, several scripts are available to allow for extraction of PSD and PDF data directly from a PQLX database. This is provided to allow for the case when further analytical processing might be desired and where this requires raw data as opposed to a picture.

The following extract scripts are provided:

exPDFhour – extract the PDF for the given channel and bounding parameters: month range, and start and stop times

exPDFfreq – extract the PDF for the given channel and bounding parameters: date range, frequency, and power

exPSDhour – extract the PSDs for the given channel and bounding parameters: date range, and start and stop times

exPDFhour

Returns the PDF for the given channel and bounding parameters.
Output Format: Frequency Power number_of_hits

Usage:

exPDFhour [HOST:]DBName NTW STN LOC CHN \
start_MONTH end_MONTH start_HOUR end_HOUR [--PNG]

Examples:

sh> exPDFhour dbName IU ANMO -- BHE 1 1 02:00 14:00

sh> exPDFhour dbName IU ANMO -- BHE 1 12 00:00 24:00

Extract to pqlxPNG example:

--PNG option indicates that output should be made in a format intended as input to pqlxPNG plotting program.

This output can be piped to pqlxPNG as:

exPDFhour ... -PNG | pqlxPNG --inputPDF --pngName=name --pngDir=/tmp

See pqlxPNG for a complete list of possible options to pqlxPNG.

exPDFfreq

Returns the PDF for the given channel and bounding parameters
Output Format: Frequency Power number_of_hits

Usage:

exPDFfreq exPDFfreq [HOST:]DBName NTW STN LOC CHN \
Start_DATE End_DATE Start_FREQ End_FREQ Min_POWER Max_POWER [-PNG]

Examples:
sh> exPDFfreq micros IU ANMO -- BHE 2003-04-03 2004-05-03 8.0 8.5 -180 -60

Extract to pqlxPNG is same as for exPDFhour.

exPSDhour
Usage
exPSDhour DBName NTW STN LOC CHN Start_DATE End_DATE Start_TIME End_TIME

example:
sh> exPSDhour micros IU ANMO -- BHE 2003-04-03 2004-05-03 02:00 14:00

returns the PSDs for the given channel and bounding parameters
format: DATE HOUR Frequency Power

PQLX Bug Reporting

The web address for reporting all PQLX-related issues, be it a fatal bug, interface annoyances, comment on lack of documentation, or enhancement request, is:
http://wush.net/bugzilla/PQLX

Following a standard Bugzilla installation, from the home page you have a choice of three PQLX issue-related actions:
Search existing bugs

Submit a new bug
Summarize bugs in a report

Actions 1 and 3 do not require a Bugzilla Login, these can be executed either anonymously or from a logged-in state.

Submitting a bug, however, requires a Bugzilla Login. This login is the email address where you would like to receive updates on your reported bugs. The password will be saved server-side as well as client-side (if you choose to allow saving cookies from this website), it is advised to use a nontrusted password for this purpose. To create a bugzilla login, choose "Open a New Bugzilla Account" just under the Login: entry area.

Standard Bug Reporting Procedure

Generally, the following actions are carried out when wanting to report a bug (this assumes a login has been previously created):

- Search the existing bugs and determine if your issue has already been reported by someone else.
- If YES, call up the existing bug, and, after perusing the current state of known information and/or its status, determine if you possess any new information that would be helpful if added.
- If YES, add all new information to the 'Additional Comments' section (or new attachments, or whatever is possible) and Commit the change. You may also add yourself (and any other known bugzilla user) to the CC: list to receive all updates to this bug in the future.
- If NO, then nothing further is to be done: the bug is reported and known --> EXIT SYSTEM.
- If NO --> Proceed to Step 2.
- Enter a new bug report
- Please provide all relevant information when submitting a bug report. The more information that is provided, the more complete the picture will be describing the various aspects of the state of things at the time the bug or issue exhibits itself. Thus, the more likely it is that the developer will be able to recreate the problem. This is the most fundamental aspect of bug-solving: if the developer cannot recreate the problem, it cannot be solved.
- Once satisfied that all fields have been properly set and the description is as complete and accurate as possible, select 'Commit'. This will generate an email to be sent to the developer assigned to fix the bug.
- On the next-displayed screen you will be given the opportunity to add yourself to the CC list of email addresses to receive updates on this bug in the future, if you so choose.

Your bug has been submitted, and if you have chosen to do so, all future updates to the bug will generate an email to you detailing the update. If at any time you wish to be removed from receiving updates for a given bug, simply call up the bug and remove yourself from the CC: list of emails receiving updates.

Products
Currently, there are two products for which bugs can be reported:

- PQL II
- PQLX

PQL II is a separate entity from PQLX. More specifically, if the bug exhibits itself within the Trace Viewer portion of PQLX, then the bug should be reported under product=PQLX and component=PQL. Bugs should be reported under product=PQL II only when executing PQL II itself, as 'pql', not 'pqlx'.

Other Information
By default, no email is sent to the reporting user when reporting a bug (all subsequent bug events do generate emails). If you would like to receive an email of the bug when you report it, the following instructions may help:

Once logged in, at the bottom of the page follow the hyperlink Prefs under the Edit: list of options.
Under the Tab Email Preferences, you will find all sorts of options pertaining to when you will receive

an email.
The 2nd line from the bottom, described as "The bug is in the UNCONFIRMED state", you will notice is completely unchecked, indicating no emails will be sent when a bug is input.
If you would like to receive an email when entering a bug yourself, check the Reporter column;
If you would like to receive an email when being added to the CC: list of a newly reported bug, check the CCed column.
And while you're here, have a look at all the other email settings and options and modify as per your wishes.

PDF Server and PQLX System Credits

The PDF server and PQLX system has been a joint effort of many people and parties. All software comprising the PQLX system is open-source, including PQLX itself. Funding for various aspects of the system have come from multiple sources. This document attempts to list all organizations ultimately making this software possible.

Third Party Software

All software not written exclusively for the PQLX system is freely available elsewhere on the internet; the following table defines all third-party software employed, in whole or in part, and where it may be obtained:

Name	Description/Function	Download
MySQL	Database Server - server side	MySQL Download
gdbm	Database Interface - client side	gdbm Download
GTK+	Graphical User Interface library	GKT+ Download
libmseed	mini-SEED file format reader	IRIS
evalresp	SEED Response file reader	IRIS
vfbb	DR100 Data Format reader	no download
FFTW3	Fourier Transform library	FFTW3 Download
High Resolution Bi-Cubic Spline Interpolator	PDF Image Magnification	Discussion and Download
Probability Density Function Algorithm	Algorithm used to produce PDF plots	Discussion

Funding

The following funding sources (listed in alphabetical order) have made this development possible:

IRIS Data Management Center, Seattle, Washington.
IRIS - PASSCAL Instrument Center, Socorro, New Mexico.
National Science Foundation, Washington, D.C.
United States Geological Survey - Golden, Colorado.

PQLX Source Code Organization

Directory Structure

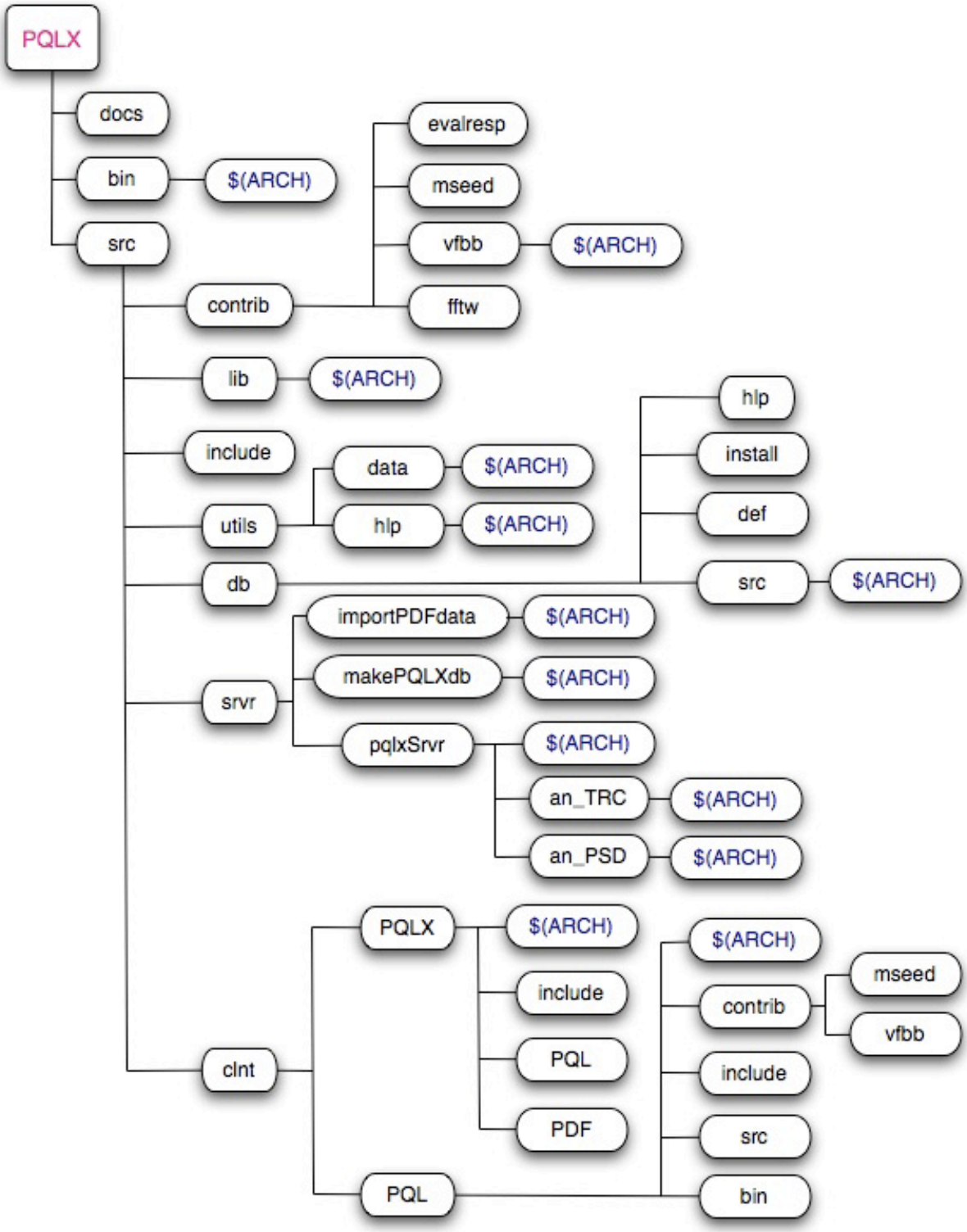
Directory	Description
project/PQLX	Base Directory
project/PQLX/docs	Important documents related to PQLX
project/PQLX/bin/${ARCH}	Location of all PQLX executables
project/PQLX/src	Uppermost source code directory
project/PQLX/src/include	Include files
project/PQLX/src/contrib	Source code contributions (written by an external party)
project/PQLX/src/utils	Utilities libraries
project/PQLX/src/clnt	PQLX client-side viewer applications
project/PQLX/src/clnt/PQL	Trace File Display application (PQL II)
project/PQLX/src/clnt/PQLX	PQLX GUI application - PQL II, PDF Viewer, STN Viewer
project/PQLX/src/srvr	PQLX server-side programs
project/PQLX/src/srvr/pqlxSrvr	PQLX analysis program – provides all analysis statistics to DB
project/PQLX/src/srvr/pqlxSrvr/an_TRC	Trace analyzer: META info, gaps, overlaps, and so forth.
project/PQLX/src/srvr/pqlxSrvr/an_PSD	PSD analyzer
project/PQLX/src/srvr/pqlxSrvr/script	pqlxSrvr related scripts
project/PQLX/src/srvr/importPDFdata	Import PDF stats to a PQLX DB from a flat-file instance of the former standalone version of the PDF system
project/PQLX/src/db/mysql	PQLX MySQL database interface; this database is server-side
project/PQLX/src/db/mysql/sql	MySQL Database definitions: tables, triggers, stored procedures, and so forth.
project/PQLX/src/db/mysql/src	MySQL db interface source code
project/PQLX/src/db/mysql/api	MySQL scripts for data extraction
project/PQLX/src/db/mysql/script	MySQL scripts for database management
project/PQLX/src/db/mysql/upgrade	MySQL scripts to upgrade versions of database
project/PQLX/src/db/gdbm/src	GNU db interface source code; this database is client-side, all tables stored in directory ~/.pqlx

Contribution Source Code
(all located in directory project/PQLX/src/contrib)

Directory	Description
.../fftw	Fast Fourier Transform library provided by http://www.fftw.org
.../gdbm	GNU Database Manager provided by

	http://ftp.gnu.org/gnu/gdbm/gdbm-{version#}.tar.gz
.../libmseed	mini-SEED file reading routines provided by http://www.iris.edu/pub/programs/libmseed-{version#}.tar.gz
.../vfbb	DR100 file format reading routines
.../evalresp	Response File reader library provided by http://www.iris.edu/pub/programs/evalresp-{version#}.tar.gz

PQLX Source Code Tree


PQLX
docs
bin
$(ARCH)
src
contrib
evalresp
mseed
vfbb
$(ARCH)
fftw
lib
$(ARCH)
include
utils
data
$(ARCH)
hlp
$(ARCH)
db
hlp
install
def
src
$(ARCH)
srvr
importPDFdata
$(ARCH)
makePQLXdb
$(ARCH)
pqlxSrvr
$(ARCH)
an_TRC
$(ARCH)
an_PSD
$(ARCH)
clnt
PQLX
$(ARCH)
include
PQL
PDF
PQL
$(ARCH)
contrib
mseed
vfbb
include
src
bin