DATABASEFNS

DataBaseFns makes the construction and maintenance of MasterScope database essentially an automatic process.

DataBaseFns modifies the behavior of the Lisp functions MAKEFILE, LOAD and LOADFROM, such that writing out a file also updates and saves a MasterScope database, and loading the file also loads the database for you to use.

For example,

```
(LOAD ’FILE1)
```

loads FILE1, then looks for the corresponding database file FILE1.DATABASE. If the database exists, it is also loaded. The result is the same as if you had typed:

```
(LOAD ’FILE1)
.ANALYZE ALL ON FILE1
```

The database is maintained automatically for any file (containing functions) whose file name has the property DATABASE with value YES. Whenever such a file is dumped via MAKEFILE, MasterScope analyzes any new or changed functions on the file, and a database for all of the functions on the file is written on a separate file whose name is of the form FILE.DATABASE. Whenever a file that has a database property with value YES is loaded via LOAD or LOADFROM, then the corresponding database file, if any, is also loaded. The database is not dumped or loaded if the value of the DATABASE property for the file is NO. The DATABASE property is considered to be NO if the file is loaded with LDFLG=SYSLOAD.

If you change some of the functions defined in FILE1, and perhaps add new ones, then do:

```
(MAKEFILE ’FILE1)
```

Then FILE1 is written out. MasterScope analyzes all changed or new functions and writes FILE1.DATABASE, which contains MasterScope data for all functions on FILE1.

Requirements

MASTERSCOPE

Installation

Load MASTERCOPE.DFASL from the library, then load DATABASEFNS.LCOM.

User Interface

If DataBaseFns is loaded, the first LOAD of a file will ask if you want to load the corresponding database:

```
(LOAD ’FILE1)
Do you want to load the data base for FILE1?
```

And MAKEFILEing a new file will ask if you want to save the database:

```
(MAKEFILE 'FILE1)
Save the data base for FILE1?
```

Once you tell the system YES or NO for a particular file, it will remember and will not ask you again (until you load a new sysout).

Functions

You can set up default answers to these questions by means of the following variables:

LOADDBFLG [Variable]

This controls whether you are asked before the database file is loaded:

Yes Always try to load the database when a file is loaded.

No Never try to load the database when a file is loaded.

Ask (default) Ask whether to load the database when a file is loaded.

SAVEDBFLG [Variable]

This controls whether you are asked before the database file is saved:

Yes Always try to save the database when a file is saved.

No Never try to save the database when a file is saved.

Ask (default) Ask whether to save the database when a file is saved.

(DUMPDB *FILE PROPFLG*) [Function]

Dumps a database for *FILE* then sets the *DATABASE* property to YES, so that database maintenance for *FILE* will subsequently be automatic.

(LOADDB *FILE ASKFLG*) [Function]

Loads the file FILE.DATABASE if one exists. After the database is loaded, the DATABASE property for FILE is set to YES, so that maintenance is automatic thereafter.

Database files include the date and full file name of the file to which they correspond. LOADDB prints out a warning message if it loads a database that does not correspond to the in-core version of the file, and asks you if you approve.

Note: LOADDB is the only approved way of loading a database. Attempting to LOAD a database file will cause an error.

[This page intentionally left blank]