author: Michael Sannella
file: {Phylum}<LispUsers>IMNAME. (& .DCOM)
loads file: {Phylum}<LispUsers>HASH.DCOM
loads file: {Phylum}<LispUsers>IMTRAN.DCOM [loaded by IMNAME.UPDATE.HASHFILE]

****** The IMNAME Database Package *****

IM format is the text formatting language that the Interlisp Reference Manual is represented in. It is somewhat like TEX source code, in that there are keywords, and brackets are used to delimit text. However, IM format was specifically designed for representing the Interlisp Manual, so the "text objects" used are semantically meaningful objects within the manual, such as "function definition", "lisp code", "subsection".

IMNAME is a package designed to help people who may frequently need to modify IM format files. Usually, a large document (such as the Interlisp Manual) is stored in a number of seperate files, and it is difficult to know which file contains a particular piece of information. IMNAME contains functions for analyzing a set of IM format files and building a database of "IM Names" (functions, variables, property names, etc) with pointers to the files where they are defined. Using this database, other functions allow the user to specify an IM name, and bring up a Tedit window on the appropriate file, with the cursor positioned at the right place. This tool has been very helpful to the people updating the Interlisp Reference Manual.

****** Using IMNAME to access IM format files. *****

The normal way of using an IMNAME database is by creating an "IM name inspector". This can be done using either of the following two functions:

(MAKE.IM.INSPECTOR hashFileName)
or
(IMNAME hashFileName)

hashFileName should be the name of the IMNAME database hashfile to use for this inspector. One can create multiple IM name inspectors point to the same hashfile, or to different hashfiles. Currently, I know of only two IMNAME hashfiles; the one for the Interlisp Manual and one for the Loops Manual. If hashFileName is the atom INTERLISP or LOOPS, this Inspector will point to the appropriate hashfile. If hashFileName is ommitted, it will assume that the Interlisp Manual hashfile is desired for people whose default host is Phylum, and the Loops Manual for people on Ivy. (Other people can set up their own default IMNAME hashfile by setting the global variable IM.NAME.DEFAULT.HASHFILE).

MAKE.IM.INSPECTOR sets up an "IM Inspector Window", which contains a menu. Initially, this contains the single selection "Type an IM name", which (when buttoned) prompts the user to type a name which will be looked up in the database. If an IM name is found in the hashfile, another window will appear below the first one, containing all of the "types" that the given name is known by. For example, a name may be known as both a variable and a function. If one of these types is selected, a third menu will apear below the second, listing references to the name in different files. Selecting one of the references will move the cursor in a TEDIT window (if there exists an active TEDIT window to the appropriate file), or create a new TEDIT window to the file. [Note: If a name is only known to be of one type, the "type menu" step is ommitted.]

Other functions:

(INSPECT.IM imname hashFileName)
This allows the searching for a single IM name, using several pop-up menus. In general, it is easier to use the IM name inspector.

(GET.IM.NAME.LIST hashFileName)
Returns a list of all of the IM names known within the specified hashFileName.

****** Updating an IMNAME database. *****

IMNAME works by referencing a hashfile database containing IM names and pointers to files where these names are referenced. As a set of files is modified, this information will become obsolete. In particular, simply adding a few characters to a file will invalidate all pointers to positons later in the file. This can be tolorated for awhile (it is rare that IM format files will be changed enough such that the pointers are totally off), but eventually, it is necessary to update the IMNAME database, using the following function:

(IMNAME.UPDATE.HASHFILE oldHashFileName addFileList deleteFileList flushDisappearedFlg)

This function updates an IMNAME database hashfile. First, it looks at the list of files referenced in the hashfile, and determines which of these files have been updated, by comparing version numbers. Next, every updated file is reanalyzed with IMTRAN, and updated index information is stored into an in-core hasharray. Finally, the entries in the old hashfile are read in, merged with the new info, and written out to a new hashfile.

oldHashFileName is the name of the hashfile to be updated (Note that this file must be named explicitly --- no file searches are done so that the user will not inadvertantly start updating the main manual database). The new hashfile will be created as the new version of the same file name. addFileList is a list of files that will be analyzed, and added to the database. deleteFileList is a list of files that will be deleted from the database. addFileList and deleteFileList can be used to "manage" a database, as new files are added to a document, and old ones are removed, split up,or renamed. flushDisappearedFlg determines what IMNAME.UPDATE.HASHFILE will do if it finds that some of the files in the database have disappeared (and they are not named on deleteFileList). If flushDisappearedFlg = T, the info for those files will simply be deleted. If flushDisappearedFlg = ERROR, IMNAME.UPDATE.HASHFILE will return without doing anything if files have disappeared. If flushDisappearedFlg = <anything else>, the info on the disappeared files will simply be retained.

To create a new IMNAME hashfile, pass a non-existant file name as oldHashFileName, and give a list of files as addFileList. In this case, a new hashfile will be created just from the internal hasharray info.