HashFile is an object class that implements a bridge between the hashed database library functions, db(3) (used to implement the 'defaults' database and Digital Librarian index files), and the HashTable object. It can be viewed as an alternate interface to the database library functions or as a type of HashTable that survives program execution.

See **HashFile.wn** for details. The **HashFile** binary code is included as both a **HashFile.o** object file and as a **libHashFile.a** archive file (see **Makefile** for usage). To rebuild the object and archive files, just do '`make`'.

The example program, **HashExample**, can be built by doing '`make HashExample`'. It's database can be built by doing 'HashExample -b < /usr/dict/web2' (or any other word listsyou want to use). It reads up to 7-digit (telephone) numbers from stdin and searches for them in the database file **PhoneWords.{DL}**, printing the result, if any, to stdout, eg:

```
> echo '2277666 227 7666' | HashExample
2277666 barroom bassoon cassoon
227     bap bar bas cap car cas CBS ccs
 7666    Pomo poon room roon soon
>
```

The **PhoneWords** database file has integer, "i", keys representing telephone digits and string, "*", values (one to seven

characters) which are the equivalent words (from the **/usr/dict/web2** file) that can be dialed instead.   (Not every string that matches from **web2** is necessarily a useful mnemonic--nor even in **Webster**!)

Note that, of the nearly 2.4 million possible seven or less digit numbers ($8^7+8^6+...+8^1$ since zero and one don't have letter mappings), less than 2% have word mappings in **PhoneWords**, so you'll probably have to try a lot of numbers before you get a   match.   (Of the 2% of numbers that do have a match, over 15% have multiple matches).

Comments, suggestions and/or bug reports on the HashFile object and HashExample welcome.

- Christopher Lane (lane@sumex-aim.stanford.edu)