

Published: 08/19/68  
(Supersedes: BX.8.03, 04/07/67)

### Identification

listacl

C. A. Cushing, C. Marceau, K. Martin

### Purpose

The user may wish to check access control information for segments in his own or in other users' directories. He can, using listacl, print access control information for a particular segment or for all segments in a directory. The user of listacl should be familiar with access control conventions as described in BG.9.00 and BX.8.00. BX.8.02 discusses commands for setting access control information.

### Usage

```
listacl entry acname1 -- acnamen
```

listacl lists information from the common access control list (CACL) of the directory containing entry, and information from the access control list (ACL) of each segment in the set entry. The CACL of a directory contains the access control information that applies to every entry in the directory.

Entry is the pathname, relative to the working directory or to the root directory, of a segment or a set of segments specified using the \* convention. If entry is null, only information on the CACL of the working directory is listed. (Entry may be omitted entirely only if acname1 -- acnamen are omitted.)

Acname1 -- acnamen are the three-element access control names (personal name, project ID, instance ID). If acname1 -- acnamen are omitted, listacl prints the access control information for all access control names in the CACL of the directory containing entry and in the ACL of the entry.

### Comments

The user issuing the command listacl must have the read attribute on in the directory defined by entry. If entry defines a link, it is necessary for the caller to have

the execute attribute on for the directory containing the link and all directories containing intermediate links; the read attribute must be on for the directory containing the non-link entry to which this link eventually points. An error message will be printed if access control information cannot be listed for an entry in the set entry.

To list access control information from the CACL of a directory b with pathname a>b, the user specifies as entry argument "a>b>". The final ">" signifies that information from the CACL of b is to be listed, rather than from the ACL of the branch which points to b. I.e., the user wants access control information about the segments in b, not about b itself.

### Errors

There are two types of errors which the user may make in using listacl:

- 1) The user may not have the requisite access privileges for reading access control lists. For example, he attempts to read the ACL of segment b in directory a, but does not have the Read attribute on in directory a.
- 2) The user supplies incorrect arguments to listacl. For example, he specifies entry "a>b>", but b is not a directory, hence contains no CACL.

### Implementation

The ACL for a segment (as well as the CACL for a directory) consists of a list of access control names, with access control information for each name, as discussed in sections BX.8.00 and BX.8.02. An ACL is represented by the following structure:

```

dc1 1 acl (aclct) ct1 (aclp),
    2 acname,
    3 name char (24),
    3 projid char (24),
    3 instance_id char (2),

```

2 mode bit (5),

2 plistp ptr,

2 trapp ptr;

acl - an array of access control names with associated information. The number of names is aclct.

acname - an access control name, consisting of three elements: a personal name, a project ID (projid) and an instance ID.

mode - five bits, representing TREWA (the attributes Trap, Read, Execute, Write, and Append). If a bit is 1, the corresponding attribute is on.

plistp - pointer to an adjustable bit string containing the protection list.

trapp - pointer to an adjustable character string which represents a trap procedure with arguments.

Listacl weights each of the names in acname1 -- acnamen (or the user's name if they are omitted) through a call to the external function weighter described in section BX.8.02). Each name is entered into the following array:

dc1 1 namelist (n),

2 name char (24), /\* personal name of user \*/

2 projid char (24), /\* project ID \*/

2 instance\_id char (2),

2 weight fixed bin (17); /\* weight attached to access control name \*/

Here  $n$  is the number of names in acname1 -- acnamen. The external function orderusers (see BX.8.02) orders namelist so that namelist(1) contains the "heaviest" access control name and namelist(n) the "lightest" access control name.

Next listacl considers the argument entry, a path name relative to the working directory or to the root directory. Listacl calls the entryarg procedure (see BX.8.05) which analyzes entry and returns:

- 1) the path name (relative to the root directory) of the directory which contains the entry or set of entries defined by the symbolic entry name,
- 2) the segment name (not a path name).

If the segment name is null (i.e. the argument entry ends in ">") listacl lists only information from the CACL of entry. Otherwise, listacl obtains the CACL of the directory containing entry by calling readacl (see above) which contains the common access control list for the directory. Listacl now extracts and prints at the user's console the CACL access information for the names in acname1 -- acnamen.

Next listacl goes on to the set of entries in the directory which were defined by entry. If the entry name returned by entry\_arg has \* or \*\* as a component (signifying a set of segments in the directory) then the entries it defines are obtained by calling the file system library procedure star (see BY.2.08).

For each entry returned by star or list\_dir, listacl calls readacl to obtain the ACL for that entry. (Readacl supplies a CACL if a null (zero-length) entry name is supplied to it, an ACL if a segment name is given to it as the second argument.) Listacl now extracts the access information for the names in acname1 -- acnamen from the ACL of the entry or, if they are not found there, from the CACL of the directory containing the entry. Listacl prints this at the console.