# 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:
1. LINUX
2. Mac OS
3. 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;
- PQLX Database definition and creation, executed once for each database required.

## Server-side Technical Setup

Server-side technical environment setup is comprised of 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.

N.B. 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 comprised 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 Prerequisites

### 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 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:

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

If this returns a version greater than or equal to 2.14, 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 according to the platform-specific standard procedures.

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:

1. Define the installation directory where the PQLX system will be held on disk and de-tar the PQLX source distribution to this directory.
2. Add the following two lines to the login .rc script (e.g., ~/.bashrc) for the executing user of the compilation:

   ```
   export PQLX=PQLX-install-directory-fullpath
   ```

3. Source the .rc file modified in step 2, or open a new window.
4. In the uppermost PQLX installation directory, compile all source code:

   ```
   bash> cd $PQLX
   bash> 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 output 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:

```
bash> 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 and sub-directories 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}/PROD/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=${PQLX}/PROD
export PQLXLOG=${PQLXPROD}/log
export PQLXBIN=${PQLXPROD}/bin/${PQLXARCH}
export PQLXDBDEF=${PQLXPROD}/dbdef
export PQLXXML=${PQLXPROD}/XMLINPUT
export PQLXTRASH=${PQLXPROD}/trash
```

Once compilation and installation has completed successfully, the MySQL database is ready for PQLX initialization and usage. Please see documents **PQLX-1_Getting_Started.pdf** and **PQLX-2_DB_Setup_&_Prep.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 PQLX system is initialized with **initPQLXdb.sh**.

### Client/GUI Installation

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):

1. Verify if GTK+ already installed and is minimum version required (2.14).
2. If GTK+ is not installed, or is a version less than 2.14, install or upgrade to a more recent version.
3. 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 held on specified MYSQL server are deleted forever! |
| $(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 from within a shell |
| $(PQLXBIN)/listQ.sh | bash shell script – list status of EVENTQ in various formats |
| $(PQLXBIN)/pqlxDBMaint | binary executable – perform necessary database maintenance on a PQLX database. |
| $(PQLXBIN)/rePSD | bash shell script – re-execute PSD analyses for a given channel and time range. |
| $(PQLXBIN)/refreshSysPDF.sh | bash shell script – create events to recompute all System PDF's. |
| $(PQLXBIN)/updatePQLXdb.sh | Re-read contents of $PQLXDBDEF (PQLX database definition) file and update to the PQLX database. |
| $(PQLXBIN)/PQLXDataDirectory.sh | Manage data directories for scanning by a PQLX database. |
| $(PQLXBIN)/delChannel.sh | Remove a specified channel completely from a PQLX database. |

| **Client-Specific Files** | **Description** |
|---|---|
| $(PQLXBIN)/pqlx | binary executable - **PQL**-e**X**tended -<br>PQL II - Waveform Viewer<br>PDF Viewer/Analyzer<br>STN (station) Viewer/Analyzer |
| $(PQLXBIN)/pqlxPNG | binary executable – produce PNG versions of PDF plots (can also be executed server-side |
| $(PQLXBIN)/listPSDChannels.sh | bash shell script – return all channels qualifying for PSD analysis |
| $(PQLXBIN)/exFREQS | bash shell script – return list of PSD frequencies for a given channel |
| $(PQLXBIN)/exPDFfreq | bash shell script – return a PDF for a given date range and frequency range |
| $(PQLXBIN)/exPDFhour | bash shell script – return a PDF for a given channel and time and month range |
| $(PQLXBIN)/exPSDhour | bash shell script – return PSDs for a given channel and date and time range |
| $(PQLXBIN)/exPDFstat | bash shell script – return requested statistics for specific System PDF and channel |
| $(PQLXBIN)/zeroFillPDF.sh | Bash shell script – zero fill period/dB bins (output from exPDF* scripts outputs only bins with hits) where no hit is defined. Useful when extracting PDF data for onward use in other analysis programs, e.g., MatLab. |

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

# PQLX - Getting Started

## Introduction

Once the PQLX analysis system has been installed and initialised, getting the system up and running is a simple matter of executing the following steps:

1. **Compile** and **Install** the PQLX system following the instructions found in: '**PQLX-0_Compilation_&_Installation.pdf**'

2. **Organise** the trace and response file data for all seismic channels to be contained in a PQLX database:

   Documentation detailing this can be found in:
   '**PQLX-2_DB_Setup_&_Prep.pdf**'

3. **Create** a PQLX database:

   Documentation detailing all aspects of PQLX Database Administration can be found in:
   '**PQLX-3_DB_Adminstration.pdf**'

4. **Execute** the Server-Side Analysis Program:

   Execution is possible in a number of ways:
   - From within the 'pqlx-admin' tool
   - From the command line
   - As a cron job

   Documentation detailing the PQLX Server-Side Analysis Program can be found in:
   '**PQLX-4 Server-Side Programs.pdf**'

5. **View** the analysis results using the PQLX Client-Side Program 'pqlx':

   Documentation detailing the PQLX Client-Side Program can be found in:
   '**PQLX-5 Client-Side Programs.pdf**'

## Documentation Set

The complete list of documentation describing the various aspects of the PQLX System is:

| **Name** | **Description** |
|---|---|
| PQLX-0_Compilation_&_Installation.pdf | All information regarding compiling and installing the PQLX system. |
| PQLX-1_Getting_Started.pdf | This document |
| PQLX-2_DB_Setup_&_Prep.pdf | Describes how to prepare and organize trace and response file data for PQLX analysis. |
| PQLX-3_DB_Administration.pdf | Describes the PQLX DB Administration Program 'pqlx-admin'; how to create and manage a PQLX database and execute the |

| | |
|---|---|
| | Server-Side Analysis Program. |
| PQLX-4_Server-Side_Programs.pdf | Describes the main PQLX Analysis Program as well as other Server-Side Programs that are available. |
| PQLX-5_Client-Side_Programs.pdf | Describes the main PQLX Data Viewer Program as well as other Client-Side Programs that are available. |
| PQLX-6_Data_Extract_API.pdf | Describes the existing API's allowing shell- and script-based extraction of PSD and PDF data.<br>This allows for data to be extracted for import to other analysis programs, e.g., Matlab. |
| PQLX-7_XML_Input.pdf | Describes how to use XML files to import Channel Meta-Data and Seismic Event Information |
| PQLX-8_Travel_Time_Tables.pdf | Describes how to generate and import travel-time tables for use within PQLX |
| PQLX-Bug_Reporting.pdf | Describes how to submit a bug, enhancement request, comment, etc. |
| PQLX-Support_Services.pdf | Describes how to arrange for PQLX support services when desired |

# PQLX Database Setup & Preparation

## Introduction

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

- Define which channels are to be contained in a single PQLX Database
- Define the location of all data to be analyzed per PQLX Database

There is no limit to the number of PQLX databases that may be created, hence, consider the most logical way that data should be segregated. For example, one database for a real-time network, another for new instrument testing, another for archived data, etc.

## Data Preparation

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

Decide for each PQLX database those channels (or stations) whose data should be grouped together in a single database. Subsequent 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 thereof 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 and/or sub-directories, how this is managed is completely left to the implementor.

However, experience suggests that ultimately the most user-friendly configuration is one where individual directories are maintained for response files and trace data files. Both of these directories being separate directories to be scanned by the PQLX server process. In this manner, it is possible to maintain a single directory of response files that may be used across multiple PQLX database instances.

## 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). (N.B. 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.)

Response file requirements are:

1. Each response file must contain information for only a single channel.
2. Each response file must contain response information over the entire lifetime of the instrument. 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 a likely culprit. Check the errors on the EVENTQ (via helper scripts listQ.sh, pqlx-ErrorDetail.sh, and pqlx-ErrorSummary.sh), as well as the server error logs. As well, confirm that the response file adheres to the format described here and start again.

## Data Set Preparation

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

1. Decide how data should be segregated by database, i.e., decide which channels should be contained within a single database instance.
2. 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.

## MySQL/PQLX Environment Initialization

If the PQLX environment was automatically initialized during installation, this section may be skipped. (You can check if the system has already been initialized by executing the init script below. When the system has already been initialized, the script will refuse to execute.)

If system initialization was deferred, no aspect of PQLX may be executed without this initialization.

The PQLX system is initialized with the following command, (located in $PQLXBIN):

```
bash> 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 request the MYSQL user 'root' password, necessary for PQLX system initialization. This is the only time that the MYSQL root password is required. It is not saved or ever used again.

## PQLX Database Creation

Use the PQLX client program 'pqlx-admin' to create a PQLX database; see the document '**PQLX-3 DB Administration.pdf**' for more details.

Alternatively, it is possible to create a PQLX database manually outside 'pqlx-admin'. Previously, before the admin tool existed, PQLX DB management was based on the existence of a PQLX DB definition file, used in conjunction with various PQLX scripts.

The following describes this process for those cases when the Administration tool is not possible. However, the DB definition directory is now marked read-only for non-root users. Thus, creation of the database definition file must be done with the 'sudo' command.

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

1. Define a database definition file (located in directory $PQLXDBDEF) for each PQLX/PDF database instance desired. The name of this file must adhere to the following file naming convention:
   ***dbName***.pqlx
   where
   ***dbName*** is the name of the PDF Server Database, and
   .pqlx is as is
   (See an example of this file contained in **$PQLXDBDEF/samplePQLXdb.pqlx**.
   See file **$PQLXDBDEF/PQLXdbTemplate.pqlx** for a template of this file.)

   Edit the DB 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 (i.e., permanent) or SCRATCH (i.e., 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)
   - Define which PDF statistics should be displayed when creating PDF PNG plots.
   - Define the date format to use when creating PDF PNG plots.

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

2. 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).

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.  Settings having an effect on server output will take effect the next time the server is executed.

**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, and **enabling** and **disabling** the reading of a data directory.  Execute without arguments for complete details.  N.B.  This can only be executed on the server machine itself.

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

# PQLX DB Administration

## Introduction

This document describes the PQLX GUI program 'pqlx-admin'. 'pqlx-admin' provides all functionality related to PQLX database management, including database creation, modification, and deletion; reading of server execution log files; viewing various database statistics (by channel, by data directory), etc. Where 'pqlx' is used to view trace files and their analysis results, 'pqlx-admin' is used to display the various information related to a database itself.

## Execution

'pqlx-admin' may be executed in one of two ways:

1. From a bash shell command line using the command 'pqlx-admin', usage:

   ```
   bash> pqlx-admin [--dbName=NameOfDB]
   ```

   specifying ***NameOfDB*** as the argument to the optional parameter --dbName will automatically connect to the named PQLX database at startup.

2. From with the 'pqlx' GUI program by clicking on the 'Admin' button found on the Main Tab side-bar of the PDF and STN System Viewers. When invoked via 'pqlx', if 'pqlx' is, itself, connected to a PQLX database, the 'pqlx-admin' tool will automatically be connected to the same database.

## GUI Layout

The GUI is laid out much like 'pqlx': A side-bar containing buttons as well as selectable choices; with a notebook filling the rest of the application window, containing tabs to display the various definition, configuration and statistics data of a PQLX database.

The application will be in only one of the following display states at any given time:

- Not connected to any PQLX database
- Creating of a new PQLX database
- Connected to a PQLX database in read-only mode
- Connected to a PQLX database in edit mode

Each display mode will result in a specific combination of display items (buttons, tabs, fields, etc.) being visible and/or available for use.

When invoked without arguments, the program is initially displayed not connected to a database. From here, the user has a choice of:

1. Creating a new PQLX database by clicking on the 'NEW' button on the side-bar, or
2. Connecting to an existing PQLX database by clicking on the 'Manage' button on the side-bar.

## Creating a NEW PQLX Database

When creating a new PQLX database, the user is required to define all the various aspects of a PQLX database found in the sub-notebook of the Definition tab; namely, its description, the directories holding the data files to be analyzed, the PSD Channel Configuration, and the System PDF PNG output settings.

At any time in the database creation process, clicking the 'Cancel' button at the bottom will discard all information and return the program to the non-connected state.

### Database Description

As the name implies, the Description tab supplies the various values that describe the database:

| Field Name | Description & Function |
|---|---|
| DB Name | The name of the PQLX database, must be unique on this database host machine |
| Hostname | The name or IP address of the machine to host the DB. May only be selected using the '**...**' button to the right of the entry area. |
| Description | A general description of the DB |
| Organization | The name of the organization supporting this DB |
| Type | PRODUCTION or SCRATCH - Used only to decide how easy or difficult the DB can be deleted |
| Access | PUBLIC or PRIVATE, where:<br>• PUBLIC allows all users to see and connect to the DB, and<br>• PRIVATE allows only the creating user to see and connect to the DB |
| Data Admin | The name and email address of the person responsible for the administration of the seismic data represented by this DB |
| Technical Admin | The name and email address of the person responsible for the administration of the technical aspects of the DB. |

Use the 'Next' button (bottom right) to be taken to the next screen:

### Data Directories

The Data Directories tab defines the data directories that will be analyzed as part of the execution of the PQLX server program. When adding a directory, a file/directory dialog will appear allowing the user to select which directories should be analyzed. Directories are enabled by default, but may be, at any time, disabled from being scanned using the 'Disable' button on the right side-bar. Deleting a directory is only possible in DB Creation mode, once a database has been created, a directory can only be disabled, permanently if necessary, and never deleted.

A word on PQLX Data Directory set-up: the recommendation is that response files and trace files be separated into two distinct directory structures. And that when creating a database, the directory containing the response files be specified first. This way, the first time the server executes, the response files are identified first. For large datasets, when response files are identified only after thousands of trace files have been identified, problems may (or may not) occur; this can be avoided by specifying the directory containing response files first.

**PSD Config Settings**

Various PSD configuration parameters may be set once DB creation has taken place. Including, defining which channel groups (e.g., BH*) should be analyzed, the length of the PSD window (in seconds), the lower and upper dB bounds, and the maximum frequency to resolve to.

When not wanting a specific channel group to have PSD's computed, remove all corresponding records from the PSDCFG table corresponding to the channel group in question (*PSDCFG.channel*).

**PNG Settings**

The PNG Settings tab defines how the System PDF's output by the PQLX server program will look, these being generated as the last step of the PQLX server's execution. These PNG files may be used as external and static PDF plots to be displayed outside of the PQLX environment. For example, these PNG's can be located in a directory that is pointed to by a web page (internal or external to the organization), allowing for easy and automatic publishing of PQLX results.

The various settings and their descriptions are:

| **Field Name** | **Description & Function** |
|---|---|
| Directory | Clicking the '**...**' button will pop up a file/directory dialog allowing the user to select the output directory where the System PDF PNG's will be written to. Leave this field blank to have the server generate no PDF PNG files. |
| Date Format | Specifies the date format that will appear on the System PDF's |
| Plot HLNM | Check button indicating if the Peterson High/Low Noise Model should be plotted on the PDF |
| Normal PNG | The normal PNG is the System PDF displaying all information, in a normal size |
| Icon PNG | The icon PNG is a smaller version of the normal PNG displaying less information, but enough to get an idea of the PDF plot. This may be used as a pop-up, for example, on a web page listing all channels. |
| Width/Height | Clicking the '**...**' button will pop up a sample PDF plot. By dragging the lower right corner, the user can visually determine the exact width and height of the PDF PNG output. |
| Border & Color | Indicates if the PDF plot should display a border. The color picker allows the user to define the color to be used. |
| Statistics | Via checkboxes, the user can define which statistics should be displayed on the PDF plot. |
| TEST | The TEST button displays a sample PDF plot using the settings provided by the user. |

**Password**

Provide the password that will be associated with a database. This password will be needed in the future when wanting to edit any of the database configuration parameters.

Provide a hint here to remind yourself of the password when it has been forgotten. (This hint will be displayed exactly as input when requested.) Either the hint should be private (only you understand the hint to resolve the password), or, if the database should be completely open to everybody for editing, set the hint phrase to explicitly state what the password is. The hint may be blank, but this is discouraged.

Once the password has been provided and confirmed, click the 'Create DB' button to create the PQLX database.

*Access to DB Edit functions not yet currently password-protected.*

*(Documentation continued below...)*

## Connecting To a PQLX Database

Use the Manage button on the side-bar to select a PQLX database to connect to. Like 'pqlx', it is possible to connect to any PQLX database on any server the program is able to make a connection to. If the database server is on another machine, i.e., is not 'localhost', provide the server name or IP address in the 'New PDF Server' entry area. Selecting and double-clicking on a PQLX Server will display all PQLX Databases being hosted on that machine.

Once connected, the database information will be displayed in read-only mode; no database configuration attributes are available for editing. However, additional information will be available in the 'Data Directories' tab, namely, for each directory, the date of the most recent file identified, the total number of trace files, and the total number of response files will be displayed.

Use the 'DISCONNECT' button to disconnect from the currently connected database, returning the program to a non-connected state.

In addition to the database Definition tabs, additional tabs 'Channels' and 'Exec/Logs' are available for viewing.

### Channels Tab

The 'Channels' tab lists all 'active' channels existing in the database. An 'active' channel is defined as a channel that has at least one trace file associated with it. ('Inactive' channels can exist in the database, for example, when response files have been identified for a specific channel while no trace files exist. This is interesting since it makes it entirely possible, for example, to specify the same single response file directory for multiple PQLX databases that contains ALL response files for all PQLX databases, while at the same time the data directory(-ies) for any specific PQLX database contains trace files for only a sub-set of the channels defined by the response file directory.)

Top of the tab displays the following information:

- Search: - type in any part of a Channel Name to restrict the display list, list is automatically updated.
- Total Channels - Defines the total number of channels currently on display
- Start Date - List the Start Date and Time of the channels currently on display
- End Date - List the End Date and Time of the channels currently on display

The list of channels provides for the following statistics to be displayed:

- Start and End Dates of Identified Trace Files
- Total number of Traces
- Total number of Computed PSD's
- Total number of Sample Rate Epochs
- The full-path name of the associated response file

Double-click on a channel row to view the Sample Rate Epoch date boundaries for a given channel. In the case where multiple sample rate epochs exist for a channel, System PDF's will be created only for the most recent sample rate epoch.

### re-PSD

The re-PSD button on the side bar provides the option to have one or more trace file re-analyzed for PSD computation. Select one or more channels for re-analysis, click the re-PSD button, and select/specify the date range for which the trace files will be re-analysed.

Once all channels and date ranges have been selected, either:

- go to Exec/Logs tab to execute the server, or
- wait until next (possible) automatic execution of server

### Exec/Logs Tab

The 'Exec/Logs' tab lists all executions of the PQLX server-side programs (currently only 'pqlxSrvr'). Selecting a specific execution in the upper-most data viewer will display the log output of that execution in the lower data viewer. If the server program is currently executing, the log viewer will automatically be updated as new messages are generated by the server. Use the 'LOG Display' pull-down to define if LOG messages, ERROR messages or BOTH should be displayed in the log viewer. As well, when the server is currently executing, an additional window will be provided to display the current event queue contents, allowing the user to follow a server's execution as it proceeds.

Using the 'Refresh' button the side-bar will update the list of server executions, guaranteeing that the list on display is up-to-date and current.

### Executing the Server Analysis Program

Use the 'Execute' button on the side-bar to execute the server from within the 'pqlx-admin' tool. It is still, and always will be, possible to execute the server from the command line as before.

Once an execution has started, the Server Execution and Log windows will automatically update. The 'Execute' button will change to 'Pause', allowing the user to suspend the server's execution. When a pause request is made, the server must first finish processing of any active events currently executing; scanning of data directories is not pausable. Use the 'Resume' button to continue the server's execution.

During execution, an "Event Q" display will appear describing the current state of the PQLX Event Queue, all PQLX events falling into one of three categories:

| **Type** | **Description** |
|---|---|
| **Executing** | Event is currently being executed |
| **Queued** | Event is queued awaiting execution |
| **Pending** | Event is sitting on the queue, but execution is pending until relevant execution phase of the program is reached. |

Once execution has completed, analysis results may be viewed with the Client-Side Program 'pqlx'. Selecting the '**pqlx**' button on the side-bar will invoke the Client-Side Display Program, automatically connected to the same PQLX database 'pqlx-admin' is currently connected to.

### Server Configuration

Use the 'Configure' button on the side-bar to view various server execution options, specified in the following sections:

- General Options:
    - Number of CPU's - How many CPU's should the server use to analyze the data

    - Directory Scanning - Do not scan **any** data directories
    - Refresh - Create events to have all System PDF's and Spectrograms re-computed and output

- Process Specific Files:
    - File Name - Specify/Select the input file containing the list of trace files that should be re-analyzed.
      N.B. When specifying this option, Directory Scanning is disabled by default.

- Time Restriced Execution:
    - Time Restriction - Allows the user to specify that the server should execute only during specific hours during the day. When 'On' is selected, additional arguments define exactly during which hours the server is allowed to execute.

      When using this argument, the server will enter into a Pause state at the point that the End Time is breached. When the Start Time once again comes around (on the clock), the server will issue a 'Resume' command and execution will again resume until End Time is again breached.

*(Documentation continued below...)*

## Editing a PQLX Database

After having connected to an existing PQLX database, it is possible to edit its definition and configuration parameters. For example, a user can enable or disable Data Directory scanning, modify the Description definitions, modify the PNG Output Settings, or even delete the database completely.

Use the 'EDIT' button at the bottom to start edit mode, *providing the proper password to gain edit access*. Once in edit mode, modify all fields as desired. When complete, use the 'SAVE' button at screen bottom to update the database configuration parameters. All settings take immediate effect (though may not actually do anything until the server is next executed).

N.B. The Database Name and Database Host cannot be changed once a database has been created!

# PQLX Server Analysis Program

## Introduction

Once a PQLX database has been instantiated, 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 data directories defined as part of the DB definition, 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 PSD and PDF display.
  - Additional global statistics for each channel over all time are also maintained.

Execution may be carried out in one of several ways:

- via client-side program 'pqlx-admin', see document 'PQLX-3 DB Administration.pdf' for complete details
- via direct execution from the command line:

```
bash> $PQLXBIN/pqlxSrvr --dbName=[srvrHost:]PQLXdb-name --numCPU=numCPUs
```

where:

- ***[srvrHost:]*** is the name of the server machine hosting the PQLX database, i.e., where the MySQL database server is running. No specification defaults to ***localhost,*** i.e., 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.

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. As well, the list of trace files causing the crash may be used for any further script processing, e.g., 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 re-processed 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 general order of analysis operations is:

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

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

3. 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, etc.) may require days to complete. In the meantime, new data files may have come to exist which now require processing. This guarantees, then, that once the server completes the database 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 itself on the same day. That is, they will not be “re-identified” as part of this step, thus avoiding a rather inconvenient infinite loop.

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

5. Process any necessary deletions of PSD data. PSD data is required to be deleted when a trace file has been replaced or re-identified during the file identification phase.

6. Update channel and directory statistics held on the database.

7. 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*, 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.

8. If PNG output of the System PDF's is specified (see PQLX DB Administration document for details) , all updated system PDF's are output to the defined directory as PNG plots.

## pqlxSrvr_safe & 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).

A sample shell script that can be used as a template for a cron job is provided in $PQLXBIN/pqlxCron.sh. Edit this file and copy to a separate location (subsequent upgrades to the system will overwrite any previously edited version) for execution by cron.

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 a daily update of a PQLX database, it is recommended that the server be executed locally sometime after 00:00 GMT. In order to minimally guarantee that all trace files from yesterday are fully processed.

## 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 which has already been analyzed (or, said otherwise: the PSD data is 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
--noScan                  Do Not Scan Data Directories – optional
--TRStart=HR:MN           Time-Restricted Execution – Start Time – optional
--TREND=HR:MN             Time-Restricted Execution – End Time – optional
```

Providing a file of traces via the **–identFile** argument will force the server to re-analyze 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. N.B. No scanning of the data directories occurs when executing using the --identFile option. Once the re-analysis is complete, all channel statistics are updated and system PDF's recomputed.

## Log file Output

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

1. **srvr.*dbName.YYYY.JJJ.HH:MM.*log** – standard log file providing information related to the normal course of execution.
2. **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.
3. **srvr.*dbName.YYYY.JJJ.HH:MM.*crash** – file containing, one per line, the full pathnames of all trace files which 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.

## EVENTQ Monitoring and Errors

In order to monitor the state of the server, as well as determine the nature of errors encountered during normal processing, a helper script **listQ.sh** is provided and is described here.

**listQ.sh** – Outputs EVENTQ table information currently held by the database. The EVENTQ table contains all data processing events processed by the server. Examination of this table is very helpful, if not crucial, in ascertaining the origins of various problems that can arise. This script outputs information of two sorts: current status of the EVENTQ table itself (useful when wanting to know where the server is in the course of its processing), and listing of the errors encountered by the server.

Option 1: **listQ.sh *dbName* ALL** - will output all events currently existing on the EVENTQ.

Possible events include:

- Trace file meta-data analysis,
- Trace file PSD Analysis
- Overlapping Trace file PSD Analysis
- PSD Data Table – index creation
- PSD Data Table – delete PSD's
- System PDF – creation and statistics
- PDF PNG – creation of PNG files

Possible STATUS values:

- INPROC – event is currently being processed by the server
- UNPROC – event is yet unprocessed, waiting for server execution, currently queued
- NOPROC – event is yet unprocessed, waiting for server execution, not yet queued
- PROCD – event is complete and has been processed by the server. If this STATUS is listed, then by definition there was an error processing this event since all events successfully processed are immediately deleted from the EVENTQ table.

This script is helpful during execution of the server and when wanting to know which events are currently being processed, how many to go before completion, what kinds of errors are being produced, etc.

Option 2: **listQ.sh *dbName* ERRORS DETAIL** – will write a **detailed list** of all errors currently existing on the EVENTQ table to output file in $PQLXLOG directory.

Errors include, but are not limited to: missing response files, trace files that do not overlap, trace files not containing enough data for processing, etc.

Option 3: **listQ.sh *dbName* ERRORS SUMMARY** – will write a summary of all errors currently existing on the EVENTQ table; output written to $PQLXLOG directory.

## Other Server-Side Programs

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

1. **pqlxPNG –** generate PNG format plots of PDF's.
2. **pqlxDBMaint** – a database maintenance program
3. **rePSD** – re-execute PSD computations for a particular channel and time range

## pqlxPNG

This program generates PNG plots of PDF's. Execution may be either automatic (e.g., 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 PDF's 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 PDF's, command-line usage is the following:

```
bash> $(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 (e.g., --dbName=SrvrName:MYPQLX).

Optional Arguments:

**--icon**
If specified, will additionally output an icon version of the System PDF plot.

**--pngDir=**
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. If this is case, or when wanting to override the database-defined directory, specifying a directory with this option will output the PNG plots here.

### Option 2

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

```
bash> $(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 himself. 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
--icon
```

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 thumb-nail 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.

--icon option will additionally create an icon version of either the System PDF's or the input data.

## pqlxDBMaint

Previously, execution of pqlxDBMaint was required to be managed by the PQLX DB administrator. This has been reworked whereby the server will automatically execute the pqlxDBMaint program according to the following rules:

- If server has executed at least eight (8) times since the last time the pqlxDBMaint program was executed
- If server is executing on a Saturday
- Whichever happens first

### Reasoning

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, as well as possibly be moved to a “crashed” state by the MYSQL database server.

As well, all events with errors existing on the EVENTQ are deleted, with one exception. Error “No Response File” are never deleted, thus retained forever. This allows for the case when new channels of data arrive perhaps before the meta-data describing the instrument is known or available. Once the response file is made available to the system and identified, all events previously marked with this error are reset to status 'UNPROC', essentially returning them to the EVENTQ for processing.

## rePSD

One method of forcing a re-computation 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 re-analyze 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
      re-initiating the server.
    Not specifying numCPUs will delete the PSDs and recreate pqlxSrvr events
    for subsequent processing, but WILL NOT re-initiate the server.
    This is useful when wanting to delete several ranges of PSDs
    (in time or by channel) before re-initiating the server.
```

examples:

```
bash> $(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 re-computation, but DO NOT execute the server

```
bash> $(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 re-initiate the server to process all outstanding PQLX events, to execute on 4 CPUs

## PQLX Client Execution

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

```
bash> $PQLXBIN/pqlx
```

Once started, different databases (and/or servers) may be connected to. Please see the document '**PQLX-5 Data Display Program.pdf**' for a quick overview of the client program's functionality and usage.

# PQLX Client-Side 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:

1. pqlx – GUI program to visualize database data
2. 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 time- and frequency-domain data stored in the PQLX database. Time-domain data (i.e., trace files) may be either files that have been analyzed by the server, or trace files on disk that have not been analyzed (via Trace Viewer).

### Startup

To execute pqlx:

```
bash> $PQLXBIN/pqlx
```

will 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 or the keys F1, F2, or F3, three data display systems are available:

- **Trace Viewer** - view trace data in detail (this does not utilize a connection to a PQLX database)
- **PSD Viewer** - view PDF and Spectrogram plots of PSD data stored in a PQLX database
- **STN Viewer** - view trace data and its availability by station and channel that has been analyzed by the PQLX server.

### Trace Viewer

A broad overview of the functionality provided by the Trace Viewer:

- **Trace Tab** - view trace data, originating from:
  - individual trace files,
  - data passed from the PDF viewer, or
  - data passed from the STN viewer
- **Magnify Tab** - Zoom In and Out on sub-selected data segments
- **Spectra Tab** - display and transform the fourier way on sub-selected data segments
- **Split Tab** - display multiple data views simultaneously
- **Header Tab** - display all trace file header information

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

### PSD Viewer

Upon initial execution, **pqlx** is not connected to any specific database. In order to view PDF and Spectrogram plots, a connection to a database must exist. This is achieved via the Servers button available on the sidebar of the Main tab. The first time a server (not the localhost) is connected to, 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.

(N.B. Connection requirements for **pqlx** and the server database: MYSQL communicates over port 3306. If you can ping the server machine 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).

In the Controls Panel, it is possible to specify that a server and database should automatically be connected to at program startup. This is useful if a single database is connected to more than others.

The PSD Viewer is broken up into two notebooks, PDF and SPECTRO, where PDF displays various PDF types, and SPECTRO displays optional PDF's as well as spectrograms of PSD data.

## PDF Notebook

The PDF notebook is further broken down into various sub-notebooks, these being:

### Main Tab

Display system PDFs computed by the pqlxSrvr program.

Display By options:

- **Group -** Display a selected system PDF for the selected stations and channel group
- **Station** - Display three user-selected system PDFs for a particular station and channel group
- **PDF** - Display a selected system PDF for three individual stations and 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 <u>available</u> PSDs (previous 30 days)
- **Week** - PDF of last week of <u>available</u> PSDs
- **L Month** - PDF of previous month of <u>available</u> PSDs (previous 60 - 30 days)
- **L Week** - PDF of previous month of <u>available</u> 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 Colour Bar

- **CLICK** - on PDF Plot:
  - take PDF to Detail Tab

**Detail Tab & Panes**

- **NW Pane** - Main PDF of Detail Tab
- **SW Pane** - PDF of sub-select of Main PDF. This PDF is made up of all PSDs intersecting the point or region defined by the NW Detail Pane sub-select
- **NE Pane** - displays the first 15 traces representing the PSDs of the NW pane sub-select
- **SE Pane** - displays the start time and duration of all PSDs returned by the NW pane sub-select

SideBar Options:

- Back and Forward Buttons:
  When making sub-selects of a PDF, a list of PDF's is maintained allowing the user to cycle backwards or forward through the list.
- Trace Plot:
  User options defining how trace data is displayed in the NE pane:
  - Do Not Plot –No trace data will be retrieved or displayed
  - Entire Trace – Display the entire trace containing the computed PSD
  - PSD Portion – Display only that part of the trace corresponding exactly to the computed PSD
- Hour Sub-Select:
  User options defining how PSD's are sub-selected in the SE pane:
  - Inclusive – Select all PSD's falling inside the bounding box
  - Exclusive – Select all PSD's falling outside the bounding box
- Save Data:
  Allows the user to save the various types of data on display in the Detail Panes:
  - PDF – PDF data (frequency, dB, number of hits, probability) of the NW pane (or SW pane when sub-select PDF exists) is saved to file specified by user.
  - PSD – PSD start times are saved to file specified by user.
  - Stats – Statistics (frequency, dB) for the selected statistic saved to file specified by user.

Mouse Functions:

- **NW Pane** - Click:
  - define sub-select point
- **NW Pane** - Click+Drag
  - define sub-select region
- **SW Pane** - Click:
  - make sub-select PDF the new Detail Main PDF, i.e., move to NW Pane
- **NE Pane** - Click:
  - Take all traces of sub-select to PQL
- **SE Pane** - Click+Drag:
  - Select PSD's by hour and date to display a new PDF in the NW Pane

### Request Tab & 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 a 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

### Movie Tab & Panes

The Movie Tab allows the user to make animations of PDF's over a series of time.  Display of 1 or 4 simulaneous panes is possible.  Using the Make Movie side-bar buttons, select the channel and time range desired as well as the number of days which should be contained in each movie frame.  As well, define if a movie should be made up of cumulative PDF's or individual PDF's per movie frame.

## SPECTRO Notebook

The SPECTRO notebook is further broken down into various sub-notebooks, these being:

### Main Tab

Display system PDFs (optional) and corresponding spectrogram computed by the pqlxSrvr program.  Like the Main Tab of the PDF notebook, via the side-bar, a specific System PDF may be selected, the specific set of stations selectable via the 'STATIONS' button, and/or a Channel Filter may be selected to restrict the list of channels to display.

**Next** and **Previous** buttons advance the display list accordingly.

**Period Bounds** allows the user to specify the exact period boundaries that should be used when display specific channel groups.

**PDF Display** indicates whether the corresponding PDF should be displayed or not.

Mouse functions include:  (see 'mouse tips' at side-bar bottom for more options)

- click on PDF - take PDF to PDF::Detail screen
- click on Spectrogram - take Spectrogram to SPECTRO::Detail screen

### Detail Tab

Taking a spectrogram to the Detail tab allows for further analysis and investigation of the spectrogram, allowing, for example, zooming in and out on the spectrogram. Zooming can be done in multiple ways:

- mouse-click-down at zoom start + drag + mouse-up
- mouse-wheel scroll up - zoom in
- mouse-wheel scroll down - zoom out

## STN Viewer

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

### Main Tab

The STN (station) viewer allows for display of data coverage or data itself 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, etc.

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 themselves. This is good for 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 itself.
- **Data** - In Data mode, actual data files are read from disk and displayed, much like as in PQL. Unlike in PQL, however, all displayed data is internally held as a single trace. Important when, for example, applying a filter; no effects of data discontinuity (e.g., 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

  N.B. 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**, **CTRL**, **a-**, and **s-** keys above:
        - Select entire **station** on display
- On Channel Label:
    - As for **SHIFT**, **CTRL**, **a-**, and **s-** keys above:
        - Select entire **channel** on display
- Between Day Labels (X-Axis region):
    - As for **SHIFT**, **CTRL**, **a-**, and **s-** keys above:
        - Select entire **day** of data on display

**Analysis Tab**

Bringing data to the Analysis Tab from the Main Tab of the STN viewer allows the user to perform various analytical transformations on the data:

- Demean
- Differentiate
- Integrate
- Reverse Polarity
- De-Trend
- Spectra - Compute spectra of data on display, provided in a separate pop-up. See Controls for more options.
- PSD - Compute total power spectral density of data on display, provided in a separate pop-up. Access to the response files as defined in the database is required. See Controls for more options.
- UVW - Convert XYZ data coordinates back to UVW coordinates. Valid for STS-2 and Trillium instruments only. (Configuration setting for this can be found in Controls panel of the Trace Viewer Display.)
- Deconvolution - Remove the instrument response from the trace data (time domain)
- Particle Motion - Display two-component (horizontals) particle motion plot

All other side-bar controls are the same as found elsewhere in pqlx and pql.

## pqlxPNG

The 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 using the GUI client **pqlx**:

1. those users who do not have the GUI client pqlx installed on their machine
2. those users not having access to the database itself, for example, external users that are not permissioned to access the LAN on which the server resides.

As well, in providing PNG format output, it is very trivial to create graphical images which can be used for publication.

### 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
  --icon             Additionally create icon versions – optional
  --systemPDF        Make PNG plots of System PDF's...
  --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 are possible to make with pqlxPNG:

1. All System PDF plots for a given database
2. 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. Either 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:

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

```
bash> 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, specified as seconds (not Hz)
- **power** – the power
- **number of hits** – the number of occurances 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.

# PQLX PDF & 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:

- **exFREQS –** returns a list of frequencies held in the database for a given channel
- **exPDFhour** – extract the PDF for the given channel and bounding parameters: month range, 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, start and stop times
- **exPDFstat –** extract the requested statistics from the database for a given channel and System PDF.

## exFREQS

Returns a list of frequencies held in the database for a given channel.  Output written to stdout.

Usage:

```
exFREQS [HOST:]DBName NTW STN LOC CHN
```

## 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:

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

```
bash> 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:

```
bash> 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]
```

Example:

```
bash> 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:

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

returns the PSD's for the given channel and bounding parameters
format: DATE HOUR Frequency Power

## exPDFstat

Usage:

```
exPDFstat [HOST:]DBName [PDFLIST | PDF# NTW STN LOC CHN stat1 [stat2 ...]]
```

Example 1:

```
bash>  exPDFstat micros PDFLIST
```

List PDF Number Specification and All Possible Statistics

Example 2:

```
bash>  exPDFstat micros 1 IU ANMO -- BHE 10 90 mode
```

Output PDF statistics of PDF #1 for 10th and 90th Percentiles and Mode

Output file is named NTW.STN.LOC.CHN.***PDF#***.***statname***.stat
where:

***PDF#*** – is the PDF Number specified on the command line
***statname*** – *is* the Statistic Identifier

and is written to Current Working Directory

# PQLX XML Input

## Introduction

PQLX provides the possibility to import data not specifically trace-related. Specifically, using XML format files, the following information can be imported to a PQLX Database:

- Channel Meta-Data Information
- Seismic Event Information

## General Description

Using XML-format input files, it is possible to import additional information for use within the client GUI program, pqlx. The manner in which this import occurs is the following:

- XML-format input file provided and placed into directory $PQLXXML
- Data is imported during next execution of server program, pqlxSrvr
- Once data file has been imported, file is moved to directory $PQLXXML/hist with a date/time stamp added to the filename.

Since this import is automatic upon execution of the server program, it is quite easy then to create a separate process outside PQLX that is responsible for generating these files. For example, channel meta-data information may be extracted from another database and written to the PQLX XML input file ready for subsequent automatic import.

## XML File Name Format

The XML file name format must adhere to the following naming conventions:

*dbName.importType*.xml

where

*dbName* is the name of the PQLX database to import to

*importType* is one of the following values:

- **chnMETA** - representing channel meta-data information
- **eventInfo** - representing seismic event information

## Channel Meta-Data

The following channel-specific meta-data information may be imported:

- Start and End Dates information represents
- Site Name
- Position: Latitude, Longitude, and Depth
- Sensitivity
- Units
- Instrument ID
- Azimuth
- Dip
- Sample Rate

Since all information is defined to be valid for a specific date range, it is possible to input all information for a channel's entire life. If on import a record is found to have been previously imported, it is replaced. Thus, it is possible to re-import information guaranteeing that all previous records are deleted and replaced.

A sample input file can be found in directory $PQLXXML, named **dbExample.chnMETA.xml**.

## Seismic Event Information

The following event-related information may be imported:

- Description: Name and ID
- Origin ID
- Providing Catalog
- Location Information: Time, Latitude, Longitude, and Depth
- Magnitude and Type

If on import a record has been previously imported, it is replaced. Thus, it is possible to re-import information guaranteeing that all previous records are updated as expected.

A sample input file can be found in directory $PQLXXML, named **dbExample.eventInfo.xml**.

## Post-Import Usage

Once the information has been imported, it may be used in the following ways within the client program 'pqlx':

1 Channel Meta-Data Information:
  - 1.1 Within STN Viewer Main Tab - on mouse-hover over channel label, display as pop-up the channel-specific meta-data information corresponding to the specific date and time of data on display.
  - 1.2 Within STN Viewer Analysis Tab - Station location information used when computing predicted arrival times vs. a selected seismic event.
  - 1.3 Within STN Viewer Analysis Tab - Channel azimuth and sensitivity information from the XML files (not the response file) is used for the necessary computations in generating the Particle Motion plot.

2 Seismic Event Information:
  - 2.1 Within STN Viewer Analysis Tab - Imported events are selectable via the side-bar button 'Events'.
  - 2.2 Selected event used to compute predicted arrival times vs. channel data on display. Channel Meta-Data information is also needed for this.
  - 2.3 Selected event used to compute Event Back Azimuth for computation of Particle Motion plot. Channel Meta-Data information is also needed for this.

# PQLX Travel-Time Tables

## Introduction

PQLX provides the possibility to import various travel-time tables to be used as part of the Seismic Event functionality available within the STN Viewer. Specifically, this provides the ability to plot predicted arrival times for many phases vs. a selected event and stations of data on display.

The current travel-times tables delivered with PQLX are **iasp91** and **ak135**. (Please see Buland and Kennett for details.)

The overall process involves a series of steps, detailed below, and may be generally described as:

1. Compile the source code responsible for generating the travel-time tables.
2. Generate the travel-time table(s) using the program compiled in step 1.
3. Using PQLX program **importTT**, import to PQLX the tables generated in step 2.

## Step 1: Compilation

Compiling the '**ttimes**' program to generate the tables is relatively straight-forward:

1. **ttimes** source code must be compiled on the PQLX server machine
2. When executing **env/makeAll** to compile the system, additionally include the parameter ***withTTimes*** on the command line to force compilation of the **ttimes** program:

   ```
   bash> env/makeAll withTTimes
   ```

   N.B. This requires a Fortran compiler in order to succeed. The Fortran compiler used defaults to **g77**. If a different Fortran compiler is desired or needed, **gfortran** for example, it may be specified by setting the environment variable **FCMP** which will override the default setting:

   ```
   bash> export FCMP=gfortran
   ```

## Step 2: Generate the Travel-Time Tables

Once the system has been compiled, the travel-time tables need to be generated for importation, according to the following:

1. Execute the shell script **$PQLXBIN/pqlxTTtable.sh**. This script will query the user for which table should be generated, **iasp91** or **ak135**. If both tables are desired for import, execute a second time to generate both.
2. Output is written to directory **$PQLXPROD/XMLINPUT**.

   N.B. - the tables generated are approximately 600Mb in size for each, please make sure enough disk space is available beforehand.

### Step 3.0: Prepare Tables and Phases for Import

Once the tables have been generated, the program **importTT** is used to import the tables into PQLX itself. Usage details:

```
importTT [OPTION...] - Execute importTT

Help Options:
  -?, --help          Show help options

Execution Options:
  --modelName        Travel-Time Model Name - required
  --extractPhases    Extract Phases List - optional
```

### Step 3.1: Define the Specific Phases for Import

If all phases defined in the travel-time tables generated in step 2 are needed for import, step 3.1 may be skipped; please proceed to step 3.2.

If, however, only a sub-set of all defined phases are desired to be imported, specifying the **--extractPhases** argument will *only* generate a file containing **all** phase names existing in the specified travel-time table; that is, all phase names which are available for import. The output file will be named **modelName.phases** and located in directory **$PQLXPROD/XMLINPUT.**

For example, to generate the file of phases existing for travel-time table **ak135**:

```
bash> importTT --modelName=ak135 --extractPhases
```

To specify which phases to import, then, create a file named **modelName.head** containing the phases to import, one phase name per line, and place into directory **$PQLXPROD/XMLINPUT**. (And a simpler way: copy the **modelName.phases** file to **modelName.head**; edit the **modelName.head** file and delete the phases not to be imported.)

### Step 3.2: Import the Travel-Time Tables

To import the travel-time tables and (optionally) specific phases into PQLX, execute **importTT**, not specifying the **--extractPhases** option.

For example, to import travel-time table **ak135**:

```
bash> importTT --modelName=ak135
```

If file **$PQLXPROD/XMLINPUT/modelName.head** exists, it will be read and its contents used to define which phases will ultimately be imported.

N.B. Once tables have been imported into PQLX, tables saved to **$PQLXPROD/XMLINPUT** in step 2 may be deleted, as these files can be quite large.

### Travel-Time Tables Details

Tables created by step 1 are generated to produce travel-times for all phases across all combinations of event depth and receiver distance, defined as:

- Source Locations: Intervals of 5 Kilometers
- Source Depth: Zero to 650 Kilometers
- Receiver Distance: Zero to 180 degrees

Predicted arrivals are interpolated between the Source location and Receiver location using the travel-time table values imported to the database.

## Import of Additional Travel-Time Tables

It is also possible to upload any other specific travel-time table of interest to the user, for example, a local velocity model. Simply follow the format of the files generated by **times** (for example, the **ak135** table is named **$PQLXPROD/XMLINPUT/ak135.tt**), name it appropriately (using your travel-time name) and upload with **importTT** as described above specifying this new model name.

## PQLX Usage

When importing Seismic Event information as part of the server's normal execution (see document PQLX-7_XML_input.pdf for details), the client is now ready to plot predicted arrivals based on a selected Event, a selected travel-time table, and selected phases to plot.