To: Distribution
From: Andrew M. Kobziar

Subject: Initial Access Control List Manipulation Entries.

The entries in the gate "hcs_" for Initial ACLs are similar to the 18-0 ACL entries except that one more argument, "ring", is added just preceding the last argument "code." Two Initial ACLs exist for each ring, one for new segments and the other for new directories.
Initial ACL Primitives

Generic arguments:

1. dir
   is the superior directory containing the directory whose Initial ACL is being referenced. (Input)

2. entry
   is the name of the directory whose Initial ACL is being referenced. (Input)

3. area_ptr
   points to an area into which ACL data is to be allocated. (Input)

4. area_ret_ptr
   points to the returned ACL which was allocated in the above area. (Output)

5. acl_ptr
   points to one of the ACL structures described below. (Input)

6. acl_count
   is the number of entries in the ACL structure pointed to by acl_ptr if acl_ptr is not null. (Input) Otherwise it is set to the number of ACL entries in the structure pointed to by area_ret_ptr. (Output)

7. daemon_switch
   if this argument is "O"b, a *. Sysdaemon.* rw or sma entry will be added to the newly created Initial ACL. (Input)

8. ring
   is the validation number of the ring to which the Initial ACL entries apply.

9. code
   is a standard status code. (Output)

Generic structures:

dcl 1 segment_acl (acl_count) aligned based (acl_ptr),
   2 access_name char(32),
   2 modes bit (36),
   2 zero_pad bit (36),
   2 status_code fixed bin (35);

This structure is accepted by the entries hcs$_add_inacl_entries and hcs$_replace_inacl and is returned by the entry hcs$_list_inacl_entries.
1. **access_name**
   is the name which identifies the processes to which this Initial ACL entry applies.

2. **modes**
   contain the modes for this access name. The first three bits correspond to the modes read, execute, and write. The remaining bits must be zero.

3. **zero_pad**
   must contain zero. (This field for use with extended access.)

4. **status_code**
   is a standard status code for this Initial ACL entry only.

```
dcl 1 dir_acl (acl_count) aligned based (acl_ptr),
    2 access_name char(32),
    2 dir_modes bit (36),
    2 status_code fixed bin (35);
```

This structure is accepted by the entries `hcs_$add_dir_inacl_entries` and `hcs_$replace_dir_inacl` and is returned by the entry `hcs_$list_dir_inacl`.

1. **access_name**
   as above

2. **dir_modes**
   contains the directory modes for this access name. The first three bits correspond to the modes status, modify, and append. The remaining bits must be zero.

3. **status_code**
   as above.

```
dcl 1 delete_acl (acl_count) aligned based (acl_ptr),
    2 access_name char(32),
    2 status_code fixed bin (35);
```

This structure is accepted by the entries `hcs_$delete_inacl_entries` and `hcs_$delete_dir_inacl_entries`.

**Primitives:**

```
hcs_$add_inacl_entries (dir, entry, acl_ptr, acl_count, ring, code.);
```

```
hcs_$add_dir_inacl_entries (dir, entry acl_ptr, acl_count, ring, code.);
```
The above primitives add (replace if an entry with the same access name already exists) Initial ACL entries specified in the ACL structure pointed by acl_ptr to the Initial ACL in the specified directory.

hcs$_replace_inacl (dir, entry, acl_ptr, acl_count, daemon_switch, ring, code. );

hcs$_replace_dir_inacl (dir, entry, acl_ptr, acl_count, daemon_switch, code. );

The above primitives replace the Initial ACL in the specified directory with the ACL specified in the structure pointed to by acl_ptr.

hcs$_list_inacl (dir, entry, area_ptr, area_ret_ptr, acl_ptr, acl_count, ring, code. );

hcs$_list_dir_inacl, dir, entry, area_ptr, area_ret_ptr, acl_ptr, acl_count, ring, code. );

The above primitives return the contents of the Initial ACL of the specified directory. The information is allocated in the area pointed to by area_ptr and area_ret_ptr is set to point to the ACL structure. If area_ptr is null then acl_ptr is assumed to point to an ACL structure into which mode information is to be placed for the access names specified.

hcs$_delete_inacl_entries (dir, entry, acl_ptr, acl_count, ring, code. );

hcs$_delete_dir_inacl_entries (dir, entry, acl_ptr, acl_count, ring, code. );

The above primitives will cause the ACL entries specified in the structure pointed to be acl_ptr to be deleted from the Initial ACL in the specified directory.

In order to modify an Initial ACL, one must have modify access to the directory, and ring must be greater than or equal to the process's validation level. In order to list an Initial ACL one must have status permission to the directory.