

DB Ingest

# Database Ingest for EXOSAT DBMS

**HEASARC**

Laboratory for High Energy Astrophysics, NASA/GSFC, Code 662,  
Greenbelt, MD 20771

17 October 2002

# 1 Introduction

This document describes the installation and usage of the software that ingests the database tables provided as TDAT into the EXOSAT database system (DBMS), Browse. It also provides a brief description of available library routines that can be used to create a TDAT file. It assumes that all sites have already available the EXOSAT DBMS software, Perl version 5.005.02 (or higher), a F77 compiler, and a recent version of HEASoft. Building the enclosed software requires the HEASoft libraries and the LHEASOFT environment variable should be set accordingly.

A TDAT is an ASCII file with an header describing the fields followed by the data. The data is usually formatted as one record per line, with each field separated by a character defined to be the field delimiter, usually a vertical bar character (|).

Documentation on the TDAT file is available at

<http://heasarc.gsfc.nasa.gov/docs/software/dbdocs/tdat.html>

If questions arise concerning the usage of this software or a bug is discovered in the course of testing, please contact [sabol@lheamail.gsfc.nasa.gov](mailto:sabol@lheamail.gsfc.nasa.gov). Provide a detailed description and the necessary files if applicable.

## 2 Content of the distribution

The directory structure of the software distribution is divided as:

`bin/ examples/ install/ lib/ work/ doc/ install-it.sh`

The `doc/` directory contains this documentation.

### 2.1 The bin directory

The `bin/` directory contains several Perl modules. These are :

- `cp_done.pl`. Copies the collection of files used by the EXOSAT Browse DBMS from the working directory to the production area for database table creation or updating.
- `createX2.pl`. This creates a table for the EXOSAT DBSM system using SQL. It uses a lock file as a semaphore to prevent simultaneous changes to ZZDB. This can be useful when there are many people ingesting databases at the same time. Otherwise, corruption of the ZZDB Browse system table is possible.
- `NewSubs.pl`. Creates the descriptive files for the EXOSAT DBMS based on the information available in the `.info` file.
- `tdat2browse.pl`. Takes a TDAT file or a TDAT header file and creates all the files necessary for populating the EXOSAT DBMS.

## 2.2 The examples directory

The *examples/* directory contains examples of tables that can be used for software testing. The tables are provided in his original form together with the scripts that create TDAT files and EXOSAT Browse data files that can be ingested by *db\_pop*. Examples are included for the following database tables: the ASCA Galactic Plane Survey (ASCAGPS), the Chandra Deep Field South 1-Megasecond Catalog (CHANDFS1MS), the ROSAT PSPC Catalog of the Pleiades (PLEIADXRAY), the Washington Double Star Catalog (WDS), and all the current planned Swift database tables (fake data).

## 2.3 The install directory & the installation script

The *install/* directory contains all the scripts necessary to install the software included in this distribution. The scripts are :

```
change_browse_bin.pl
change_browse_lib.pl
change_browse_utils.pl
change_install_dir.pl
change_makefiles_for_linux.pl
change_mkhelp_location.pl
change_perl_location.pl
```

The scripts are used within installation script *install-it.sh* available in the top directory of this software distribution. The installation instruction are given in the next section.

## 2.4 The lib directory

The *lib/* directory contains several library files and subdirectories with additional libraries. These are :

- *Astro/* is a directory. Contains several routines that operates on coordinates and on name resolver.
- *Astro.pm* combined a subsets of modules present in the *Astro/* is a directory and used by several utilities.
- *Create\_DB.pm* contains subroutines used to create the EXOSAT DBMS tables
- *DBMS.pm* configure the environment for Sybase, Ingres, or EXOSAT DBMS
- *DBUtils/* is a directory. It contains *Class.pm*, that assigns the Browse classification based on the object type, and *CoordConv.pm*, that contains subroutines for converting astronomical coordinates. These are wrappers around the *Astro.pm* object-oriented programming interfaces.
- *DBUtils.pm* routines used to create TDAT file.

In the examples directory there are already made Perl scripts for a sample of tables. Each modules requires the setting of various configuration variables that describe the input and

output files. The `process_record()` routine is the code that is the most commonly used. The `$_::current_file` variable contains the name of the input file being processed and can be used for file-specific processing. A pre-processing of the input files and post-processing as well through the `$_::content_filter` and `$_::post_processor` variables.

- `DateTime.pm` module that convert dates in different formats.
- `GenMods.pm` contains routines to convert dates and format for different variable types. These routines are used when creating an EXOSAT DBMS.
- `HDBclass.pm` translates the HEASARC object classification code into a descriptive character string
- `HDBingest.pm` Perl module for ingesting a TDAT file into either Sybase or INGRES using `HDBingest`.
- `HTML.pm` routines that operates on HTML files
- `Logger.pm` module for keeping log files.
- `Math/` is a directory. It contains `ErrorAnalysis.pm`, a Perl module for error analysis.
- `Nice.pm` miscellaneous routines (clean up and/or check strings, lock a database file)
- `SLA.pm` is derived from C `SLALib` routines. The C source for the `SLALIB` routines should be consulted to understand the range of validity of these results (e.g., with regard to the precession calculations).
- `SQL/` is a directory. It contains `Exec.pm` which has a collection of routines that execute SQL commands to interface with Sysbase or Ingres.
- `Test.pm` module to facilitate database creation
- `UpdateDB/` is a directory. It contains the common routines and the `UpdateDB` configuration for the Swift databases
- `UpdateDB.pm` routines to update database table in Sysbase, or EXOSAT DBMS
- `Util.pm` miscellaneous routines

## 2.5 The work Directory

This directory contains under the directory `create/` two subdirectories `programs/` and `template/`. The `template/` directory contains the following set of files:

```
makefile  README.templates
```

The `tdata2browse.pl` uses the makefile template to compile a `db_pop` executable specific for each database. The `programs/` directory contains four subdirectories: `f77/`, `lib/`, `db_pop/`, and `mkhelp/`. The `f77/` directory contains routines used by the `db_pop` programs. The `lib/` directory is empty and will be populated with a library that is created when the software is installed. (See later step in the installation section.) The `mkhelp/` directory contains the source code for the `mkhelp` program which creates the help files used by EXOSAT DBMS. The `db_pop/` directory contains the generic portion of the source code required to compile a `db_pop` executable.

## 3 Installation

The HEASARC database utility software was originally developed as internal tool to be used by the HEASARC database developers only. As such, the software makes certain assumptions about the database development directory hierarchy in use at the HEASARC that may require some customization of the source code in order to work in other environments.

After the software is installed, it is recommended to follow one of the examples to test if the installation is correct.

### 3.1 Configuring and Installing Files

Download the tar file in any directory, ungzipped and untar the file. Edit the *install - it.sh* script and change the installation parameters to reflect the directory configuration on the local system. The installation parameters are :

*INSTALL\_DIR* : This parameter should be set to the directory name where this software is finally installed. The default name is */dba\_dbase/*. By default, the Perl modules in the *lib/* directory are assumed to be located at */dba\_dbase/lib/*. Many of the modules and software assume that the necessary support files for each database table are located at */dba\_dbase/work/tablename/* and that the EXOSAT DBMS command-line environment is assumed to be installed in */dbase*. Because of this, it is recommended to create a */dba\_dbase/* directory on the local system. However, it is certainly possible to set *INSTALL\_DIR* to a directory path different from */dba\_dbase/* and the installation procedure should change all the reference to the installation directory accordingly. In the following instructions it is assumed that */dba\_dbase/* is the installation directory.

*PERL\_LOC* : This parameter should point to the location of Perl (version 5.005.02 or higher) interpreter on the local system.

*MKHELP\_LOC* : This parameter should be set to the location of the *mkhelp* executable used to construct the Browse help files. Within the browse software the default location for *mkhelp* is */dbase/bin/mkhelp*. However if there is not a working version of *mkhelp* on the local system, the source code is provided within this distribution in the *work/create/programs/mkhelp/* directory. It is recommended to install the *mkhelp* version included in this tar file in */dba\_dbase/bin/*. After compiling *mkhelp* install it at *MKHELP\_LOC*. WARNING: the HEAsoft distribution includes a *mkhelp* program, however the output files are not always readable by browse. It is recommended therefore to install the version provided within this tar file.

*BROWSE\_LIB* : This parameter should be set to the Browse libraries directory. The default location is */dbase/lib/*.

*BROWSE\_BIN* : This parameter should be set to the Browse executables directory. The default location is */dbase/bin/*.

*BROWSE\_UTILS* : This should be set to the Browse *utilities/* directory which should contain the *db\_pop/* directory. The default location is */dbase/manager/utilities*. However if there not a working version of *db\_pop*, the *db\_pop* source code is provided within this tar file in the *work/create/programs/db\_pop/* directory. It is recommended to install the *db\_pop/* directory in the *INSTALL\_DIR/db\_pop/* to not make confusion with the version included with the Browse distribution.

After setting all these parameters, the installation script is ready to be run. To run it type:

```
% ./install-it.sh
```

### 3.2 Add Command to the Path

To be able to run the software included within this tar file, modify the path. For csh or tcsh with the software installed in */dba\_dbase/*, type :

```
% set path = ( /dba_dbase/bin $path )  
% rehash
```

If the software is not installed */dba\_dbase/* use the value set in the *INSTALL\_DIR*. Add the above commands to the login file to avoid repeating these steps in the future.

### 3.3 Customize Environment Settings in DBMS.pm

The Perl module DBMS.pm (located under *lib/* directory) may have to be modified so that it sets the environment variables for using your EXOSAT DBMS installation. The variables to change are in the Configure() subroutine in the DBMS.pm file, after the "Xray Browse environment variables" comment line. The values to change are the following:

```
# Xray Browse environment variables  
'DATABASE_NAME' ; environment variable set to run Browse  
'DATABASE_HELP' ; location of the help file '/dbase/doc/sql300.dhf'  
'USER'          ; User label managing the database population.  
                  Within Browse users allowed to populate  
                  database are id with a label XUSER_N where N is  
                  a number.
```

The settings for Sybase and Ingres can be ignored.

### 3.4 Build the F77 Library

To compile *db\_pop*, the program that ingests database tables into the EXOSAT DBMS, it is necessary to link to the libdbutils.a. The code for creating the library is located under *work/create/programs/f77/* and it needs to be compiled.

Please note that GNU make is required to compile this library.

```
% cd /dba_dbase/work/create/programs/f77/  
% gmake
```

Alternatively, the commands make or gnumake can be used. This depends on how GNU make is installed locally. The library is created in the */dba\_dbase/work/create/programs/lib/* directory.

### 3.5 Optional: Compile and Install db\_pop.o

The *db\_pop* program needs to link to *db\_pop.o*, the generic portion of the code that ingests database tables into the EXOSAT DBMS. This step is only necessary if on the local system

there is not available a *db\_pop* directory or if the code with the local EXOSAT DBMS installation (typically in the *manager/utilities/db\_pop/*) is not properly working.

Before compiling *db\_pop* modify the *usitedef.inc* file if the *BROWSE\_UTILS*

To compile *db\_pop.o*:

```
% cd /dba_dbase/work/create/programs/db_pop/  
% make
```

If the *BROWSE\_UTILS* is set to a long path it is possible that during compilation an error occur. This is probably due to the fact that in the include file *usitedef.inc* the path written in a data statement exceed the standard Fortran 80 characters. If that is the case, add continuation lines as necessary to the DATA statement. Recompile again *db\_pop*.

After *db\_pop.o* has been compiled, move the *db\_pop/* directory to the location set for *BROWSE\_UTILS* variable during installation. For example, if *BROWSE\_UTILS* is set to the recommended default location (and there is not a working version of *db\_pop* on the local system), the commands are:

```
% cd /dba_dbase/work/create/programs  
% mv db_pop /dba_dbase/db_pop
```

### 3.6 Build and Install mkhelp

The *mkhelp* program is used to create the help files used by the EXOSAT DBMS.

To build *mkhelp*:

```
% cd /dba_dbase/work/create/programs/mkhelp  
% make
```

Once the *mkhelp* executable has been built, this needs to be installed in the location set for the parameter *MKHELP\_LOC* . For example, if *MKHELP\_LOC* is set to the default location, then the installation step would be:

```
% cp mkhelp /dbase/bin/
```

### 3.7 Customize Makefile Template

Customize the makefile template in *work/create/templates/* so that it will properly compile *db\_pop* in the local environment. Edit the makefile and check that the path to the libraries distributed with HEAssoft is correct. The path is identified in the makefile by the variable *LHEASOFT* and make sure that it is properly initialized by the HEAssoft set-up script before using the makefile.

### 3.8 Customize UpdateDB.pm E-mail Addresses

Change the *%contacts* configuration variable in *lib/UpdateDB.pm* so that automated processes will be able to send e-mail notifications whenever new TDAT files are ingested. There are three

different recipients for the e-mail and they are identified by 'success', 'cc' and 'error'. The 'error' and the 'cc' should be set to the ingest operator, while the 'success' should be set to e-mail a list of people that needs to be notified after the ingest is successful.

## 4 Software to Ingest Database Tables: `tdata2browse` and `NewSubs`

The EXOSAT DBMS requires a number of files associated to the each tables. These files are:

- \*\_cat.dat general information about the specific database table (descriptive)
- \*\_par.dat parameter listing and their format (descriptive)
- \*\_lsm.dat line summary file containing the display format of the parameters (descriptive)
- index\_\*.xco script that generates the index files (descriptive)
- \*.shf help file in shf format (descriptive)
- \*db.dat data file

The Perl program `tdata2browse` for a given TDAT file generates all the necessary files for the EXOSAT DBMS. The `NewSubs` program allows you to regenerate, after customization, the descriptive files associated to a table.

### 4.1 `tdata2browse`

This program was developed with the purpose of taking TDAT files and turning them into tables for the EXOSAT DBMS. It also ingests the table in the EXOSAT DBMS if the appropriate flag is invoked.

The usage is the following:

```
tdata2browse [--no-conversion] [--no-pop] table_name
```

```
--no-pop:           The program will create all the necessary files, but it
                    will not actually ingest the data into the database
                    using db_pop. This is the default if only a TDAT header
                    file is found.
```

```
--no-conversion:   The program will assume the the coordinates are
                    already in equinox 1950 and the dates are already in
                    SHF format, so it is not necessary to convert these
                    types of fields.
```

The input to the program are the TDAT file, the header and .help file. These needs to be located at one of the following directories with the following naming conventions:

- /dba\_dbase/work/*tablename*/hdb/*tablename*.tdat



- `/dba_dbase/work/tablename/hdb/heasarc_tablename.tdat`
- `/dba_dbase/work/tablename/hdb/tablename.hdr`
- `/dba_dbase/work/tablename/hdb/heasarc_tablename.hdr`
- `./tablename.tdat`
- `./heasarc_tablename.tdat`

where *tablename* is the name of the database table. Only one pair of files is required (`.tdat` `.hdr`). These directory paths can be altered in the source code as described in the Installation section.

T2B expects the existence of either a TDAT (TDAT header and data) file or a TDAT header file. If only a header file is provided, T2B will produce all the output files described below, but will not have any data in the table and the indexes will not be created correctly. If T2B is supplied with a `*.help` file (see below for the format), the program will use that information to make the help documentation for the table. This is useful, but not required. If the `*.help` is not found the help will also be limited to whatever information is offered in the short descriptions in the `.hdr` file.

The `tdat2browse` (T2B) program uses a collection of Perl subroutines to produce the files necessary to ingest the table in the EXOSAT DBMS. These are:

- `tablename_par.dat`
- `tablename.info` file
- `index_tablename.xco`
- `tablename_cat.dat`
- `tablename.lsm`
- `tablename.shf`
- `db_pop_input.f`
- `tablename.load`

The `*_par.dat`, `index_*.xco`, `*_cat.dat`, `*.lsm`, `*.shf` are the descriptive files created in the form required by the EXOSAT DBMS system. The `*.info` file contains instead general information on the database, the description of the parameters and the help if provided. This file is used by the *NewSubs* program to create customized versions of all the descriptive files necessary to the EXOSAT DBMS.

The `db_pop_input.f` is a FORTRAN that translates the data into the `*db.dat` file. The coordinates and times in the EXOSAT DBMS must be imported as B1950 and as SHF (time since 1980) respectively. T2B evaluates the header information and determines if the time and the coordinates need to be transformed in the appropriate systems. Coordinates and times in the TDAT header are written in J2000 and MJD respectively therefore T2B transforms those in B1950 and SHF. The makefile is copied from the `/dba_dbase/work/create/templates` directory

and *db\_pop* is compiled using the *make* command. The FORTRAN code is compiled and checked for errors in compilation. If errors occur, the processing stops.

The \*.load file contains the responses to the *db\_pop* prompts for ingesting into the EXOSAT Browse DBMS (see later trouble-shooting problem). It assumes that the input file is either */dba\_dbase/work/tablename/tablename.dat* (if the *--no-conversion* option was specified) or */dba\_dbase/work/tablename/hdb/tablename.tdat*.

When all the files are created T2B starts the ingest procedure. It copies all the files in the appropriate directories of the */dbase/* area, creates an empty table, and attempts to execute the population code (*db\_pop*) with the *table\_name.load* file. If successful, it creates the indexes, makes the help, and informs the user of success. If the flag *-no-pop* is specified T2B does not initiate the ingest procedure. This can be done later manually the following command:

```
% createX2 -dc tablename
% ./db_pop < tablename.load
% createX2 -dr tablename
```

## 4.2 NewSubs

The *NewSubs* script allows to re-generate all descriptive files for the EXOSAT DBMS without the need to re-create the table. This is useful when there is the need to change the help text, the line summary display, or the parameters to use for the indexes after the table is created.

The input file to *NewSubs* is the .info file. Therefore any change to appear in the descriptive files needs to be done first in the .info file. The usage is the following :

```
NewSub [-a|-cilh] tablename
```

Flags:

- a: This is the default. It means you want to recreate all the dependent files.
- c: This option is used when you want to recreate the \*\_cat.dat file.
- i: This option is used when you want to recreate the index\_\*.xco file.
- l: This option is used when you want to recreate the \*.lsm file.
- h: This option is used when you want to recreate the \*.shf and hdb/\*.html files.

Note that the software overwrites any existing files in the directory. The software works in the current directory only, except in the following special case: if there is an *hdb/* directory in the current directory, the software will create .html files in that directory. The html file is not relevant for the EXOSAT DBMS but is the help format used at the HEASARC.

## 4.3 Format of the .help & .info Files

The .help and .info files are ASCII files where the information is provided with specific labels for section and parameters. The format for the label is the following:

*label => information about the label*

The information can span multiple lines and can be written directly after '`=>`' or starting from the following line. Information or listing that should appear exactly as typed has to be preceded by and followed by a full line of dashes starting from the beginning at the margin.

The `.help` file is a subsection of the `.info` file. The `.info` file also contains the following sections: general database information, the parameters displayed by Browse and the parameters to be indexed.

A line with the following string '`# HELP information`' segregates the help documentation from the other part of the info file. If it does not exist, the software interpret each line as being a new entry instead of allowing for multi-line values. It must be specified exactly as shown above.

The information to the left of the '`=>`' are case-sensitive. The label for parameter names must be all lowercase. In the help section, the label for non parameters with the same name as a parameter must include a `%` (with or without spaces) at the end of the key, for example, "References `% => ...`". Here is an example of `.info` file

```
#
# General Information
#
db description => Gamma-Ray Bursts from the IPN
w3b description => Gamma-Ray Bursts from the Interplanetary Network
search radius =>
default index => EVENT_NUMBER
observatory => GAMMA-RAY BURSTS
unique key => event_number
#
# Display and Index information
#
display => time|f='(a20)', num_sat|f='(i4,3x)', event_id, ulysses|f='(3x,a4)'
indexes => event_number, num_sat, event_id, comment, time, ulysses
#
# HELP Information
#
Overview =>
The Interplanetary Network (IPN) is a group of spacecrafts equipped
with gamma-ray burst detectors used to localize gamma-ray bursts (GRB).
Description % =>
This database table contains the time of the event and the number of
satellites that have detected the event.

#
# Parameter Help Information
# (Parameter names must be all lowercase.)
#
event_number => A sequential number associated to each events. This is not a
```

unique identifier of the burst number. I.e., further updates can have a different number associated to the same event. This should be regarded only as an index for the table.

```
#
date_event => The date of the event as day month and year (e.g. 11 NOV 90).
#
day_year => The day of the year of when the event occurred.
#
```

## 4.4 Examples

With the distribution are included tables that can be used to test the software. Here is a summary of what needs to be done step by step to ingest these examples into the EXOSAT Browse DBMS.

### 4.4.1 Ingest Examples That Use `make*.pl`

These instructions apply to the `ascagps`, `chandfslms`, and `pleiadxray` examples.

- Change directory to `/dba_dbase/work/tablename/`
- Execute `./make_tablename.pl`

The `make_tablename` script translate the original table into a TDAT file first and the following steps are to create the database table for the EXOSAT DBMS and to ingest the table.

- `mv tablename.info tablename.info.save`
- `tdat2browse -no-conversion -no-pop tablename`
- `mv tablename.info.save tablename.info`
- `NewSubs -a tablename` (Answer “y” to any question that NewSubs prompts you.)
- `./db_pop < tablename.load`
- `createX2 -dr tablename`

### 4.4.2 Ingest Swift Examples

These instructions apply to all Swift examples. The instructions for the WDS example are very similar to the Swift instructions. Please refer to the Readme file in the `wds` subdirectory. Please note that the WDS example requires the `GNUwget` and `md5` commands to be already installed in your environment. If these modules are not installed in your system, skip the WDS example or download these packages from the Internet and install them.

The tables here are assumed to be already in TDAT format, therefore the original reformatting as seen for the previous example is not necessary.

- Change directory to `/dba_dbase/work/tablename/`
- `mv tablename.info tablename.info.save`
- `tdat2browse -no-conversion -no-pop tablename`
- `mv tablename.info.save tablename.info`
- `NewSubs -a tablename` Answer “y” to any question that NewSubs prompts you.
- Copy `hdb/fake_tablename.tdat` to the incoming directory.
- `touch tablename.dat`
- `./update-tablename.pl`

The `touch` command creates an empty file, that is then populated by the `update-tablename.pl` script.

Once the process is working, it can be fully automated by changing the `$UpdateDB::dont_ingest` configuration variable to 0. Then, if any TDAT file is placed in the incoming directory, the update program will automatically ingest it without an operator .

#### 4.4.3 Generic Instructions for Ingesting Only TDAT to EXOSAT DBMS

If the TDAT file (header and data) is provided without the associated files (i.e., `.hdr` or `.info` files), the ingest procedure is possible following these instructions:

- `mkdir -p /dba_dbase/work/tablename/hdb/`
- `cp tablename.tdat /dba_dbase/work/tablename/hdb/`
- `cd /dba_dbase/work/tablename/`
- `tdat2browse tablename`

Note the difference with the first example. `tdat2browse` is invoked without flags therefore the script also ingests the table.

## 5 Script logs and Troubleshooting problems

### 5.1 tdat2browse

```
tdat2browse -no-conversion -no-pop swiftobs
Using file: /dba_dbase/work/swiftobs/hdb/heasarc_swiftobs.hdr...
Parameter hash made fine.
The help has been acquired.
Par file made without errors.
```

The info file has been made.  
The db\_pop\_input was made.  
The compile files copied fine.  
The compile occurred just fine.  
The \_cat made just fine.  
The index file made just fine.  
The lsm file made just fine.  
The shf file made just fine.  
The HTML file made just fine.  
The load file made just fine.

## 5.2 NewSubs

```
NewSubs -a pleiadxray
```

Parameter information acquired...

Info file parsed...

Help documentation acquired...

Acquired information corrected for defaults as necessary...

Creating output files as directed...

Cat file produced.

Index file produced.

Lsm file produced.

Shf file produced.

HTML file produced.

SPARE size check: SPARE=C\*8 /Spare Room

Would you like to do a cp\_done now? (y=yes,n=no) y

cp\_done results:

The copy of the tables to the live test area with flag pcih was successful.

Would you also like to create this table anew? (y=yes,n=no,r=recreate) y

Table created. No errors were reported.

You can check now...

### 5.3 createx2

This is a screen capture when *createx2* is run. The message *par\_c80 error: PARENT\_1 not found* is not an error and the database and the indexes are created

```
createX2 -dr pleiadxray
SQL 3.11 at 23:17:47 on 07-Oct-2002
SQL> decreate table pleiadxray
SQL> recreate table pleiadxray
There are 214 rows
Making catalogued VIEWS and INDEXes
  par_c80 error: PARENT_1                not found
SQL>
SQL> @/dbase/ss_files/index_zzdb
!SQL> CREATE INDEX * ON ZZDB(NAME)
!SQL> CREATE INDEX * ON ZZDB(OBSERVATORY)
!SQL> CREATE INDEX * ON ZZDB(DB DESCRIPTION)
!SQL>
!SQL> EXIT

SQL 3.11 at 23:17:47 on 07-Oct-2002
!SQL> create index * on total_pleiadxray(name)
!SQL> create index * on total_pleiadxray(ra)
!SQL> create index * on total_pleiadxray(dec)
!SQL> create index * on total_pleiadxray(lii)
!SQL> create index * on total_pleiadxray(bii)
!SQL> create index * on total_pleiadxray(vmag)
!SQL> create index * on total_pleiadxray(spect_type)
!SQL> create index * on total_pleiadxray(log_lx)
!SQL> create index * on total_pleiadxray(class)
!SQL>
!SQL> exit
```

### 5.4 db\_pop

When running *db\_pop* the following warning message will appear on the screen

```
./db_pop < pleiadxray.load
DB_POP 1.05: 07-Oct-2002 23:11:10 Address=browseadmin@somewhere.lab
(browseadmin,System)
readline: warning: rl_prep_terminal: cannot get terminal settings
Database name:
readline: warning: rl_prep_terminal: cannot get terminal settings
Next entry to be written is          2 enter new value if desired:
readline: warning: rl_prep_terminal: cannot get terminal settings
Enter INPUT file name:                216 records processed
```

Do not forget to create an index before reading the db

This is due to a recent change in readline and complains if parameters are passed from the command line. This is only a warning and the database file in the EXOSAT DBMS format is actually created correctly