1. Information Product Overview

A JSPC information product is an integrated package of seismic data, software and documentation, which can be immediately useful in research efforts. It is delivered on tape; an installation script facilitates extracting information from the tape, and allows unloading small subsets of the waveform data.

The seismic data includes detailed instrument and site characteristics. In addition, it incorporates a base level of processing, such as waveform deglitching, phase arrival picking and event association. The software programs provide a means of inspecting the data, and the libraries provide convenient interfaces for manipulating the data. The documentation, delivered in postscript format, has at least the following major parts:

❍ *Technical Reference Manual* - This describes and summarizes the data.

❍ *Installation/Tutorial Manual* - The current document introduces the information product concept, describes the overall organization, explains how to install the product, and provides a brief tutorial on the database and the use of the software.

❍ *Schema Reference Manual* - This provides an in depth description of the database schema, in which the parameter information for the data is contained.

❍ *Software Reference Manual* - This is the postscript version of the manual pages for the software, including documentation for both programs and libraries.

By providing an integrated package, the JSPC hopes to eliminate much of the drudgery of wading through large volumes of data to separate the interesting data from the uninteresting data, as well as the waste of time finding, for instance, calibration and response information. Although we have spent a considerable effort preparing each information product, there may well still be problems or omissions. If you discover a problem, please contact us:

IRIS Joint Seismic Program Center
University of Colorado
Physics Department, Campus Box 583
Boulder, CO 80309-0583
303/492-5243
problems@jspc.colorado.edu

We plan to release new versions or corrections whenever necessary. We welcome any help from the community in identifying problems or suggesting future information products.

2. Database Introduction

At the JSPC, we use a relational database to contain parameter information for seismic data. This technology provides a high degree of flexibility, allowing data to be entered as it becomes available, and errors to be fixed as they are discovered. All relational databases are simply a collection of flat tables. By flat, we mean a matrix of rows (*records*) and columns (*fields*) with each cell occupied by a single number or character string. Our database model is actually a heterogeneous relational database, since the waveform data is stored in binary files outside the flat tables. The database tables only reference the waveform files, rather than containing them.

Relational databases are usually tied to expensive, proprietary software products: Relational DataBase Management Systems (RDBMS); ORACLE and SYBASE are examples. These RDBMS' are used to manipulate the data, and it is impossible even to examine the data without using the RDBMS. To make our data more generally accessible, the database in a JSPC information product is represented as plain ASCII files, one per table. You can read (and modify *cautiously*) these tables with any text editor. We also provide software which uses these tables directly.

A *schema* refers to the definition of a set of tables and the fields which make up each table. The schema is the most important aspect of the design of a database. DARPA's Center for Seismic Studies (CSS) designed a schema specifically for seismic data. This schema has evolved over more than ten years, and is currently at the heart of an operational seismic processing system used by DARPA to provide global monitoring for underground nuclear testing. Our schema is derived from the CSS version 3.0 schema as defined in Anderson, et. al. (1990). For the most part, it is a direct sub-set of the official CSS schema. The schema reference manual describes the schema in detail. This document provides a brief introduction.

It's convenient to separate the database into two major parts; figures 1 and 2 present a simplified view of these two major parts, showing the tables and their primary contents.

Waveform Data Tables

The first part comprises the information about the waveforms. The **wfdisc** table contains a reference to each data segment for every station and channel, and has the basic information needed to read the waveform data. The much smaller **site** and **sitechan** tables specify the station location, channel orientation and emplacement depth. The **sensor** and **instrument** tables provide calibration and response information. The **instrument** table references external files which contain the response information, which is encoded as either poles and zeros, and/or digital FIR or IIR coefficients. Finally, the **affiliation** and **network** tables provide a way of clustering stations into affiliated groups.

Event Related Tables

The second major part focuses on information derived from the waveform data. Phase picks are assigned an arbitrary key, *arid*, and kept in the **arrival** table. Arrivals are then grouped together by the **assoc** table, and used by a location program to generate hypocenters. Hypocenters are assigned another arbitrary key, *orid*, and stored in the **origin** table. Note that there may be several hypocenters computed for a single event; all are kept in the **origin** table. Events are assigned an arbitrary key, *evid*, and enumerated in the **event** table. Hypocenters corresponding to the same event reference the same event id (*evid*). The **event** table may (should) indicate a preferred origin. Finally, a magnitude may be calculated for arrivals at various stations. These station magnitudes are kept in the **stamag** table. An average of these station magnitudes may be computed; it would then be kept in the **netmag** table.

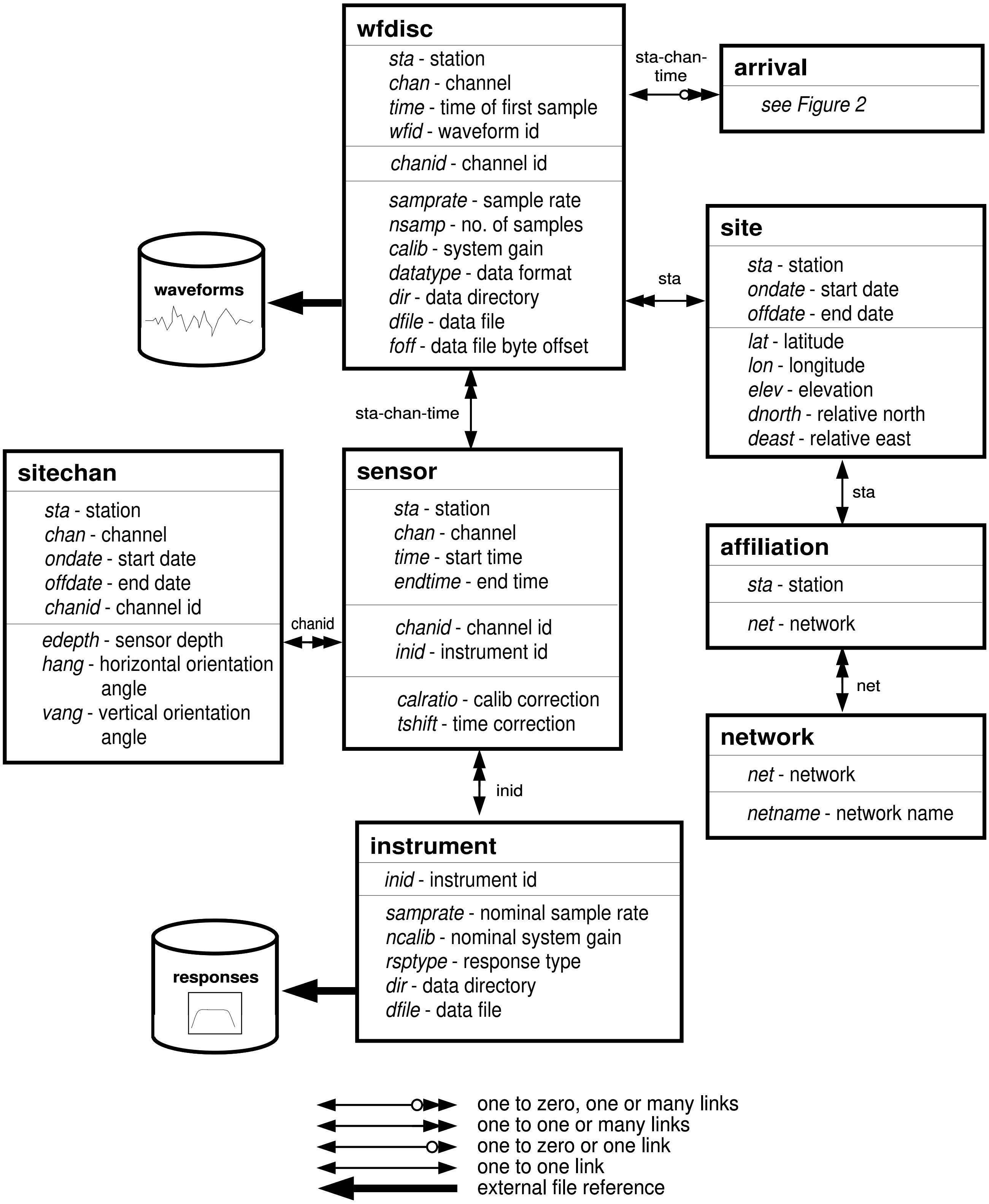

Figure 1. Database schema map for JSPC information products. Waveform data tables.

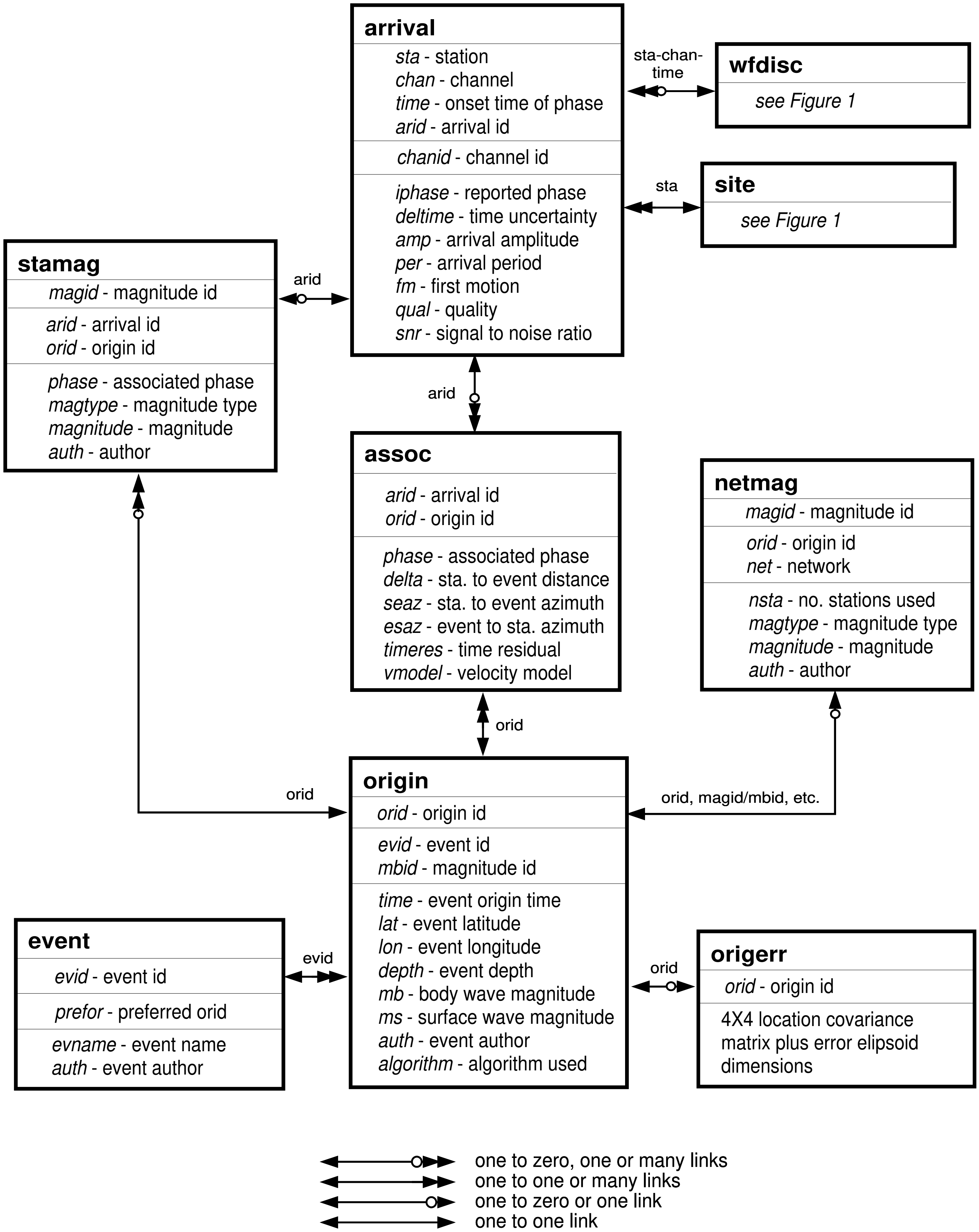

Figure 2. Database schema map for JSPC information products. Event related tables.

The power of a relational database originates in the logical grouping of information into tables, in such a way that information is not duplicated. This means, for instance, that the station latitude and longitude are specified once in a single table, instead of repeatedly within fixed data headers attached to the waveforms (as in SAC or AH formats). This simplifies maintaining the accuracy of the information.

Unfortunately, this simplicity has a cost: information may not be immediately accessible; you may need to find it in another table. For instance, while looking at a waveform, you may want to know the latitude and longitude of the recording station. However, that information is not in the **wfdisc** table; you must look it up in the **site** table. This process of finding associated information in different tables is called *joining* in database parlance.

If the **wfdisc** table were joined with the **site**, **sitechan**, **sensor** and **instrument** tables, this would form a large table in which each row would contain all the related parameter information (like latitude and longitude) for each waveform. Each row would correspond somewhat to the header for a SAC or an AH file. A DBMS provides a language (usually SQL) which facilitates joins. The JSPC provides the **db** library which facilitates the join process. In addition, the particular join described above is provided through the **scv2** library. This software is described at more length in the software section, and in the man pages.

The linkages between the various tables in the database is fairly intuitive. For instance, each waveform described in the **wfdisc** table has a station code, a channel code, and a time period. To find the associated latitude and longitude, you must look for the same station code in the **site** table. To find the associated channel orientation, you must look for the same station code and channel code in the **sitechan** table. Calibration and response functions often change over a period of time, and so to find these parameters, you must look in the **sensor** table for the same station and channel codes, at the particular time the waveform was recorded. (Actually, you must also look into the **site** and **sitechan** table for the particular time the waveform was recorded, though in practice, the parameters in these tables are more stable.)

For the derived, event related information, the joins are mostly specified by synthetic keys, like *arid*, *orid*, *evid* and *magid*. These keys have no special significance in and of themselves, they are just arbitrary numbers assigned in an ad hoc way. (In contrast, the primary keys in the waveform related files, the station and channel codes, and the time are somewhat standardized. Their values usually have some significance beyond their use as keys in the database).

There are three varieties of linkages between rows in the database tables. The simplest linkage is a one to one (or zero) link in which a single row in table A is linked to a single row in table B. Such a link exists between the **origin** and **origerr** tables. Second, a one to many (or zero) link exists when a single row in table A is linked to zero or more rows in table B. The link from **arrival** to the **assoc** table or from **site** to **wfdisc** are examples. Finally, there are many to many links; these must be mediated by an intermediate table. An example is the multiple links between **origin** and **arrival** tables.

The relational database contained in an information product is represented as a set of ASCII fixed-format files. Each table is contained in one or more files. The table filenames are composed as *databasename.tablename*. For example, if the database is named **cnet**, the **wfdisc** table is contained in the file *cnet.wfdisc*. The precise definitions of the fields and their character positions in each table are given in the *Schema Reference Manual*. It may be helpful here to mention the more important conventions.

- Distance units are in kilometers. For instance, *elev* (station elevation), *depth* (source depth) and *edepth* (receiver depth) are all expressed in km. The exception to this is the event to station distance, *delta*, in the **assoc** table which is expressed in degrees.

- All angles (eg, latitude and longitude) are expressed in decimal degrees.
- Times are expressed in three different ways in the database: as epoch time, as julian days, and as a date. Epoch times are expressed in seconds: the nominal number of seconds since 1970 Jan 1 00:00:00 GMT. Nominal means that leap seconds have not been included, although leap days have. In the software, epoch times are internally represented as double precision floating point numbers which gives sub-millisecond accuracy. Note that an epoch time can be positive or negative, to account for times prior to 1970 Jan 1. Julian days are the concatenation of the year with three digits specifying the day of the year, for example 1987083. *Jdate*, *ondate* and *offdate* are time fields expressed in these Julian days. Finally, there are plain dates, which are usually expressed as MM/DD/YYYY. The *lddate* fields are expressed in this format.
- The units of the actual waveform data samples within the external files can be anything. Waveform amplitude units are standardized through the *calib*, *calper*, and *segtype* fields within the **wfdisc** table. Depending on the value of *segtype* ('D', 'V', or 'A'), *calib* is the amplitude factor, in nm (nanometers), nm/s (nanometers/second), or nm/s^2 (nanometers / second squared) per external amplitude unit, that is valid at *calper* period for that waveform segment.
- Instrument responses are specified as displacement, velocity or acceleration, in the analogous way, depending on the value of the *rsptype* field ('D', 'V', or 'A') in the **instrument** table. The instrument response files in directory *tables/response/* are ASCII files containing poles and zeroes, and coefficients for digital FIR filters. Comment lines indicate some additional information. These instrument response files are referenced by the **instrument** table and their contents are automatically included as part of the waveform segments in the software libraries. Instrument responses are normalized at *calper* period so that the total system response function can be computed by dividing the normalized response amplitude by *calib*.

3. Software Tools Introduction

A small set of programs are provided. These programs may be used to inspect the data, and to convert the data to SAC or AH format. Manual pages are provided for each of these programs, both in the *sw/man* directory in the installation directory, and in postscript form in the *docs* directory. Following is a brief description of each program.

dblook This program is a read only spreadsheet-like program which provides viewing of any of the tables within a database. It is mouse driven and allows scrolling through tables with large numbers of rows and rearranging of table columns. Each table is represented by a separate window. Clicking the mouse within rows in the **wfdisc** table causes additional windows to appear with the waveform segments displayed as trace plots. Clicking the mouse on a row of the **instrument** table displays the corresponding instrument response.

dbpick This is an interactive waveform viewing and phase arrival picking/editing program. Multiple stations and channels can be displayed within the same window. Subwindows can be spawned with different arrangements of traces and magnify subwindows can be made for arrival picking. Phase arrivals can be added or deleted. The time and phase code of existing arrivals can be modified. Phase uncertainties associated with a pick can be entered, and amplitudes and periods can be measured directly from a trace. Time scrolling/zooming and arrival editing are done with the mouse. Filters can be applied to any traces. A variety of amplitude scaling options exist. In addition to a mouse driven user interface, a type-in command driven user interface can be used. Keyboard accelerators can be used to enable/disable automatic time scale fitting to waveform segments or to jump scroll to the next or previous event, phase arrival or waveform segment.

dbmapevents This program will plot events from a database on a map using equal distance projection. A cartography database is required and is included in this information product. Event symbols can be color and size coded in a variety of ways and multiple hypocenters per event can be displayed. A simple interactive mode is supported for panning and zooming on the map. A color PostScript file of the map is created.

db2sac This program will convert data from a database to SAC form.

db2ah This program will convert data from a database to AH form.

4. Software Libraries Introduction

In addition to the programs described above, source for several useful libraries is provided. These libraries provide access to the database and travel time curves, coordinate transformation, graphics and some general utilities. The documentation for the libraries is of variable quality; some are not documented at all, others have extensive sets of manual pages. The **db** (database) and **scv2** libraries are fairly completely documented. Together, they provide the primary means for programmers to directly use the database. Both C and FORTRAN interfaces are provided. Following is a brief description of the library sources provided; refer to the manual pages (or the source itself when no manual pages exist) for more detailed information. The manual pages for the libraries do not arrive installed; you must run make in the *sw* directory to install the library manual pages.

db The **db** library provides basic read/write access to database tables and fields, as well as more complex operations like joins, sorts and expressions. Both C and FORTRAN interfaces are provided, and a large number of manual pages exist. Start by reading dbintro(3).

scv2 Libscv2.a provides a particularly useful joined view of the information contained in the tables, called a station-channel view. A single station-channel object consists of all waveform segments and associated phase arrivals for an arbitrary time period. It includes all station, channel and instrument parameters including station latitude, longitude and elevation, sensor orientation angles, sensor emplacement depth, total system gain and instrument response function. In other words, using the Station Channel View interface allows the programmer to avoid dealing directly with the multiple files in which the parameter information and waveform data are contained, and concentrate on his/her particular problem. Both C and FORTRAN interfaces are provided, and the interface is documented in several man pages (try man -k scv).

response This library provides utility procedures for reading the external instrument response files and for computing complex instrument responses as a function of station, channel, time and frequency. This library has Fortran and C interfaces, and is documented in response(3) and response(5).

gpl The **gpl**, or gather plot, library provides the X-window based interactive trace plotting functionality used in programs **dblook** and **dbpick**. This has a C interface only, and has no documentation.

ol This library is a simple home-brew X-window widget library that is used by the **gpl** library. This has a C interface only, and has no documentation.

coords Libcoords.a contains a variety of coordinate transformation functions. Probably the most interesting are the routines for conversions between epoch time (seconds since 1970) and commonly used time representations like "May 21, 1992 5:30 AM". There are both C and FORTRAN interfaces for most routines, and manual pages for everything. Try looking at epoch(3).

tttaup Libtttaup.a is an interface to a travel time library, derived directly from the IASP-91 subroutines written by Ray Buland and available through the IRIS DMC bulletin board. This has primarily a C interface to an underlying set of FORTRAN routines. The documentation is largely restricted to the original IASP-91 documentation from the IRIS DMC distribution.

tbl This library provides a simple XView based table spreadsheet window and is used by program **dblook**. There is only a C interface, and no documentation.

grx Libgrx.a provides a set of routines for drawing graphs and other two dimensional figures in color for X-windows and postscript output. This library has only a FORTRAN interface (which can be called from C, of course), and has a number of man pages.

stock This library provides general utility functions such as associative arrays and variably sized lists, searching and sorting, GNU regular expressions, string processing, parsing argument lists and error handling. Generally, these routines have both C and FORTRAN interfaces, and all have manual pages.

The **db** and **scv2** libraries provide a fairly comprehensive interface to the relational database contained in this information product. The **db** library can be used to read or write any of the information from the database tables, as well as join, sort and compute expressions on tables. The **scv2** library provides joining functionality by composing a fixed view which is intended to satisfy most user queries that would be made about a single channel of seismic data. The station-channel view abstraction was created primarily from a research seismologist's standpoint and should be useful for accessing the database without mastering the details of the schema.

5. Installation Procedure

A JSPC Information Product Tape may contain a few gigabytes of waveform data and associated parameter information, documentation, and software. The installation process facilitates unloading smallish portions of this information in order to avoid overwhelming local disk space. At the JSPC, we use Sun workstations exclusively; consequently our information products tend to be oriented toward Sun workstations running UNIX. It should be possible to extract the software and data on other architectures, however, you may find it necessary to a) modify the installation process somewhat, b) possibly deal with byte ordering problems for the waveform data and c) compile the software for a different architecture.

The tape actually has a number of files on it; these are described in more detail in the later section **Manual Installation**. The installation script will install these tape files into a hierarchy as shown under *Installation Directory* below. Be careful to maintain this directory structure, as the software depends on the organization shown.

Installation Directory

docs/	postscript versions of documentation		
sw/	software		
	bin/	executables	
	lib/	libraries	
	include/	include files	
	man/	manual pages	
	proto/	makefile prototypes	
	data/	program data files	
		tables/	travel time tables
		schemas/	database schema definitions
	src/	software source	
		bin/	source for commands
		lib/	source for libraries
tables/	database tables		
wf/	waveform data		

5.1 Extracting the installation scripts

On a Sun workstation running SunOS 4.1.x, simply run the standard utility **/usr/etc/extract_unbundled**. Answer the questions about what tape drive and machine the tape is mounted on. The **extract_unbundled** script then reads in the second file from the tape and runs the script **install_unbundled** (from the tape). (This step may fail if a previous run of **extract_unbundled** has left a copy of **install_unbundled** in **/usr/tmp/unbundled**. If so, you need to remove or rename the old copy of **/usr/tmp/unbundled** -- try **mv /usr/tmp/unbundled /usr/tmp/unbundled.old**. You may need root permission to remove the old files, since **extract_unbundled** is often run as root).

If you are not using a Sun or SPARC compatible workstation, or are running Solaris, you will need to load the installation scripts by hand. The second file on the tape contains the installation scripts. All the files except the first are written with a blocksize of 126 blocks. Use **mt** and **tar** to extract the installation scripts:

% **mt -f <no-rewind-tape-device> fsf 1**

% **tar xvfb <no-rewind-tape-device> 126**

This should unload the following files into your directory:

README	descriptive file
extract_unbundled	script to extract data from the tape
install_unbundled	actual installation script (run by extract_unbundled)
TOC	Table of Contents for the tape
tapefileX.contents	list of contents of file #X on tape

Now, run the script **extract_unbundled**. The script (and install_unbundled, which it runs) use the following programs:

awk	**mkdir**	**sh**
cat	**more**	**tar**
dd	**mt**	**true**
dirname	**mv**	**uncompress**
echo	**rsh**	

On some machines, you may need to edit **install_unbundled** . The scripts have been tested on Sun systems running SunOS 4.1.x and Solaris, and on a DEC machine running ULTRIX, and an IBM machine running AIX. The tar on AIX does not have an option to use the contents of a file as a list of files to extract, and consequently the script may fail if you attempt to extract a large portion of the waveform data at once. It is possible to use GNU tar under AIX, however.

Run **extract_unbundled** ::

% **./extract_unbundled**

5.2 Extract the files you want

You will be asked what directory you wish to install into; answer with the path of a directory. As mentioned above, the entire tape might require gigabytes of free space; you will need about 50 Mbytes free to extract the documentation, software, and a small sample of the data.

At this point, you will be presented with a list of options; each option reads some of the tape into the directory you selected. You will be presented with this set of options multiple times until you select quit. The choices should be self explanatory; extracting the documentation, the parameter data, the software executables, the software source, and the waveform data. The first time, you should probably extract the documentation, the binary software (if you have a Sun running SunOS 4.1.x), the parameter data, and some subset of the waveform data.

5.3 Manual Installation

The tape is organized as follows:

Tape Table of Contents

File no.	Approx Size	Format	Contents
1	1 Kb	dd	descriptive file
2	2 Mb	tar	installation scripts
3	10 Mb	tar	documentation
4	10 Mb	tar	parameter data (database tables)
5	15 Mb	tar	software executables
6	15 Mb	tar	software source
7	84 Mb	tar	cartography database file
8-?	?? Mb	tar	waveform data

You may choose to extract the data by hand, using **mt** and **tar**, rather than using the installation scripts. Consult your local manuals for assistance with **mt** and **tar**.

5.4 Running Programs

If you install the executable software, follow the instructions in the file *sw/README.bin* to run programs. Basically, you need to include the unloaded *sw/bin* directory on your path, and set the environment variable JSPC to point to the directory *sw*. You may also wish to include the *man* directory on your MANPATH.

If you've installed the contents of the tape in *<IDIR>*, then the following c-shell commands would set up your environment correctly:

setenv JSPC *<IDIR>*/sw

set path=($JSPC/bin $path)

setenv MANPATH $JSPC/man:$MANPATH

In addition, if you're using Sun's Open Windows, you should add the following line to the *.Xdefaults* file in your home directory

OpenWindows.FocusLenience: True

Other environment variables which may interest you are described separately in the file *sw/Environment*.

The executables provided were statically linked, and should run on any Sun running 4.1.x. If you are using Solaris on a Sun, you must recompile -- see the next section.

5.4 Compiling Software

If you extract the software source and wish to compile it, follow the instructions in the file *sw/README.src*. On a Sun system, you should be able to simply run **make config**, and then **make** in the *sw* directory. On other systems, problems may arise in conjunction with the makefiles and possibly the source itself which are beyond the scope of this manual.

Appendix A: Installation Example

Following is an example of the installation process on a Sun workstation running SunOS 4.1.x. The process should be the same for other workstations, once you extract the installation scripts and run **extract_unbundled** from the tape..

lemond% **/usr/etc/extract_unbundled**

Enter media drive location [local | remote]: **remote**
Enter hostname of remote drive: **spice**

Enter Device Name (e.g. rst0, rmt0, rfd0c) : **/dev/rst9**

Please mount the release media if you haven't done so already.

Press return when ready:
The following product will be installed:
****

This tape contains triggered seismic data collected at the Caucasus
Network for January 1992 to November 1992.

The tape was assembled by IRIS' Joint Seismic Program Center at
the University of Colorado at Boulder.

For further information, consult the documentation from
this tape, or contact JSPC directly:

Joint Seismic Program Center
University of Colorado
Campus Box 583
3100 Marine Room 357
Boulder, CO 80309-0583
303/492-5243
problems@jspc.colorado.edu

****

Do you want to continue [y|n]? **y**

****

JSPC INFORMATION PRODUCT Installation Script

****

This script extracts all or parts of a JSPC Information Product tape.
Please begin by extracting the documentation.

In what directory do you want to load things?
[/auto/home/moser/danq/knet_triggered/extract] **/spc/lemond1/cnet**

/spc/lemond1/cnet does not exist; make it? **y**

Choose a file number from the table of contents:
(Please begin by extracting the documentation.)

2	Documentation
3	parameter data (CSS database tables)
4	Software (executables only)
5	Software (source only)
6	Cartography database file (needed for dbmapevents)
7	Waveform data

You may also choose:

a	All the files
q	Quit

Your choice: **2**
Selecting files from Documentation
Extracting files from tape; this will take some time...

Choose a file number from the table of contents:
(Please begin by extracting the documentation.)

2	Documentation
3	parameter data (CSS database tables)
4	Software (executables only)
5	Software (source only)
6	Cartography database file (needed for dbmapevents)
7	Waveform data

You may also choose:

a All the files
q Quit

Your choice: **3**
Selecting files from parameter data (CSS database tables)
Extracting files from tape; this will take some time...

Choose a file number from the table of contents:
(Please begin by extracting the documentation.)

2 Documentation
3 parameter data (CSS database tables)
4 Software (executables only)
5 Software (source only)
6 Cartography database file (needed for dbmapevents)
7 Waveform data

You may also choose:

a All the files
q Quit

Your choice: **4**
Selecting files from Software (executables only)
Extracting files from tape; this will take some time...

Choose a file number from the table of contents:
(Please begin by extracting the documentation.)

2 Documentation
3 parameter data (CSS database tables)
4 Software (executables only)
5 Software (source only)
6 Cartography database file (needed for dbmapevents)
7 Waveform data

You may also choose:

a All the files

q Quit

Your choice: **7**

Selecting files from Waveform data

Please choose from the following options to
create a list and extract files :

A)ll the files
L)ist the files available on the tape.
S)elect files from list (with egrep).

V)iew the current extraction list.
D)elete files from the current extraction list.

X)tract files
Q)uit without extracting files.

s
Select: **1992/142**
adding 0.1 Mbytes in 18 files to extraction list.
You have selected 0.1 Mbytes in 18 files.

Please choose from the following options to
create a list and extract files :

A)ll the files
L)ist the files available on the tape.
S)elect files from list (with egrep).

V)iew the current extraction list.
D)elete files from the current extraction list.

X)tract files
Q)uit without extracting files.

x
Extracting files from tape; this will take some time...

Choose a file number from the table of contents:

(Please begin by extracting the documentation.)

2	Documentation
3	parameter data (CSS database tables)
4	Software (executables only)
5	Software (source only)
6	Cartography database file (needed for dbmapevents)
7	Waveform data

You may also choose:

a	All the files
q	Quit

Your choice: **q**
Quitting installation script

Appendix B: Using the Software: Example

This is a brief tutorial which will bring up a number of windows where you can look at and play with the data.

1. You must have installed the software and set up your environment as described earlier. If you installed the software and data into *IDIR*:

    set path=(*IDIR*/sw/bin $path)

    setenv JSPC *IDIR*/sw

2. You must be running X-windows or OpenWindows.
3. Change directories to *IDIR/tables*, and run the program **dblook**:

    cd *IDIR*/tables

    dblook cnet &

    A window should appear with a button for each database table along with a Quit button.

4. Click the left mouse button on the wfdisc button. Another window should appear with a display of the **wfdisc** table. You can scroll through this window be clicking on the vertical or horizontal scrollbars. Click the left mouse button on one of the rows in the table, somewhere in the middle of the row. Another window should appear with the waveform segment displayed. Click on the Done buttons in both the trace display and wfdisc display windows.
5. Click the left mouse button on the instrument button. Another window should appear with a display of the **instrument** table. Again, you can scroll through the table using the scrollbars. Click on a row to bring up a display of the response function. (Be patient; this will take a few seconds.) You can kill this response function display with control-c in the window, or just close the window.
6. Run the program **dbpick**:

    dbpick cnet

    It will take a few seconds for the program to start; be patient. After the prompt,

    Do you wish to see it?:

    type

    y

    A window will appear with seismic data, and phase picks indicated. In the original window where you started the program, a new prompt, *dbpick>*, will appear.

7. In the original window, after the *dbpick>* prompt, enter the following.

    tw 150

    dw

    ae

    The time window in the display changes to 150 seconds, then the blank traces are deleted, and finally the displayed picks are associated with an origin, and predicted arrivals are shown.

8. Choose one or more trace(s) by pressing the left mouse button on the station/channel label on the left side of the display. Then choose **NewWin** under the Traces menu button at the top of the screen. A new window will appear with only the selected trace(s) in it.
9. Place the cursor in the new window, press and hold down the middle mouse button and drag the display left and right along the time axis. When the button is released, the display is redrawn.

10. Hold down the shift key, then select a region around an arrival by clicking the left mouse button first to the left of the arrival and then to the right. The selected region is expanded to take up the entire window.
11. Hold down the control key, and press the mouse on an arrival flag. A visual indication of the arrival time uncertainty as a hollow box, and the amplitude and period measurement are shown as a solid (filled) box. The amplitude and period measurements are shown in the upper left corner.
12. Experiment with various filters by selecting them from the Filter menu button. When you are through, click on the **DONE** button to close the window and return to the main window.
13. Note that the amplitude scaling is automatic for each trace. To force automatic amplitude scaling on the first trace only with the other traces using the same scale, press and hold the left mouse button on the upper button labeled **AMP:** A which will display a pull-down menu. While holding the left mouse button move down the menu to **Auto1** and then release the button.
14. Place the cursor in the trace display window and press the n key on the keyboard once. The display will jump scroll to the next arrival flag. The display will jump scroll to the next arrival flag every time the n key is pressed. If the p key is pressed then the previous arrival is displayed. The N and P keys behave similarly, but jump to later and earlier waveform segments regardless of whether an arrival flag is present.
15. Reset the amplitude scaling to **Auto** and type the following sequence.

    sw 0 sc * fw tfit dw ae sw 1

    This sequence of commands causes the waveform display to be turned off (**sw 0**), the set of station-channels to be reset (**sc ***), the display to be time scrolled to the first set of waveforms (**fw**), the time window to be put in auto-fit mode so that it automatically fits itself in time to the displayed waveform segments (**tfit**), all dead channels to be deleted from the display (**dw**), the database is searched for an event that associates with the displayed arrivals (**ae**) and the waveform display is turned back on (**sw 1**). The **sw 0 ... sw 1** command sequences are used often so that the scrolling and channel re-arranging occur quickly without having to incur the expense of redrawing the wiggle traces during intermediate steps. The display should look approximately the same as after step 7. Now the following sequence of commands can be entered repeatedly to scroll the display to the next set of waveforms and display associated events.

    sw 0 ne na dw ae sw 1
16. To quit **dbpick** go back to the original typein window (by moving your cursor to that window) and type

    quit

    Quit **dblook** by clicking on the **Quit** button.
17. Run the program **dbmapevents**:

    dbmapevents cnet KIV 180.0 -c

    It will take a few seconds for the program to start; be patient. A window will appear and a map of the world will be drawn centered at station KIV. After another pause the events from the database will be displayed with symbols that are color coded according to whether or not they are in the PDE catalog and size coded according to magnitude. Station locations are shown as gray triangles. A cross-hair cursor will appear in the window and the mouse pointer will be warped to the window cursor.

18. Type the following character (no RETURN is needed).

    ?

    A help menu is printed in the original window where **dbmapevents** was started.

    Type the following character.

    I

    The map zooms in by a factor of 5 to show the European, Middle Eastern and Western Asian region.

19. Type the following character.

    I

    The map zooms in by a factor of 5 again to show the region around the Caucasus.

20. Type the following character.

    6

    The symbol color coding according to whether or not the event is in the PDE catalog is disabled.

21. Type the following character.

    2

    The symbol color coding is now keyed on event depth.

22. Move the cursor to an event of interest and type the following character.

    e

    The closest event is identified and its characteristics are listed in the original window where the program was started.

23. Move the cursor to a station of interest and type the following character.

    s

    The closest station is identified and its characteristics are listed in the original window where the program was started.

24. To quit **dbmapevents** type the following character.

    Q

    A file named dbmapevents.ps will contain the last map displayed in color PostScript form.

This concludes the brief tour of **dbmapevents**, **dbpick** and **dblook**. Consult the man pages for more detailed descriptions of each.