Published: 05/26/67

## Identification

User Identification Data Bases
C. Marceau

## Purpose

When a user logs in, the procedure which identifies him (user_who, see BQ.2.03) must have access to certain information about the person who is logging in, the project he wishes to work on, and the console from which he is logging in. For example, the procedure must ascertain that the person logging in is really the person he claims to be (he must give a secret password), that he may work on the project he names, and that he is allowed to use the console from which he is attempting to log in.

This section contains an overview of the directory structure containing validation information about persons, projects, and consoles. Section BQ.4.01 discusses the dedicated console list, a list which records the association of certain consoles with certain users. Section BQ.4.02 discusses the personnel list, a list of all persons who are users at the installation. Section BQ.4.03 discusses the user profile of a user, a collection of information about the user consulted by system procedures.

## Discussion

The reader of these sections should be familiar with section BQ.2.03 on the User Control Process, which discusses the user_who procedure and the method it follows to identify the user logging in.

To summarize briefly, user_who first ascertains the id of the console from which the user is attempting to log in. It checks the console id against those in the dedicated console list, and if the id appears on the list, logs in a specified user, whose name and project id are associated with the console id on the dedicated console list. (See below, dedicated console list.)

If the console id does not appear in the dedicated console list, user_who must identify the user (person and project id). User_who asks at the console for the user's personal password and checks it against a password recorded in the personnel list (see BQ.4.02). If the user specified

no project id, user_who also looks up his default project id in the personnel list. Next user_who looks up the user in the project directory (see below) of the project which he specified. If a profile for the user appears in the project directory then the user is allowed to work on that project.

Access control

Access to the directories and segments discussed in this section is controlled so that unauthorized users cannot tamper (either inadvertently or deliberately) with user identification information. The file system access control mechanism is described in section BX.8.00; a brief summary of salient points is included here for the reader's convenience. For more detail, consult section BX.8.00 and BG.9.00.

Briefly, it is possible to define the mode of access of a particular user (or even a particular user-process-group) to a directory or segment. It is further possible to limit that user's access so that he may only get at the directory or segment from a particular protection ring. Three protection rings concern us here: the hard core ring, which contains the access control mechanism and other equally sensitive modules; the administrative ring, which contains administrative modules and data bases; the user base ring, which is unprotected user area. (A more complete description of the purpose and philosophy of protection rings may be found in BD.9.00.) Note that the ring in which a segment or directory resides depends on the user who is trying to access it; i.e. user A may be able to access a directory from the user ring, while user B can only get that directory from the hard core ring.

There are five usage attributes which define a user's access to a segment or directory. The interpretation of these usage attributes differs for directory and non-directory segments. Only directories are of interest here. If the user has the Read attribute on for a directory, he may "read" the directory to get information about any or all of the entries in the directory, including access control information for the entries. If the user has the Execute attribute on, he can search the directory for specifically named entries in order to use them, but cannot get access control information about the entries (unless he also has the Read attribute on). If the user has the Write attribute on for a directory, he can delete

or rename specifically named entries and change access control information for specifically named entries. If the user has the Append attribute on for a directory, he can add entries to the directory but cannot rename or delete any existing entries, nor modify their access control information (unless he also has the Write attribute on). Finally, if the user has the Trap attribute on, it causes execution of a specified trap procedure before the user can access the directory.

Dedicated Console List

The dedicated console list associates with each console id on the list the name and project id of the user, who may be either

a) a very special user who is known to be the only user who has access to this console (e.g. the console is in a vault in his private office and only he has access to the console);

b) an unknowable person working on a project which allows him only restricted access to the system (e.g. a student in a large class which uses the console for a short time, or a random troll using a pay console - like a pay phone).

Section BQ.3.01 explains how the unknowable person is restricted in his use of the system. Section BQ.4.02 discusses the dedicated console list itself in some detail. Here our concern is with the place of the dedicated console list in the system skeleton.

The dedicated_console_list segment is immediately inferior to the login directory (login_dir), which is immediately inferior to the root directory. Its path name, thus, is

(root) > login_dir > dedicated_console_list

(Figure 1 shows the section of system skeleton containing all segments and directories discussed in this section.)

Login Directory

The login directory is a system-wide data base containing (at present) two entries:

1) the dedicated console list (see above),
2) the personnel list (see below).

Only the system administrator has access to read or modify the login directory, for example, to add a new segment to the directory or to see what segments are in the directory.

The system process which logs in the user (the User Control Process) may search the login directory from the administrative ring (execute attribute). No other users may access the directory. (Fig. 2 is a chart showing access control to the directories discussed in this section.)

Personnel List

The personnel list is a directory immediately inferior to the login directory. It is used to identify persons logging in and is not an administrative data base containing addresses and social security numbers. Such personnel records are kept elsewhere in the system. The personnel list contains one entry for each person (personal name) known to the installation. A person is known to the installation by virtue of having an entry which bears his name in the personnel list directory. A personal name is at most 24 characters in length. In the segment for each person is the personal password of the person (if any), a default project id, and a list of proxies (names of persons who may log in for this person). The password is used to identify the person when he logs in and, if he types no project id when logging in, his default project id is assumed. A proxy may log in for the person, giving his own password, but thereafter is identified as the user for whom he proxies. The personnel list is discussed in detail in BQ.4.02. Here we are interested in its place in the system skeleton: the personnel list is a directory named "personnel_list" and has path name

(root) > login_dir > personnel_list

The personnel list directory is accessible to the User Control Process for searching the administrative ring (execute attribute). All persons in the system have the execute attribute on for the directory in the hard core ring, so that each person may locate his segment in the directory. The system administrator has Read, Write, Execute and Append attributes for the directory from the administrative ring.

Project Directory

For each project at an installation there is one directory, which contains one user profile directory for each person working on the project. User profile directories are discussed in section BQ.4.02. Here it is only necessary to note that each user profile has as its name the personal name of the user, as the user gives it when logging in (e.g. John_Doe). Note that a person is a user on project A by virtue of having a user profile in project directory A.

Each project directory has as its name the id of the project. A project id is at most 24 characters long. All project directories are located in the project directory directory, which is immediately inferior to the root directory. The project directory for project "T234", for example, has path name

(root) > project_dir_dir > T234

and user John_Doe who works on T234 has user profile

(root) > project_dir_dir > T234 > John_Doe

The project directory directory can be read and written only by the system administrator, since adding or deleting a project directory is equivalent to adding or deleting a project to the system. The administrator can access the project directory directory only from the administrative ring; this restriction serves to discourage him from unintentionally modifying the directory.

All project administrators have the execute attribute on in the project directory directory so that each project administrator can locate the project directory for his project. (Remember that to "execute" in a directory means to locate a specified segment.) Similarly, a project directory can be read and written only by the administrator of the project and then only in the administrative ring, since to add or delete a user profile from the directory is equivalent to adding or deleting a user from the project. The process which logs in the user must be able to search in every project directory so that it can determine whether, for example, John_Doe is a user on project T234 (i.e., whether directory T234 contains an entry named "John_Doe", John Doe's user profile).

In its first implementation, Multics does not include the concept of subproject. However the current system can easily be expanded to include the case of a project which is under the control of another project. The subproject

may have its own administrator or may be administered by the leader of the "superproject"; this will be reflected in the access control to the subproject's directory. The subproject's directory is immediately inferior to the "superproject's" directory. Some obvious consequences of this method of implementation are:

1) user_who (and other system procedures) must be able to distinguish between subproject directories and user profile directories;

2) no subproject of a project may have the same name as a user of that project;

3) it must be possible to locate the subproject directory without searching through all project and subproject directories: viz, user_who must recognize hierarchical project names. The system administrator might in addition create a link from the project directory directory to subproject directories, so that users would not always be forced to use the hierarchical project names.

Figure 3 shows a hierarchy of subproject directories (subproject directories are not included in the access control chart in figure 2 because access control of these directories depends on installation or project conventions concerning subprojects).

User Profile

The user profile of a user is a directory containing segments which are of interest to the Multics system. The term "user profile" is also used to refer to the set of segments in that directory. The user profile directory is not controlled by the user himself, but by certain system personnel who determine the number and names of, and the access to, segments in all user profile directories. These segments contain information about the user, some of it supplied by the system administrator, some by the project administrator, and some by the user himself.

Section BQ.4.03 contains a list of segments in the user profile directory. Here three segments are mentioned to give a flavor of the nature of the user profile:

a) permanent options list (perm_op_list) - a segment which records permanent settings of user options. Options (described in BX.12.00) offer a means whereby the user can exert some measure of control over the actions of

both system and user programs. The user himself sets options and can add new options to the permanent options list. Other users, including the project and system administrators, have no access to this segment.

b) enforced searching advice - a segment, which, if present, indicates that the project administrator of the user enforces searching advice on the user. The segment contains the rules which the Search Module (see BD.4.00) will use in searching for segments during dynamic linking. The user may not write in this segment; only the project administrator has the write attribute on for this segment.

c) project subsystem - this segment contains information about a subsystem which the user's project administrator enforces as either a mandatory or a default subsystem. If the subsystem is specified as "default" the user may specify his preference (if any) in another segment of the profile.

The concept of the project subsystem is discussed in BQ.3.01. Briefly, when a user logs in, he is "hooked up" to a particular subsystem, usually the Multics command system (see BX.0.00). However, the user's project administrator may enforce an alternative subsystem on the user, by modifying the "project subsystem" segment in the user's profile directory (See BQ.4.03).

The subsystem mechanism may also be used to provide a "password" mechanism on the project level; in this case, after execution of a "password" procedure, the user is hooked up to the Multics command subsystem. See BQ.3.01 for the implementation of subsystems on the project level.

Some of the segments in the User Profile directory are used often by user processes. These segments are copied into the process directory of user processes and are known as the process profile. Normally these segments are copied only once, when the user logs in, and as processes are created in the group their process directories are linked to the group-wide process profile.

Figure 1: Portion of the File System Hierarchy Showing User Identification Data Bases

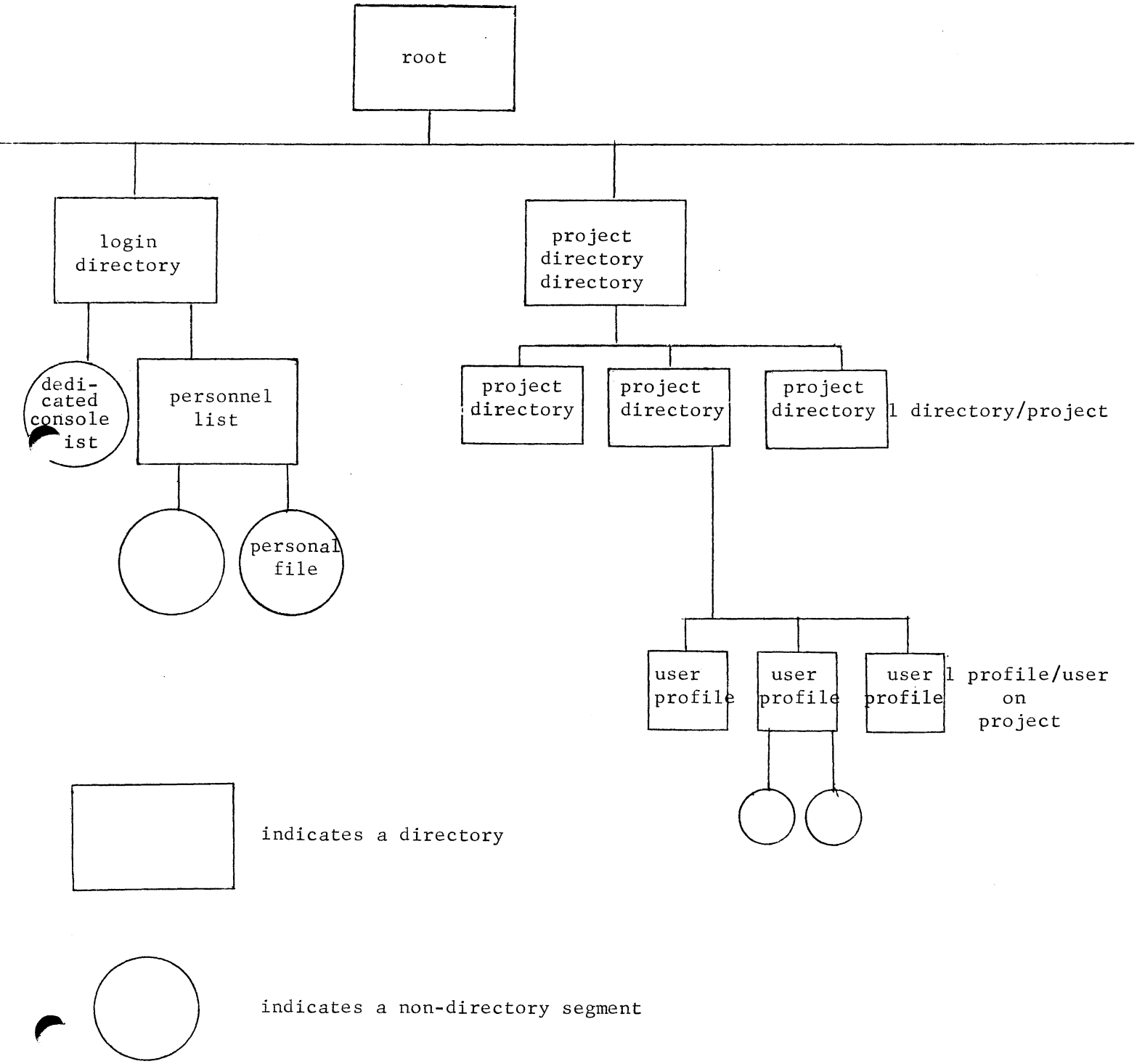

Figure 2 Summary of access control for user identification data bases (directories)

| directory | user* | mode ** | protection ring ≠ |
|---|---|---|---|
| login_dir | system administrator.system | REWA | administrative ring |
| | .User_Control ⇕ | E | user ring |
| project_dir_dir | system administrator.system | REWA | administrative ring |
| | administrator of (project)A. | E | administrative ring |
| | Project A | | |
| (project directory) "A" | administrator of A. project A | REWA | administrative ring |
| | *.user_profiles⇕ | E | administrative ring |
| | .User_Control ⇕ | RE | user ring |
| user profile of | A.A | E | user ring |
| user "X.A" | administrator of A.A | E | administrative ring |
| | *.user_profiles⇕ | REWA | administrative ring |
| personnel_list | system administrator.system | REWA | administrative ring |
| | .User_Control ⇕ | E | administrative ring |
| | *.* | E | hard core ring |

* a user is a person working on a project. Notation used here is "X.A" stands for person X working on project A.

** mode of access, where R,E,W,A stand for Read, Execute, Write, and Append attributes respectively.

≠ the protection ring defines the ring from which the user has access (defined by mode) to the directory.

⇕ the User Control Process which logs in the user is a system process and is represented here by .User_Control". User Profile directories can only be modified by certain system programmers who work on the project "user_profiles".

Figure 3 A Hierarchy of Projects