Identification

rename, addname, delname
P. Smith

Purpose

The commands rename, addname and delname provide a way to change the list of entry names for an entry in the working directory.

Usage

rename (oldname1 newname1 oldname2 newname2 ...)

Rename replaces each oldname with the newname immediately following it in the argument list. Each pair of names is processed independently and from left to right. That is, any two pairs may refer to the same or different entries in the working directory. If an oldname is the name of a link entry, only the name of the link entry in the working directory is changed, not the name of the entry to which the link points.

By using the * convention, a user may change corresponding components of a group of entry names. For example, rename (alpha.* beta.*) causes the first component, alpha, of the entry names in the group alpha.* to be replaced by beta. The second component of each name remains unchanged.

If any newname is not unique in the directory, the user is asked if the existing segment with the same name should be deleted. Upon receiving a "no" answer no name change will occur and rename proceeds with the next pair of names. If the user types YES, rename deletes the existing segment and then performs the requested name change. The question may be stopped by the interjected command [no_ques], or by setting the no_questions option on. In this case rename proceeds to change the entry name, deleting an existing segment if necessary.

The user issuing the command rename must have the write attribute on in his working directory. The read attribute is also necessary if the * convention is used.

addname entryname (list)

Entryname is any name defining a segment in the working directory. List is a list of names, separated by blanks,
that are to become additional names for the segment \texttt{entryname}.

If any name in \texttt{list} is not unique in the directory, \texttt{addname} asks the user if the segment having the same name should be deleted. If the user replies YES, or if the no_questions option is on, the file is deleted and the name in \texttt{list} is added to the names of the file \texttt{entryname}. Upon receiving a negative reply, \texttt{addname} does not add the name to the list of names of \texttt{entryname} and proceeds to the next name in \texttt{list}.

The user must have the write attribute on in his working directory.

\texttt{delname \texttt{entryname} (list)}

\texttt{Delname} deletes the names in \texttt{list} from the names of the segment \texttt{entryname}. If \texttt{list} includes all the names of a segment, \texttt{delname} asks the user if the segment should be deleted. If the user replies YES, \texttt{delname} deletes the segment. No action is taken if the reply is negative. Should the no questions option be on, \texttt{delname} deletes the segment.

The user must have the write attribute on in his working directory.

\texttt{Implementation}

Implementation is included for \texttt{rename} only, since \texttt{addname} and \texttt{delname} will be coded at a later date.

The procedure \texttt{get_wdir (BX.8.12)} returns the path name of the working directory. This path name is needed for calls to Directory Supervisor primitives.

\texttt{Rename} handles each pair of names separately. If \texttt{oldname} does not contain the character *, the library routine \texttt{change_name} is called. \texttt{Change_name}, in turn, simply calls the Directory Supervisor primitive \texttt{cname} to change the name of the file \texttt{oldname} to \texttt{newname}.

If \texttt{oldname} does contain the character *, the routine \texttt{star (BX.2.02)} is used to obtain the names for all the segments in the directory that are included in \texttt{oldname}. \texttt{Rename} puts these names array of character strings, \texttt{oldnames}.

Another array of character strings, \texttt{newnames} is constructed using the argument \texttt{newname} and the array \texttt{oldnames}. The library routine \texttt{change_name} is then called successively.
with a name from `oldnames` and the corresponding name from `newnames` until the names of all the segments in `oldnames` have been changed.

If `change_name` signals the error "entryname not unique in directory", `rename` asks the user if the existing segment with the same name should be deleted. If the user replies YES, `rename` deletes the segment and calls `change_name` again. If the user replies NO, `rename` skips that entry name and proceeds with the next pair in the argument list. If the no_questions option is on, `rename` deletes the segment and calls `change_name`. 