

Published: 12/08/66
(Supersedes: BX.8.04, 01/14/65)

Identification

delete
P. Smith, E. Bjorkman

Purpose

delete (entries)

Delete deletes each entry defined by the list (entries). The user must have the write attribute on for the directory containing (entries) as well as for each branch entry in the list.

If an entry in (entries) is a link, the entry is deleted. The segment pointed to by the link is not touched.

If an entry in (entries) is a directory, all segments listed in that directory are also deleted, providing the user has the write attribute on for the directory as well as for all the segments inferior to it. If the user does not have the write attribute on for all the necessary entries, delete deletes as many entries as possible. No directory may be deleted until all segments pointed to by branches in that directory are deleted.

When an entry in (entries) is a directory delete informs the user with the following comment:

"Entry is a directory. Do you want entry and all the segments in it deleted?"

Delete then acts according to the "yes" or "no" answer typed by the user. The above question may be stopped by the interjected command [no_ques], or by turning on the no_questions option.

When there is no console attached to a process, or no_questions in on, delete proceeds to delete the directory.

Comment

A message is printed when an entry cannot be deleted. If delete is called from a procedure and an entry cannot be deleted an error is signalled.

Implementationdelete (entries)

For each entry in the list (entries) a separate call must be made to the Directory Supervisor.

An entry is well defined to the file system by a path name which points to a directory and a name of an entry contained in this directory. A symbolic entry name in the list entries is converted to the strings dir and name by the library routine entryarg (BY.2.04) which locates this entry in the file system hierarchy.

If any component of the name name is denoted by `*`, the library routine star is called to obtain the entry names in dir defined by name. Each name returned by star must be treated separately and is stored in the string, name.

An entry is deleted from dir by the library routine delete_entry (BY.2.01) which in turn simply calls Directory Supervisor primitive delentry.

delete_entry can signal the following errors:

1. Write attribute not on in the entry pointing to dir. Name then cannot be deleted.
2. Write attribute not on for the entry name. Name cannot be deleted.
3. Name points to a directory. Delete sends the following message to the user.

"Entry is a directory. Do you want entry and all the segments in it deleted?"

If the user types "no", delete processes the next entry in (entries) or returns to command status if there are no more entries to process. If the user types "yes" or there is no console associated with the process calling delete, then the file system library routine donode (see BY.2.04) is called as follows:

```
donode (name, ln, d, 'go_away', 1)
```

Donode executes the routine go_away in the directory name and in all directories ln or fewer levels inferior to name. (See BY.2.04) ln is set to a large number to insure that all directories inferior to name are reached. Go_away

is a small piece of coding which calls delete-entry. Specifically, go_away handles error signals from delete-entry as follows:

1. Write attribute is not on in entry pointing to dir. Dir and its contents cannot be deleted. A comment is sent to the user. A normal return is made to donode which then proceeds with the next directory.
2. Write attribute not on for an entry in dir. The entry cannot be deleted. Go_away proceeds with the next entry in dir.
3. Entry to be deleted points to a directory. Since donode is working from the bottom of the hierarchy to the top, this directory has had as many of its entries deleted as possible. If the hierarchy contains no entries, this error return should not occur. Therefore, go_away assumes that it cannot be deleted because it is not empty. A comment is sent to the user that this directory may not be deleted. Go_away proceeds with the next entry.

Upon return from donode, delete makes another call to delete_entry to delete the entry name. If the directory name is still undeletable, delete will comment and proceed with the next entry in (entries).