Maintaining Users [7]

UNICOS user account information is stored in a user database (UDB). This chapter describes the following topics:

- Brief descriptions of the UDB and the `/etc/xadmin`, `/etc/nu`, and `/etc/udbgen` utilities
- A brief summary of the procedure for adding a user record to the UDB
- Principal UDB files and commands
- Creating a user login
- Modifying user login information in the UDB
- Deleting a user record
- Maintaining user environment files
- Transferring user records to another file system

For information on ways to communicate with users, see Chapter 8, page 217.

7.1 Related user accounts documentation

The following documentation contains detailed information covered in this section and additional information about the UDB:

- *UNICOS User Commands Reference Manual*, Cray Research publication SR–2011: `chgrp`(1), `chown`(1), `passwd`(1), `su`(1), and `udbsee`(1) man pages
- *UNICOS Administrator Commands Reference Manual*, Cray Research publication SR–2022: `nu`(8), `udbgen`(8), and `udbpl`(8) man pages
- *UNICOS File Formats and Special Files Reference Manual*, Cray Research publication SR–2014: `acid`(5), `cshrc`(5), `group`(5), `passwd`(5), `profile`(5), `shells`(5), and `udb`(5) man pages
- *General UNICOS System Administration*, publication SG-2301, the chapter on UDB

7.2 The user database (UDB)

The user database (UDB), which is unique to the UNICOS system, contains entries for each user who is allowed to log in and to run jobs on your system. The UNICOS system maintains encrypted login passwords in the UDB, rather than in a separate password file. However, the traditional UNIX `/etc/passwd` and `/etc/group` files are still supported; when the UDB is updated, they are updated automatically.

The only way that the UNICOS system can identify an individual user is by that person's user ID. The system maps the user ID to your user record in the UDB. The system administrator assigns this unique user ID number. The user ID is also a field in the `/etc/passwd` file.

You can modify the UDB in the following ways:

- If you have access to a windowing environment, you can use the `/etc/xadmin` command, which provides a graphical user interface (GUI) for managing user login accounts. This command has all the functionality of the `/etc/nu` utility. This X Window System based interface is self-explanatory and requires no prior knowledge of the `nu` command. It contains a tutorial for an overview of the command and context-sensitive help on specific topics. `xadmin` uses the UNICOS message system to generate its error and help messages. For more information, see the `xadmin`(8) man page.

- If you do not have a windowing environment, you can use the `/etc/nu` utility (see Section 7.5, page 183).

  The `nu` utility is a full-screen, prompt-driven utility that prompts you for the user information that you want to create or modify (for example, login ID, password, and name). The `nu` utility then creates or otherwise modifies the appropriate directories, makes entries in a log file, or (for updates) merges the changes into the `/etc/udb` file. If you have configured the `nu` utility to skip prompting for specific UDB fields, you must use `udbgen` to access these fields.

- You also can use the `/etc/udbgen` utility (see Section 7.6, page 199). The `udbgen` utility is actually the program underlying the `/etc/nu` utility. You can access this underlying utility directly by issuing the `udbgen` utility and its associated directives. The `udbgen` utility does not prompt you for user information. Although using `udbgen` to update the UDB involves a more complicated syntax than `nu`, it can give you more control over the update process. The `udbgen` utility also can enable you to perform batch updates and to update many user accounts at one time. If you have configured the

nu utility to skip prompting for specific UDB fields, you must use udbgen to access these fields.

7.3 Adding user records to the UDB

The following is a summary of the procedures that you should use to add a user record to the UDB (etc/udb):

- Learn about the UDB fields and decide which values to assign to the UDB fields (more than 80 fields exist). The following section describes a suggested subset of UDB fields (see Section 7.4, page 173). For a full listing and explanation of all possible fields in the UDB, see the udbgen(8) man page. Some of the values you select will affect other factors on the system (for example, the login directory field determines in which file system the user is placed). You must make sure sufficient disk space is available to meet the user's needs in this file system.
- If the user will be placed in a new group that you will reference by name, add the new entry in /etc/group (see Procedure 19, page 181).
- If the user will be placed in a new account group that you will reference by name, add the new entry in /etc/acid (see Procedure 20, page 182).

Then, if you are using /etc/nu, do the following action:

- Follow the procedures in section Section 7.5, page 183, to make the new entry in /etc/udb by using the /etc/nu -a command. (Section 7.5, page 183 also includes procedures for modifying and deleting user records.)

Or, if you are using /etc/udbgen, do the following action:

- Follow the procedures in Section 7.6, page 199, to make the desired entry in /etc/udb by using the /etc/udbgen command. (Section 7.6, page 199 also includes procedures for modifying and deleting user records.)

To help determine when you would use the /etc/nu and /etc/udbgen utilities, see Section 7.2, page 172.

7.4 UDB files and commands

The following are the principal files related to the UDB:

File	Description
`/etc/acid`	Account name/ID map file for accounting billing group.
`/etc/group`	Group name/ID map and membership file. This file is common to all UNIX environments.
`/etc/nu.cf60`	The `nu` utility configuration file.
`/etc/passwd`	UNIX password file used for compatibility with existing commands. The * symbol replaced the encrypted password field. This file is common to all UNIX environments. Encrypted passwords are stored in the `/etc/udb` file.
`/etc/udb`	Primary user database file; binary file. It stores the user's password.
`/etc/udb.public`	Public version of `/etc/udb` with read permission for "world." All sensitive information has been set to 0.
`/etc/udb_2/udb.index`	Public index file with read permission for "world." All user (UIDs) and user names found in the UDB extension files along with their associated `udb_priva` and `udb_pubva` record offsets will appear in this file.
`/etc/udb_2/udb.pubva`	Public file with read permission for "world." New fields that would have been publicly accessible had they been added to

	`/etc/udb.public` will appear in this file.
`/etc/udb_2/udb.priva`	Private file that can be read only by privileged users. The same rules that prevent certain information from appearing in `udb.public` are applied to new fields appearing in this file.

Note: The following scripts are **not**, as released, intended to be used as is; they are only examples that you must modify for your specific site requirements.

Script	Description
`/etc/nulib/nu1.sh`	The `nu` utility uses this script to create a user directory and to change the permissions on this directory.
`/etc/nulib/nu2.sh`	The `nu` utility uses this script to initialize the contents of the user's directory.
`/etc/nulib/nu3.sh`	The `nu` utility uses this script to purge a login account.
`/etc/nulib/nu4.sh`	The `nu` utility uses this script to purge a login without removing the account from the UDB. This action is performed to preserve accounting information.

The following are the principal commands related to the UDB:

Command	Description
`/etc/xadmin`	Graphical user interface that has all of the functionality of the `/etc/nu` command.
/etc/nu	Adds, deletes, and modifies login records. `/etc/nu` uses the following scripts:
`/bin/passwd`	Creates or changes a user's password
`/bin/udbsee`	Converts information from the user database into an ASCII format
`/etc/udbgen`	Generates and maintains the user database

`/etc/udbpl`	Writes to `stdout` administrative information for designated users

The remainder of this section includes information and procedures about using the `/etc/nu` and `/etc/udbgen` utilities to maintain your user records.

Procedure 18: Determining settings for UDB fields

The UDB (`/etc/udb`) contains information for each user who is allowed to log in and run jobs on your system. The UDB also contains many other fields that are specific to the UNICOS environment. Fields that you can specify for each user include settings that specify limits for batch processing, interactive processing, account security, the data migration facility, CPU access, disk quotas, the fair-share scheduler, and many others. You must provide the appropriate settings for the fields and resource limits in the UDB for each user record.

For a full listing and explanation of all possible fields in the UDB, see the `udbgen`(8) man page, which includes several examples.

> **Note:** The following UDB fields are a suggested minimum subset of the UDB fields that you should define for each user. The " `keyword:` *value* `:` " syntax of each entry that follows reflects the format accepted by the UDB if you use the `/etc/udbgen` utility; however, when using the `/etc/nu` utility, you do not use this format (see Section 7.5, page 183).

Basic user account definition fields

You should define the following UDB fields for each user (for all possible fields, see the `udbgen`(8) man page).

Note: The global default table contains entries for some of the UDB fields; for a list of these fields, see the `udbgen`(8) man page. The release defaults are applied by `udbgen` when it updates a UDB that has a default table that contains all zeros. To create a default table in an existing UDB, execute the `udbgen -c'#'` command. This command is an empty modification request, but it causes the default table to be created with the released defaults. To change one or more entries, write the appropriate directive line (see "Adding users to `/etc/udb` by using `/etc/udbgen`," page Procedure 26, page 201).

Login name field

user_name:	The user's login name must be a unique 1- to 8-character alphanumeric representation, in which the first character is alphabetic.

Encrypted password field

`passwd`: *encryption*:	Encrypted password to be stored in the user's record. The password content is not validated.

Password aging field

`pwage :force`, `superuser`, *max*, *min*, *time*:	Manipulate password age control fields by using `pwage`. If you omit a keyword, also omit its separating comma. The *max*, *min*, and *time* fields control how old a password can become (max), how long it must exist before being changed (min), and when it was changed (time). Neither *max* nor *min* may exceed 64 weeks.
`pwage :force`, `superuser`, *max*, *min*, +*age*:	In the second form of `pwage`, a + symbol preceding the last numeric value causes *age* to be interpreted as the amount of time to subtract from the time "now" to result in a value of the time-of-day clock that is age units in the past and then stores that value in the time field. Usually, this is intended to make it easy to set the current time in the field by using the value +0 as the age. You must precede the time with two commas if you do not specify *max* and *min* ages, because this part of the directive is position dependent,

	reading from the left to determine the meaning of the value string.
`pwage :` *max* `,` *min*:	The third form of `pwage` alters the *max* and *min* age fields.
`pwage :`:	The fourth form of `pwage` removes only age control from a record. All age control fields are set to a 0 or null state, which totally removes age control. After this has been done, all historical information is lost from the record. When the `YP` permbit is set (see permbits) and the password is being accessed from the database, password aging is disabled.
User ID field	
`uid`: *n*: `uid`: `next`:	Unique number that represents the user ID. If the value is `next`, the next highest user ID from the UDB is assigned to this user. The value 0 indicates a super-user login. The highest value that you may use is defined in `sys/param.h` as `UID_MAX-1`. You can reset the user database maximum user ID value without rebuilding the entire UDB from source by executing the `udbgen -m` command.
Group ID field	
`gids = | + | -`: *n1* `,` *n2* `,...,` *nn* `: gids = | + | -`: *g1* `,` *g2* `,...,` *gn* `:`	Comma-separated list of numeric group IDs or group names to which the user belongs. The group limit is 64. If group names are used, they must be found in the `/etc/group` file before executing the `udbgen` command.
User comment field	
`comment`: *text*:	Comments that consist of a maximum of 39 characters; white space is not removed. This field is often used for indicating the user's full name,

	although a site may have other uses for this optional field.
Login (home) directory	
`dir`: *directory* `:`	Default login (`home`) directory for this user relative to the `root` directory. The `dir:` *directory*: consists of a string of up to 63 characters. Typically, root is `/`; therefore, `dir` is based on the `root` (`/`) directory, but this does not have to be true. If you do not specify a value, the user is logged in under the `/` directory.
User shell at login	
`shell`: *sh_name*:	Default login shell. You can specify a maximum of 63 characters. Default value for *sh_name* is `/bin/sh`.
Account ID field	
`acids = | + | -`: *nl* `,` *n2* `,...,` *nn* `: acids = | + | -`: *al* `,` *a2* `,...,` *an* `:`	Account IDs. This is a list of up to 64 numeric account IDs or account names separated by commas. If you use account names, they must be found in the `/etc/acid` file as it existed before `udbgen` was executed.
Login root directory	
`root`: *directory* `:`	Login `root` directory. You can specify a string of up to 63 characters. `root` specifies the directory to which the base of the user's directory tree is set. (For further information, see the `chroot`(2) man page.)
Nice value	
`nice[b] :` *n* `:` `nice[i] :` *n* `:`	The `nice` value bias in the range 0 < *n* < 19 for batch (`[b]`) or interactive (`[i]`) processes. If you do not specify this field, the value from the default table or the released default value of 0 is used. This field is useful for getting different

interactive versus batch and NQS scheduling priority.

User resource limit fields

You can specify user resource limits for both batch and interactive processing in the UDB. The following is a list of some user limits that you may want to set; for a complete list of available limits, see the `udbgen`(8) man page.

Note: A UDB field setting of 0 means "infinite," except for tape access, where 0 means the user has no tape privileges.

CPU limits

Job CPU time limit
`jcpulim[b]`: *n*:
`jcpulim[i]`: *n*:
: Job CPU time limit (in seconds) for batch (`[b]`) or interactive (`[i]`) jobs. The default is unlimited.

Per-process CPU limit
`pcpulim[b]`: *n*:
`pcpulim[i]`: *n*:
: Per-process CPU limit (in seconds) for batch (`[b]`) or interactive (`[i]`) processes. The default is unlimited.

Memory limits

Job memory limit
`jmemlim[b]`: *n*:
`jmemlim[i]`: *n*:
: Job memory limit in 4096-byte blocks for batch (`[b]`) or interactive (`[i]`) jobs. The default is unlimited.

Per-process memory limit `pmemlim[b]`: *n*:
`pmemlim[i]`: *n*:
: Per-process memory limit in 4096-byte blocks for batch(`[b]`) or interactive (`[i]`) processes. The default is unlimited.

Process limits

Job process limit
`jproclim[b]`: *n*:
`jproclim[i]`: *n*:
: Job process limit for batch (`[b]`) or interactive (`[i]`) jobs. If you do not specify a value for this

	field, the default is the value of `/MAXUP` in `sys/param.h`.
SDS limits	Secondary data segments (SDS) are not supported on CRAY J90systems; you should ignore these fields and use the default setting.
Tape limits	
Job tape unit limit `jtapelim[b][t]`: *n*: `jtapelim[i][t]`: *n*:	Job tape unit limit for batch (`[b]`) or interactive(`[i]`) jobs. The integer value `t` represents the tape type. The default tape types are defined in the `DEVICE_GROUPS` section of the `/etc/config/tapeconfig` file. The first type defined in that section is represented by `t=0`, the second is `t=1`, and so on. If *n* is 0, the user is denied tape access.
File limits	
Per-process file allocation limit `pfilelim[b]`: *n*: `pfilelim[i]`: *n*:	Per-process file allocation limit in 4096-byte blocks for batch (`[b]`) or interactive (`[i]`) processes. If *n* is 0, the user's file allocation is unlimited.

Procedure 19: Adding a group to `/etc/group`

Note: An important step in adding a user record to the UDB is to assign the user to a group or groups. You may have to add group definitions so that you can make group assignments when you add user records.

As system administrator, you maintain the `/etc/group` text file, which contains the names of groups to which users belong. Groups are created to gather together users who have common needs for accessing files or programs.

You may have to edit the `/etc/group` file to add new group names to the file. The `/etc/nu` command does not allow you to enter a group name in the `gids` field until it has been entered in the `/etc/group` file; however, you may use group ID numbers even if no entry line for that group ID number is in the `/etc/group` file. In this case, a group name is created with the form `G-` *nnnnn*; *nnnnn* is the group ID number. The `/etc/nu` utility updates this file by adding login names to the group login name field. The file contains one entry for each UNICOS group. To delimit an entry line for a group, use a newline character.

To add a group to your system, edit the `/etc/group` file by adding an entry in the following format; **you must separate fields with a colon**:

group_name`:`*unused_password_field_string*`:`*group_id*`:`

Example:

```
ops:*:62:
```

group_name	Name that you choose to reflect the group of users. The group name consists of 1 to 8 alphanumeric characters. The first character must be alphabetic. By convention, lowercase characters are used for group names.
password	This field is not implemented under the UNICOS system. Place an unmatchable character string, such as `*`, in this field.
group_id:	The values 0 to 99 are reserved, by convention, for system-related groups; therefore, you can use group ID values 100 to `UID_MAX-1` for user groups. You should select the next available group ID number for the new group. A colon must follow the *group_id* field.
user	When you create a new group, this field remains blank. Ensure that a colon follows the *group_id* field. The `/etc/udbgen` and `/etc/nu` utilities maintain this field. The list of login names from the group ID field (`gids`) is placed automatically in this *user* field.

To see all group names to which a specified user belongs, use the `groups` command.

To complete adding user records to the UDB, use either the `/etc/nu` or `/etc/udbgen` utility. Section 7.5, page 183, describes how to use `/etc/nu`, and Section 7.6, page 199, describes how to use `/etc/udbgen`. To determine which utility will work best for you in a given situation, you may want to read both sections.

Procedure 20: Adding an accounting group to `/etc/acid`

As system administrator, you maintain the `/etc/acid` file, which contains the names of accounts associated with users. Accounting groups are implemented

for the accounting subsystem, allowing reports to generate information through accounting groups.

Just like the `/etc/group` file, you may have to edit this file to add new account names to the file. The `/etc/nu` command does not allow the use of account names in the `acids` field until an entry has been made in the `/etc/acid` file; however, you may use account ID numbers even if an entry line for that account ID number is not in the `/etc/acid` file. In this case, an account name is created of the form `A-` *nnnnn*; *nnnnn* is the account ID number. The file contains one entry for each UNICOS accounting group. A newline character delimits an entry line for each account.

To add an accounting group to your system, edit the `/etc/acids` file by adding an entry in the following format; **you must separate fields with a colon**:

account_name`:`*account_id*

account_name	Name that you select to reflect the accounting group (for easy identification in accounting reports). The name must consist of 1 to 79 alphanumeric characters. The first character must be alphabetic. Typically, lowercase characters are used for account names.
account_id	(Account identifier) You can use account ID values to `UID_MAX-1`.

Example:

```
markting:93
```

To complete adding user records to the UDB, use either the `/etc/nu` or `/etc/udbgen` utility. Section 7.5, page 183, describes how to use `/etc/nu`, and Section 7.6, page 199, describes how to use `/etc/udbgen`. To determine which utility will work best for you in a given situation, you may want to read both sections.

7.5 Using the `/etc/nu` utility

The `/etc/nu` utility is a prompt-driven utility for interactively adding, deleting, and modifying user records. It uses a configuration file called `/etc/nu.cf60`. This section describes the following topics:

- Procedure for changing `/etc/nu` configuration parameters

- Procedure for creating a file system to use with `/etc/nu`
- Procedure for adding user records to `/etc/udb` by using the `/etc/nu` utility
- Procedure for modifying user records by using the `/etc/nu` utility
- Procedure for deleting user records by using the `/etc/nu` utility

Procedure 21: Changing `/etc/nu` configuration parameters

Default values for the `/etc/nu` utility are in the `/etc/nu.cf60` configuration file. To change several parameters in this configuration file, either edit the file or use the menu system. You also can turn off (hide) prompts for UDB values that you want the `/etc/nu` program to accept automatically.

The following are common parameters you may want to change; for a complete list of changeable parameters and a description of each, see the `nu`(8) man page:

Parameter	Description
`DefaultAcids`	String that contains the default account IDs assigned to the UDB `acids` field.
`DefaultDr`	Default login root string assigned to the UDB `root` field.
`DefaultGids`	Default group IDs assigned to the UDB `gids` field.
`DefaultHome`	Default directory used to create the login directory for a new user if no `GroupHome` declaration exists for the user's assigned group ID. The directory created will be `$DefaultHome/`*username*; *username* is the user login name.
`DefaultShell`	String that contains the default login shell assigned to the UDB shell field.
`GroupHome`	Default directory used to create the login directory for a new user in the associated group if no `DefaultHome` declaration exists for the user. Multiple `GroupHome` lines should exist for each group ID number. For example, if the declaration is `GroupHome = 100 "/user1"`, user `john` (who is assigned to group `100`) will have, by default, a login directory of `/user1/john`.

Security feature variables	Security feature parameters are used only when you turn on the security feature or use the `-p` option of `nu`.

Note: If you have configured the `nu` utility to skip prompting for specific UDB fields, you must use `udbgen` to access these fields.

If you are using the menu system, select the `Configure System->NU Configuration`menu and its submenus. You can import the default configuration file by executing the `Import nu configuration` line; then modify the parameter settings and activate your changes. A sample `NU Configuration` menu screen follows:

```
Configure System
    ->NU Configuration
```

```
   NU Configuration

M-> Group home directories ==>
    Password aging ==>
    Tape Limits ==>
    Miscellaneous Limits ==>
    Ask about setting up Cray Station           1
    Login shell                                 /bin/csh
    Maximum login id length                     8
    nu log file                                 /usr/adm/nu.log
    Directory creation script                   /etc/nulib/nu1.sh
    Directory initialization script             /etc/nulib/nu2.sh
    Account removal script                      /etc/nulib/nu3.sh
    Account disabled script                     /etc/nulib/nu4.sh
    Default account ids                         0
    Default group ids                           0
    Default permbits                            none
    Default security level                      0
    Default maximum security level              0
    Default minimum security level              0
    Default default integrity class             0
    Default maximum integrity class             0
    Default valid compartment name string       none
    Default active compartment name string      none
    Default category name string                none
    Default valid category name string          none
    Default permission name string              none
    Default resource group UID                  0
    Default allocation shares                   100
    Default login root                          /
    Default home directory                      /u     Import nu
configuration ...
    Activate nu configuration ...
```

If you are not using the menu system, edit the `/etc/nu.cf60` configuration file.

Procedure 22: Creating a file system to use with `/etc/nu`

The `/etc/nu` command defaults, which are set in the `/etc/nu.cf60` file, expect a default home directory path of `/usr/home`.

To use the `/etc/nu` command , you must do **one** of the following:

- Change the `DefaultHome` parameter in the `/etc/nu.cf60` file to match what your site will use for a default `/home` path. (See Procedure 21, page 184.)

or

- Invoke the `mkdir` command to create a new directory called `home` in the `/usr` directory. This means that `home` will not be a separate file system, but just a subdirectory with files in the `/usr` file system.

or

- Create `/usr/home` as a file system and ensure that it is mounted on `/usr` when in multiuser mode.

If you are using the menu system, select the `Configure System ->File System (fstab) Configuration->Standard File Systems` menu, add your entries and update the form file. Then activate your changes through the `File Systems (fstab) Configuration` menu. A sample `Standard File System Configuration` menu screen follows:

```
Configure System
    ->..File System (fstab) Configuration
        ->.......Standard File Systems
```

```
   Standard File System Configuration

    Device Name        Mount Point     FsType  Freq  Pass R/O  Quota  User
    -----------        -----------     ------  ----  ---- ---  -----  ---- >
E-> /dev/dsk/home      /usr/home       NC1FS    1     2   rw
```

If you are not using the menu system, enter the following commands to accomplish this:

1. `/bin/mkdir /usr/home`
2. `/etc/mkfs -q /dev/dsk/home`
3. `/etc/labelit /dev/dsk/home home vol1` (optional step)
4. `/etc/fsck /dev/dsk/home`
5. `/etc/mount /dev/dsk/home /usr/home`

6. To ensure that /home is always both checked and then mounted on /usr when running at multiuser level, edit the /etc/fstab file to check and mount /home automatically. Such an entry would look like the following:

```
/dev/dsk/home   /usr/home      NC1FS  rw,CRI_RC="YES"  1     2
```

   At this point, the /etc/nu command will work as intended.

Procedure 23: Adding a user record to /etc/udb by using /etc/nu

Note: Before you begin this procedure, make sure you have completed the following:

- Determined the UDB field settings for the user account.
- Added needed group(s) to /etc/group to which the user will belong.
- Added needed account(s) that will be associated with the user if you have accounting implementation on your system.
- Created a /home or site-specific file system for user with /etc/nu.

For details on these procedures, see Procedure 18, page 176 through Procedure 22, page 186.

To use the nu utility to add a user to the /etc/udb file, use the following form of the nu utility:

```
/etc/nu  -a
```

When you use the nu utility, the /etc/udb, /etc/udb.public, /etc/group, /etc/acid, and /etc/passwd files are updated, or you can maintain private (testing) versions. When you maintain private versions, you can move or copy them to /etc to install the updates.

The nu utility queries you for values to UDB fields; it also lets you accept default values that have been specified in the program's configuration file (/etc/nu.cf60).

Note: A UDB field setting of 0 means "infinite," except for tape access, where 0 means the user has no tape privileges.For more information about user account field settings, see the procedure about determining settings for UDB fields, page Section 7.4, page 173.

The user's login name is something you provide when the /etc/nu utility prompts you. The user's login name must be a unique 1 to 8 alphanumeric

representation, in which the first character is alphabetic. Typically, the name is made up of lowercase alphabetic characters.

You may want to change the UDB password aging field to `force` so that the user must change the initial password when logging in for the first time. You must remove the `off` setting for the `DefaultAge` variable in the `/etc/nu.cf60` file (either manually or by using the menu system) so that the password aging field shows up and can be set when executing the `/etc/nu` script.

The `nu` utility has other options that you might want to use when adding a user, such as the following `-p` and `-c` options:

`-p` *dirname*	Modifies UDB files found under the *dirname* directory; lets you maintain private copies or test versions.
`-c` *dirname*	Uses the configuration file, `nu.cf60`, under the *dirname* directory; lets you specify an alternative configuration file.

To determine whether a record exists for a user, use the `udbsee` command.

Example `/etc/nu` session that adds a user

The following `nu` session adds login name `jones` (**`bold`** indicates what you would type; otherwise, you can use the default values):

```
# /etc/nu -a
cmd/nu/nu.c
Login name? (1-8 characters) [quit] jones
Enter password:
Please re-enter password:
Enter actual user name: John R. Jones
User id number? (max = 60000)  [2] 624
Which groups? (names or numbers, use commas, ? for list) [0] cray,test,trng,usrsrc
```

(See procedure for adding groups)

```
You selected groups:
100 0
cray , 100
104 1
test , 104
105 2
trng , 105
98 3
usrsrc , 98
Are these correct? (y or n) [y]
Login directory?  [/hot/u1/jones]
```

(This will be the user's home directory.)

```
Enter shell [/bin/csh]
Which accounts? (names or numbers, use commas, ? for list) [0]
You selected accounts:
```

(See procedure for acids)

```
root , 0
Are these correct? (y or n) [y]
User default login root?  [/]
```

(This will be the user's root directory; set to other than / to restrict acess to file systems.)

```
Resource group ID? (name or number, ? for list) [0] Users
Which permissions? (names or numbers, use commas, ? for list) [none]
You selected permbits:
none
Are these correct? (y or n) [y]
Allocation shares? (min=0) [100]
DEFAULT security compartments? (name1,name2,... or none, ? for list)  [default]
VALID security compartments? (name1,name2,... or none, ? for list)  [default]
Security permissions? (name1,name2,... or none, ? for list)  [default] Security levels?
(default max min)  [0 0 0]
Integrity classes? (default max)  [0 0]
DEFAULT integrity categories? (name1,name2,... or none, ? for list)  [default]
VALID integrity categories? (name1,name2,... or none, ? for list)  [default]
Do you want this user locked? (y or n) [n]
Do you want this user trapped? (y or n) [n]
Per job process limit for batch? (min=0)  [100]
Per job process limit for inter? (min=0)  [100]
Per job MPP PE limit for batch?  [none]
Per job MPP PE limit for inter?  [none]
Per job MPP time limit for batch?  [none]
Per job MPP time limit for inter?  [none]
Per job MPP barrier limit for batch?  [none]
Per job MPP barrier limit for inter?  [none]
Per process MPP time limit for batch?  [none]
Per process MPP time limit for inter?  [none]
Will the user be using the Cray Station? (y or n) [y] n
```

(No for Cray J90 systems.)

```
   1)  name ....... jones
   2)  passwd ..... v0u28k2K1wtX6 (encrypted)
   3)  pwage ........ force
   4)  comment .... John R. Jones
   5)  uid ........ 624
   6)  gids ....... cray (100)  test (104)  trng (105)  usrsrc (98)
   7)  dir ........ /hot/u1/rnl/tmp/jones
   8)  shell ...... /bin/csh
   9)  acids ...... root (0)
  10)  root ....... /
  11)  resgrp ..... Users (102)
  12)  permbits ... none
  13)  shares ..... 100
  14)  deflvl ..... 0
```

```
  15)  maxlvl ..... 0
  16)  minlvl...... 0
  17)  defcomps ... default
  18)  valcomps ... default
  19)  permits .... default
  20)  intcls...... 0            21)  maxcls...... 0
  22)  intcat...... default      23)  valcat...... default
  24)  disabled ... 0            25)  trap ....... 0
  26)  jproclim[b] ..       100       jproclim[i] ..       100
  27)  jcpulim[b] ...      none       jcpulim[i] ...      none
  28)  pcpulim[b] ...      none       pcpulim[i] ...      none
  29)  jmemlim[b] ...      none       jmemlim[i] ...      none
  30)  pmemlim[b] ...      none       pmemlim[i] ...      none
  31)  pfilelim[b] ..      none       pfilelim[i] ..      none
  32)  jsdslim[b] ...   1048576       jsdslim[i] ...   1048576
  33)  psdslim[b] ...   1048576       psdslim[i] ...   1048576
  34)  jtapelim[b][type0   ]      99      jtapelim[i][type0   ]     99
       jtapelim[b][type1   ]      99      jtapelim[i][type1   ]     99
       jtapelim[b][type2   ]      99      jtapelim[i][type2   ]     99
       jtapelim[b][type3   ]      99      jtapelim[i][type3   ]     99
       jtapelim[b][type4   ]      99      jtapelim[i][type4   ]     99
       jtapelim[b][type5   ]      99      jtapelim[i][type5   ]     99
       jtapelim[b][type6   ]      99      jtapelim[i][type6   ]     99
       jtapelim[b][type7   ]      99      jtapelim[i][type7   ]     99
  35)  jpelimit[b] ...      none       jpelimit[i] ...      none
  36)  jmpptime[b] ...      none       jmpptime[i] ...      none
  37)  jmppbarrier[b]       none       jmppbarrier[i]       none
  38)  pmpptime[b] ...      none       pmpptime[i] ...      none
Are these values OK? (y or n)  [y]
Entry looks like:
jones:co:John R. Jones
jones:ui:624:di:/hot/u1/rnl/tmp/jones:sh:/bin/csh:dr:/:pw:v0u28k2K1wtX6
jones:gi:100,104,105,98
jones:ai:0
jones:rg:102:as:100
jones:dc:default:cm:default:pm:default
jones:ic:default:vc:default
jones:pj[b]:100:pj[i]:100
jones:js[b]:1048576:js[i]:1048576:ps[b]:1048576:ps[i]:1048576
jones:tp:type0[b]:99:tp:type0[i]:99:tp:type1[b]:99:tp:type1[i]:99
jones:tp:type2[b]:99:tp:type2[i]:99:tp:type3[b]:99:tp:type3[i]:99
jones:tp:type4[b]:99:tp:type4[i]:99:tp:type5[b]:99:tp:type5[i]:99
jones:tp:type6[b]:99:tp:type6[i]:99:tp:type7[b]:99:tp:type7[i]:99+ test 1 -ne 0
```

```
+ rm -rf /hot/u1/jones
+ mkdir /hot/u1/jones
+ test /hot/u1/jones != /hot/u1/jones
+ test 0 -eq 0 -a 1 -ne 0
+ chgrp 100 /hot/u1/jones
+ chown 624 /hot/u1/jones
+ chacid -s root /hot/u1/jones
Do you wish to add more new users? (y or n)  [y] n
execing udbgen - please wait


udbgen complete (at this time, nu executes udbgen)
```

Procedure 24: Modifying user records by using /etc/nu

To update UDB fields, follow the same procedures as for adding new user logins, except use the `-m` option, rather than the `-a` option.

> **Note:** A UDB field setting of 0 means "infinite," except for tape access, where 0 means the user has no tape privileges.

Example `/etc/nu` session that modifies a user's login:

The following example shows how to update user login entries in the UDB by using the `nu` utility. The example changes the account group (`acids`) for user `jones` (**`bold`** indicates what you would type):

```
# /etc/nu -m
cmd/nu/nu.c               >>> Modify mode <<<Enter user identifier
(login or uid) [quit]: jones
Entry is now:
   1)  name ....... jones
   2)  passwd ..... v0u28k2K1wtX6 (encrypted)
   3)  pwage ........ force
   4)  comment .... John R. Jones
   5)  uid ........ 624
   6)  gids ....... cray (100)  test (104)  trng (105)  usrsrc (98)
   7)  dir ........ /hot/u1/jones
   8)  shell ...... /bin/csh
   9)  acids ...... root (0)
  10)  root ....... /
  11)  resgrp ..... Users (102)
  12)  permbits ... none
  13)  shares ..... 100
  14)  deflvl ..... 0
  15)  maxlvl ..... 0
  16)  minlvl...... 0
  17)  defcomps ... none
  18)  valcomps ... none
  19)  permits .... none
  20)  intcls...... 0            21)  maxcls...... 0
Press 'return' for the rest of the entry...
  22)  intcat...... none        23)  valcat...... none
  24)  disabled ... 0           25)  trap ....... 0
  26)  jproclim[b] ..       100       jproclim[i] ..       100
  27)  jcpulim[b] ...      none       jcpulim[i] ...      none
  28)  pcpulim[b] ...      none       pcpulim[i] ...      none
  29)  jmemlim[b] ...      none       jmemlim[i] ...      none
  30)  pmemlim[b] ...      none       pmemlim[i] ...      none
  31)  pfilelim[b] ..      none       pfilelim[i] ..      none
  32)  jsdslim[b] ...   1048576       jsdslim[i] ...   1048576
  33)  psdslim[b] ...   1048576       psdslim[i] ...   1048576
```

```
  34)  jtapelim[b][type0   ]    99    jtapelim[i][type0   ]        99
       jtapelim[b][type1   ]    99    jtapelim[i][type1   ]        99
       jtapelim[b][type2   ]    99    jtapelim[i][type2   ]        99
       jtapelim[b][type3   ]    99    jtapelim[i][type3   ]        99
       jtapelim[b][type4   ]    99    jtapelim[i][type4   ]        99
       jtapelim[b][type5   ]    99    jtapelim[i][type5   ]        99
       jtapelim[b][type6   ]    99    jtapelim[i][type6   ]        99
       jtapelim[b][type7   ]    99    jtapelim[i][type7   ]        99
  35)  jpelimit[b] ...      none       jpelimit[i] ...      none
  36)  jmpptime[b] ...      none       jmpptime[i] ...      none
  37)  jmppbarrier[b]       none       jmppbarrier[i]       none
  38)  pmpptime[b] ...      none       pmpptime[i] ...      noneSelect
field to be modified (1-38, q (discard changes), or e (make changes)):
9
Which accounts? (names or numbers, use commas, ? for list) [0] jones
You selected accounts:
jones , 624
Are these correct? (y or n) [y] Entry is now:
   1)  name ....... jones
   2)  passwd ..... v0u28k2K1wtX6 (encrypted)
   3)  age ........ force
   4)  comment .... John R. Jones
   5)  uid ........ 624    6)  gids ....... cray (100)  test (104)
trng (105)  usrsrc (98)
   7)  dir ........ /hot/u1/jones
   8)  shell ...... /bin/csh
   9)  acids ...... jones (624)
  10)  root ....... /
  11)  resgrp ..... Users (102)
  12)  permbits ... none
  13)  shares ..... 100
  14)  deflvl ..... 0
  15)  maxlvl ..... 0
  16)  minlvl...... 0
  17)  defcomps ... none
  18)  valcomps ... none
  19)  permits .... none
  20)  intcls...... 0           21)  maxcls...... 0
  22)  intcat...... none       23)  valcat...... none
```

```
  24)  disabled ... 0           25)  trap ....... 0
  26)  jproclim[b] ..       100       jproclim[i] ..       100
  27)  jcpulim[b] ...      none       jcpulim[i] ...      none
  28)  pcpulim[b] ...      none       pcpulim[i] ...      none
  29)  jmemlim[b] ...      none       jmemlim[i] ...      none
  30)  pmemlim[b] ...      none       pmemlim[i] ...      none
  31)  pfilelim[b] ..      none       pfilelim[i] ..      none
  32)  jsdslim[b] ...   1048576       jsdslim[i] ...   1048576
  33)  psdslim[b] ...   1048576       psdslim[i] ...   1048576
  34)  jtapelim[b][type0   ]      99     jtapelim[i][type0   ]      99
       jtapelim[b][type1   ]      99     jtapelim[i][type1   ]      99
       jtapelim[b][type2   ]      99     jtapelim[i][type2   ]      99
       jtapelim[b][type3   ]      99     jtapelim[i][type3   ]      99
       jtapelim[b][type4   ]      99     jtapelim[i][type4   ]      99
       jtapelim[b][type5   ]      99     jtapelim[i][type5   ]      99
       jtapelim[b][type6   ]      99     jtapelim[i][type6   ]      99
       jtapelim[b][type7   ]      99     jtapelim[i][type7   ]      99
  35)  jpelimit[b] ...      none       jpelimit[i] ...      none
  36)  jmpptime[b] ...      none       jmpptime[i] ...      none
  37)  jmppbarrier[b]       none       jmppbarrier[i]       none
  38)  pmpptime[b] ...      none       pmpptime[i] ...      noneSelect
field to be modified (1-38, q (discard changes), or e (make
changes)): e

Do you want to modify any more ./udb entries? (y or n) [y]n
done.
execing udbgen - please wait
udbgen complete.
```

Procedure 25: Deleting a user record by using `/etc/nu`

Caution: Deleting a user from the system requires more prudence than adding a user to the system because you may be removing valuable data from the system. Before removing the user from the UDB, you should determine whether any pertinent files are needed from the account. If files are needed, you can disable the user account by setting the `PERMBITS_RESTRICTED` bit in `permbits` to prevent the user from running. Setting `PERMBITS_NOBATCH` prevents batch jobs from running, and setting `PERMBITS_NOIACTIVE` prevents interactive jobs from running.

To remove a user account completely, follow these basic steps:

1. Save any important files the user owned on the system. You may want to back up these files to tape or have someone in the deleted user's department copy necessary files to another directory.

Example:

```
# rsv
# tpmnt -l nl -p /tmp/tapedev -v vsn -b 4096
# ls -a /usr/trng/jones | cpio -o > /tmp/tapedev
# rls -a
```

2. Disable the user's entry from `/etc/udb` by using the `/etc/nu -d` command, as follows:

```
# /etc/nu -d
```

You will be prompted to enter the login name or UID of the user you want to disable.

Note: If you want to keep accounting records in order or if you want to ensure that the user ID is not reused, you may choose not to complete step 3.

3. Remove the user from the UDB files by using the `/etc/nu -k` command, as follows. This command removes files under the user's login directory and that directory removes the user's mailbox and accounting records:

```
# /etc/nu -k jones
```

Example `/etc/nu` session that disables and removes a user's login

The following is an example `/etc/nu` session that disables and then removes user `jones` from the system:

```
# /etc/nu -d
cmd/nu/nu.c                >>> Deletion mode <<<Enter user identifier
(login or uid) [quit]: jones
Entry is now:
   1)  name ....... jones
   2)  passwd ..... v0u28k2K1wtX6 (encrypted)
   3)  pwage ........ force
   4)  comment .... John R. Jones
   5)  uid ........ 624
   6)  gids ....... cray (100)  test (104)  trng (105)  usrsrc (98)
   7)  dir ........ /hot/u1/jones
   8)  shell ...... /bin/csh
   9)  acids ...... jones (624)
  10)  root ....... /
  11)  resgrp ..... Users (102)
  12)  permbits ... none
  13)  shares ..... 100
  14)  deflvl ..... 0
  15)  maxlvl ..... 0
  16)  minlvl...... 0
  17)  defcomps ... none
  18)  valcomps ... none
  19)  permits .... none
  20)  intcls...... 0            21)  maxcls...... 0
  22)  intcat...... none           23)  valcat...... none
  24)  disabled ... 0            25)  trap ....... 0
  26)  jproclim[b] ..       100       jproclim[i] ..       100
  27)  jcpulim[b] ...      none       jcpulim[i] ...      none
  28)  pcpulim[b] ...      none       pcpulim[i] ...      none
  29)  jmemlim[b] ...      none       jmemlim[i] ...      none
  30)  pmemlim[b] ...      none       pmemlim[i] ...      none
  31)  pfilelim[b] ..      none       pfilelim[i] ..      none
  32)  jsdslim[b] ...   1048576       jsdslim[i] ...   1048576
  33)  psdslim[b] ...   1048576       psdslim[i] ...   1048576
```

```
34)  jtapelim[b][type0  ]     99    jtapelim[i][type0   ]        99
     jtapelim[b][type1  ]     99    jtapelim[i][type1   ]        99
     jtapelim[b][type2  ]     99    jtapelim[i][type2   ]        99
     jtapelim[b][type3  ]     99    jtapelim[i][type3   ]        99
     jtapelim[b][type4  ]     99    jtapelim[i][type4   ]        99
     jtapelim[b][type5  ]     99    jtapelim[i][type5   ]        99
     jtapelim[b][type6  ]     99    jtapelim[i][type6   ]        99
     jtapelim[b][type7  ]     99    jtapelim[i][type7   ]        99
  35)  jpelimit[b] ...      none       jpelimit[i] ...      none
  36)  jmpptime[b] ...      none       jmpptime[i] ...      none
  37)  jmppbarrier[b]       none       jmppbarrier[i]       none
  38)  pmpptime[b] ...      none       pmpptime[i] ...      none
Do you want to delete this entry? (y or n) [y] y

Entry for user jones has been deleted.
Do you want to delete any more users? (y or n) [y] n
execing udbgen - please wait
udbgen complete.

# nu -k jonescmd/nu/nu.c     71.7    10/30/92 09:04:35 (hot:./nu.cf60)

User jones is already disabled; no directory deletion done.

Entry for user jones has been killed.
execing udbgen - please wait.
udbgen complete.
#
```

7.6 Using /etc/udbgen

The /etc/udbgen utility lets you make changes to the UDB either interactively or as a batch job. The /etc/udbgen utility is actually the program underlying the /etc/nu utility. The batch capability of /etc/udbgen lets you add or modify many accounts at one time. When used with the udbsee command, the udbgen command with its directives can be a powerful and efficient tool in maintaining many accounts.

Note: If you have configured the nu utility to skip prompting for specific UDB fields, you must use udbgen to access these fields.

When using the `/etc/udbgen` command to create a new user login, you must specify the `create` directive. You may use the `/etc/udbgen` program in the following three ways:

- Interactive submission: When using `udbgen` interactively, you must use the `quit` directive to exit the utility; this action updates the UDB files.
- Batch submission: You can place directives in a file and submit them all at once to the UDB.
- Individual submission: You can submit directives individually.

With all three methods, you can enter multiple UDB field names and their values. You can place more than one field on a `create` directive line or each field may be on separate lines. The recommended method for `udbgen` is the batch approach: place the directives in a file and then use that file as input to the `/etc/udbgen` command.

You may choose, for test purposes, to modify a private copy of the UDB files, rather than the ones contained in the `/etc` directory; see the example on page 180 of the following procedure.

The format of the `create` directive is as follows:

`create:`*user_name*`:uid:`*uid_number*`:gids:`*group_names*`:field_name:`*field_value*`:`

Field	Description
`create:`	Adds the specified user's information to the UDB; if a UDB record already exists for this user name, a warning message is displayed and the record is not changed.
user_name `:`	User login name to be created; must be a unique name within the database.
`uid:` *uid_number* `:`	Specifies a `uid` value or `next` to have `udbgen` assign a value.
`gids:` *group_names* `:`	Specifies one or more group IDs or names.
`field_name:` *field* `_` *value* `:`	One or more field values; you can put multiple field values

on one line, or you can put each field on a separate line.

Note: You **must** include the colon at the end of the directive.

The remainder of this section provides the following:

- Procedure for adding users by using the `/etc/udbgen` utility
- Procedure for transferring initial files to the login directory when using the `/etc/udbgen` utility
- Procedure for updating user logins in the UDB by using the `/etc/udbgen` utility
- Procedure for deleting users from the UDB by using the `/etc/udbgen` utility

Procedure 26: Adding users to `/etc/udb` by using `/etc/udbgen`

Note: Before you begin this procedure, make sure you have completed the following:

- Determined the UDB field settings for the user account.
- Added needed group(s) to `/etc/group` to which the user will belong.
- Added needed account(s) that will be associated with the user if you have accounting implementation on your system.

For details on these procedures, see Procedure 18, page 176 through Procedure 20, page 182. You also may choose, for test purposes, to modify a private copy of the UDB files, rather than the ones contained in the `/etc` directory; to do this, see the example at the end of this procedure.

1. Complete **one** of the following three methods of using `/etc/udbgen` to add a user to your system:

- Create a file of `/etc/udbgen` directives that has this format; you must include the colon at the end of the line:

`create:`*username*`:uid:`*uid_number*`:gids:`*group_names*`:field_name:`*field_value*`:`
`field_name:`*field_value*`:field_name:`*field_value*`:`

Then submit the changes to the `/etc/udbgen` database by entering the following command line:

```
# /etc/udbgenudbgen_directives_filename
```

Example of directives submitted in a batch file (**bold** indicates what you type):

```
# vi udb.source
(Enterudbgen directives.)
# cat udb.source
create:john:uid:next:
comment:John Smith:
pwage:force:
gids:cray,test,trng:
acids:testing,training:
dir:/user1/trng/john:
shell:/bin/csh:
resgrp:102:
psdslim[b]:1000000:
pmemlim[i]:8000:
psdslim[i]:1000000:
shares:100:
# udbgen udb.sourceInput style: udb
Added 1 record
#
```

or

- Type `/etc/udbgen` to enter interactive mode and reply to the prompt that has the following format; you must include the colon at the end of the line:

`create:`*user_name*`:uid:`*uid_number*`:gids:`*group_names*`:field_name:`*field_value*`:quit`

or

- If you must make only one or two changes, you can submit directives to the database individually by typing a line that has the following format:

  `/etc/udbgen -c "create:`*user_name*`:uid:`*uid_number*`:field_name:`*field_value*`:"`

Example of directives submitted individually (**bold** indicates what you would type):

```
# /etc/udbgen -c "create:john:shell:/bin/sh:uid:next:gids:cray,test,trng:"
# /etc/udbgen -c "update:john:resgrp:102:"
# /etc/udbgen -c "delete:mary:"
```

Verify that your entry was added by using the udbsee(1) command.

Assign the initial password for the user.

If you use the udbgen command, you cannot set the password field to an initial password. Instead, you must use the /bin/passwd command to change the password. You and the new user must agree on an initial password for the account. Choose one that is not easy to guess. Only the super user may change another user's password. You may have chosen to set the UDB passwd field to * or left it empty (indicated by two contiguous colons, ::). If no assignment was made to this field during the user's login creation, the field is assigned the * symbol.

Caution: If you have left the password field empty, anyone who knows the login can use this account. Your system is open to abuse.

Example:

```
# /etc/udbgen  -c "create:john:uid:next:gids:cray,test,trng:"
# /bin/passwd john
New password:

(The password is not visible on your screen.)

Reenter new password:
#
```

Create a login directory for the user.

The /etc/udbgen command does not automatically create a login (home) directory. The dir for each entry in /etc/udb specifies the initial working directory (home) for each user at login time. As the system administrator, you must create that directory by using /bin/mkdir. Because you currently have a user ID of 0 and a group ID of 0, the directory created also will be assigned these permissions. You must make the user's directory accessible to the user by changing the permissions, group, and ownership of the directory. This involves executing the chmod(1), chgrp(1), and chown(1) commands.

The following is a brief review of how UNICOS permissions are defined (followed by examples).

Format:

`/etc/chmod` *permissions filename*

Permissions are set up in three groups, and they can be displayed by using the `ls -l` command:

	User	Group	Other
-	rwx	rwx	rwx
d	rwx	rwx	rwx

The `-` symbol indicates that the file is a regular file. The `d` indicates that the file is a directory file. The `r`, `w`, and `x` indicate permissions for read, write, and execute, respectively. If the `r`, `w`, or `x` is present, that permission is set for that category of users (user, group, or other). If a `-` symbol is in any of the fields, except for the first field, that permission is turned off for that category of users. You can represent these fields numerically, as follows:

400, 40, and 4 = Readable by user, group, and other, respectively.

200, 20, and 2 = Writable by user, group, and other, respectively.

100, 10, and 1 = Executable by user, group, and other, respectively.

Example:

To give user, group, and others read, write, and execute permissions, calculate the permission fields to use with the `chmod` command:

Also see the following examples.

Example of creating a login directory:

1. Create the directory by using the `/bin/mkdir` command.

   Format: `/bin/mkdir` *new_login_directory_name*

   Example: `# mkdir /user1/trng/jones`

2. Change the ownership of the directory by using the `/bin/chown` command.

   Format: `/bin/chown` *new_login_name new_login_directory_name*

   Example: `# /bin/chown jon /user1/trng/jones`

3. Change the group of the directory by using the `/bin/chgrp` command.

Format: `/bin/chgrp` *new_group new_login_directory_name*

Example: `# /bin/chgrp swtng /user1/trng/jones`

4. Change the permissions of the directory by using the `/bin/chmod` command.

   Format: `/bin/chmod` *permissions new_login_directory_name*

   Example: `# /bin/chmod 761 /user1/trng/jones`

   Note: If you want to move existing files into the login directory, use the procedure to transfer initial files to the login directory (see Procedure 27, page 208).

Examples of `/bin/chown`, `/bin/chgrp`, and `/bin/chmod` follow:

```
sn1601% pwd
/sn1601/sdiv/unicos/jones%
su root
Password:
# ls -la
total 21
drwx------   3 jones   os           4096 Mar 21 17:43 .
drwxr-xr-x 100 root    root         4096 Mar 24 13:14 ..
-rw-r--r--   1 jones   os           121 Sep 13  1991 .cshrc
-r--r--r--   1 root    root         192 Oct 11 17:28 mnt
-rw---x--x   1 root    os           82 Oct 11 17:31 umnt

# /bin/chown jones mnt
# ls -la
total 21
drwx------   3 jones   os           4096 Mar 21 17:43 .
drwxr-xr-x 100 root    root         4096 Mar 24 13:14 ..
-rw-r--r--   1 jones   os           121 Sep 13  1991 .cshrc
-r--r--r--   1 jones   root         192 Oct 11 17:28 mnt
-rw---x--x   1 root    os           82 Oct 11 17:31 umnt

# /bin/chgrp tng mnt
# ls -la
total 21
drwx------   3 jones   os           4096 Mar 21 17:43 .
drwxr-xr-x 100 root    root         4096 Mar 24 13:14 ..
-rw-r--r--   1 jones   os           121 Sep 13  1991 .cshrc
-r--r--r--   1 jones   tng          192 Oct 11 17:28 mnt
-rw---x--x   1 root    os           82 Oct 11 17:31 umnt

# /bin/chmod 761 mnt
# ls -la
total 21
drwx------   3 jones   os           4096 Mar 21 17:43 .
drwxr-xr-x 100 root    root         4096 Mar 24 13:14 ..
-rw-r--r--   1 jones   os           121 Sep 13  1991 .cshrc
-rwxrw---x   1 jones   tng          192 Oct 11 17:28 mnt
-rw---x--x   1 root    os           82 Oct 11 17:31 umnt
```

Example of using a private copy of UDB files for test purposes:

You may choose, for test purposes, to modify a private copy of the UDB files, rather than the ones contained in the `/etc` directory. After you have set up a private UDB, use the `-p` option with the `/etc/udbgen` command, as follows:

1. Create a directory to contain your private UDB, as follows:

```
# mkdir/myudb
```

2. Create a `group` file in your new directory, as follows:

```
# cp /etc/group ./myudb/group
```

3. Create an `acid` file in your new directory, as follows:

```
# cp /etc/acid ./myudb/acid
```

To verify that the user login was created correctly, use the `udbsee` command. Then move or copy the UDB files contained in the directory specified by the `-p` option into the `/etc` directory, as shown in the following example.

Example of directives submitted interactively (**bold** indicates what you would type):

```
# /etc/udbgen -p /user1/jones/etc
/etc/udbgen: 1>create:john:uid:next:
/etc/udbgen: 2>comment:John Smith:
/etc/udbgen: 3>pwage:force:
/etc/udbgen: 4>gids:cray,test,trng:
/etc/udbgen: 5>acids:testing,training:
/etc/udbgen: 6>dir:/user1/trng/john:
/etc/udbgen: 7>shell:/bin/csh:
/etc/udbgen: 8>resgrp:102:
/etc/udbgen: 9>psdslim[b]:1000000:
/etc/udbgen: 10>pmemlim[i]:8000:
/etc/udbgen: 11>shares:100:
/etc/udbgen: 12>quit
Added 1 record
```

Procedure 27: Transferring initial files to the login directory when using /etc/udbgen

To transfer initial files to the login directory when using /etc/udbgen, follow these steps:

1. You may want to create a directory, such as /usr/skel, to hold templates for such files as .profile, .cshrc, .login, and .exrc. The /etc/udbgen command does not automatically copy skeleton files to the user's home directory.

For descriptions of how to set up the /etc/profile and /etc/cshrc files, see Section 7.7, page 212.

2. After you have created /usr/skel and the template files, copy the desired files to the user's home directory by using the cpset command, which lets you specify the mode, owner, and group of the destination files. cpset installs object files in binary directories.

Example:

```
# cpset /usr/skel/.cshrc /usr/home/john 700 john trng
# cpset /usr/skel/.login /usr/home/john 700 john trng
```

Procedure 28: Updating user logins in the UDB by using /etc/udbgen

The method for updating user logins is similar to that for adding new user logins. Different directives are used, however, to accomplish the task. Some of the fields (such as the gids field) have editing suffixes that may be used with the /etc/udbgen command. These editing suffixes are as follows:

=	Indicates that the next value(s) replace the existing value.
+	Indicates that the following values will be appended to the current values of the field. (see Example 6, page 209).
-	Indicates that the following values will be deleted from the list of current values for the field.

You may use the following steps to change every field except the passwd field. To change the passwd field, use the /bin/passwd command, as described in step 3 of the procedure for adding users to /etc/udb by using /etc/udbgen (see the passwd man page for further information). You change the user login name by deleting the old user login and creating a new user login under the new name.

Caution: It is a **very dangerous** practice to delete the user login of a user who may be logged in on the system. This procedure should wait until a time when you know the user is not running anything on the system.

The following steps summarize the user login update process:

1. Decide which UDB fields you want to change.
2. If the user will be placed in a new group that you will reference by name, make the desired entry in `/etc/group`.
3. If the user will be placed in a new account group that you will reference by name, make the desired entry in `/etc/acid`.
4. Make the desired entry in `/etc/udb` by using the `/etc/udbgen -c` command with the `update` directive. If you change the user's login directory, create the new directory and copy over any existing files to the new directory.

Examples of updating user logins by using `udbgen`:

The following examples show how to update user login entries in the UDB by using the `/etc/udbgen` command:

Example 6: Adding a new group ID

This example adds the new group ID (`gids`) `usrsrc` to user `john`:

```
# /etc/udbgen -c "update:john:gids+:usrsrc:"
```

Example 7: Changing the user's shell

This example changes the login shell for user `john` to the POSIX shell. Because the POSIX shell was specified, you also must create a `.profile` file.

```
# /etc/udbgen -c "update:john:shell:/bin/sh:"
# cpset /usr/skel/.profile /user1/trng/john
```

Example 8: Changing the user's login directory

This example changes the login directory for user `john` to `/usr1/john`. Formerly, user `john` 's login directory was `/user1/trng/john`. The `mkdir`, `chown`, `chgrp`, and `chmod` commands are used to create the `/usr1/john`

directory and to assign proper ownership and permissions for the directory. The last three commands remove `john`'s old login directory.

Caution: If the user is running anything on the system, you should never change a home directory. This is especially critical if libraries are removed.

```
# /etc/udbgen -c "update:john:dir:/user1/john:"
# mkdir /user1/john
# chown john /user1/john
# chgrp trng /user1/john
# chmod 700 /user1/john
# cd /user1/trng/john
# find . -print | cpio -pdm /user1/john
# rm -rf /user1/trng/john
```

Example 9: Using the udbsee command as a filter to add an account ID (acid)

This example uses the `udbsee` and `udbgen` commands to add an account ID (`acid`) of 10 to all user accounts that have a group ID (`gid`) of 103. In all, 46 login accounts are affected. This is a typical example of how a large-scale update to the UDB is performed:

```
# udbsee -a -e 'gids ~ "103"' -f "name" -m 'update:%s:acids+:10:\n'|/etc/udbgen
46 entries converted to source
Input style: udb
Updated 46 records
#
```

Example 10: Changing the user's password

This example uses the `/bin/passwd` command to change the password for user `john`:

```
# /bin/passwd john
New password:

(The password is not visible on your screen.)

Reenter new password:
```

Procedure 29: Deleting a user from the UDB by using `/etc/udbgen`

Caution: Deleting a user from the system requires more prudence than adding a user to the system, because you may remove valuable data from the system. Before removing a user from the UDB, you should determine whether any pertinent files are needed from the account. If files are needed, you can disable the user account by setting the *passwd* field to `*`, using the `udbgen` command.

It is a **very dangerous** practice to delete users who may be logged in. This procedure should wait until a time when you know the user is not running anything on the system.

To remove a user account completely, perform the following basic steps:

1. Make it impossible for the user to log in by using the `udbgen` command, as follows:

```
# /etc/udbgen -c "update:john:passwd:*:"
```

2. Ensure that the user has nothing running on the system.
3. Save any important files the user owned on the system. You may want to back up these files to tape or have someone in the deleted user's department copy necessary files to another directory.

Example:

```
# rsv
# tpmnt -l nl -p /tmp/tapedev -v vsn -b 4096
# ls -a /usr/trng/john | cpio -o > /tmp/tapedev
# rls -a
```

4. Delete files from the user's home directory and any other directories on the system by using multiple `rm` commands, and remove the user's `home` directory. Also remove the user's mailbox, `/usr/mail/` *username*.

Example:

```
# rm -rf /user1/trng/john
# find  /  -user  john  -exec  rm  {}\;
# rm  -f  /usr/mail/john
```

Note: If you want to keep accounting records in order or if you want to ensure that the user ID is not reused, you may choose not to complete step .

5. Remove the user from the UDB files by using the `delete` directive of the `udbgen` command. The `delete` directive has the `delete`: *userid*: format; you must include the colon at the end of the directive:

```
# /etc/udbgen -c "delete:john:"
```

7.7 Maintaining user environment files

This section describes the following procedures:

- Setting up an `/etc/profile` file
- Setting up an `/etc/cshrc` file
- Transferring users to another file system

When the user logs in to the system, the `/bin/login` script executes the program in the UDB `shell` field. If you specify `/bin/sh` or `/bin/rsh`, the following files are executed (if they exist) by `/bin/sh` or `/bin/rsh`:

```
/etc/profile
$HOME/.profile
```

If you specify `/bin/csh`, the program executes the following files in the order shown:

```
/etc/cshrc
$HOME/.cshrc
$HOME/.login
```

As the administrator, you must maintain the `/etc/profile` and `/etc/cshrc` files, which are described in this section.

Procedure 30: Setting up an `/etc/profile` file

When the `/bin/login` script invokes the default shell (`/bin/sh`, which is the POSIX shell, or `/bin/rsh`, which is the restricted shell), it reads and executes

the commands found in the `/etc/profile` file. This lets you set up a standard environment for all users. Users may alter this setup environment through the `$HOME/.profile` file to personalize their environment.

Note: If you want to change the `.profile` file, see the `sh`(1) man page, which describes the supported shells and the `shell` script syntax.

A typical system profile, `/etc/profile`, might perform the following functions (the line references refer to the example that follows):

1. Set and export the directory search path (lines 4 and 5).
2. Set the file creation mask, using the `umask` command (line 6).
3. If using one of these shells (line 8), do the following functions:

- Display the message of the day (line10).
- If the `.motd` file exists, display it (lines 11 through 13).
- Check for the existence of mail (lines 15 through 17).
- Display the names of current news items (lines 18 through 20).
- Set the user's prompt (lines 21 through 25).
- Set effective user ID if user uses the `/bin/su` command (line 27).

An example `/etc/profile` file follows:

```
 1#       SCMID@(#) /etc/profile
 2
 3 trap "" 1 2 3
 4 PATH=/bin:/usr/bin:/usr/ucb:/usr/lbin
 5 export PATH
 6 umask 022
 7 case "$0" in
 8 -sh | -rsh | -ksh)
 9      trap : 1 2 3
10      cat /etc/motd
11      if [ -f ../.motd ] ; then
12              cat ../.motd
13      fi
14      trap "" 1 2 3
15      if mail -e ; then
16           echo "You have mail."
17      fi
```

```
18      if [ -x /usr/bin/news -a -d /usr/news ] ; then
19              news -n
20      fi
21      if id | grep 'uid=0' > /dev/null ; then
22              PS1="`uname -n`# "
23      else
24              PS1="`uname -n`$ "
25      fi
26      ;;
27  -su)
28      :
29      ;;
30  esac
31  trap 1 2 3
```

Procedure 31: Setting up an /etc/cshrc file

If a user has chosen to run the C shell (`/bin/csh`), the commands found in `/etc/cshrc` are executed. You should set up the same environment variables found in `/etc/profile` in the C shell start-up file.

> **Note:** If you want to change the `.profile` file, see the `csh`(1) man page, which describes the `shell` command syntax.

Users can alter this setup environment through the `$HOME/.cshrc` and `$HOME/.login` files to personalize their environment. A typical system profile, `/etc/cshrc`, might perform the following functions (the line references refer to the example that follows):

1. Set and export the shell variable (line 3).
2. Set and export the directory search path (line 4).
3. Start up the history mechanism (line 5).
4. Set the file creation mask, using the `umask` command (line 6).
5. Display the message of the day (line 7).
6. Check for the existence of mail (lines 8 and 9).
7. Display the names of current news items (lines 10 through 12).
8. Set the user's prompt (line 13).

An example `/etc/cshrc` file follows:

```
 1  # SCMID@(#) etc/cshrc
 2
 3  setenv SHELL /bin/csh
 4  set path = ( /bin /usr/bin /usr/ucb /usr/lbin /usr/ucb)
 5  set history = 24
 6  umask 022
 7  cat /etc/motd
 8  mail -e
 9  if ( $status == 0 ) echo "You have mail."
10  if (-d /usr/news) then
11    news -n
12  endif
13  set prompt = "'uname -n`$prompt"
```

Procedure 32: Transferring user accounts to another file system

If a group of users must be transferred to another file system, use the `cpio` command to copy them. If all users are copied at the same time, the `cpio` command helps preserve any links the users had among their files.

Caution: Be sure to update the user's home directory in the UDB. When you do this, also ensure that none of the users are running anything on the system.

Example:

```
# cd /user_a
# find john tom sue mike alice -print | cpio -pdm /user_b
# rm -rf john tom sue mike alice
```