Username and Groupname Sizes on HP-UX

ABSTRACT

HP-UX 11i Version 3 provides expanded username and groupname interfaces. This capability allows setting of user and group names that are longer than the previously supported value. HP provides this enhancement to support the user/group naming requirements of some customers.

The default operating system configuration settings ensure compatibility for applications using user and group names by enforcing compatible name sizes. However, if the administrator configures the system for expanded user and group names, code changes may be necessary to accommodate expanded names. This paper is intended for both the system administrators and program developers. It describes how an administrator can enable the capability. Also it describes how a program developer can determine whether programs have dependencies on user or group name sizes, and how to enhance them to accommodate expanded user and group names. The enhancements in most cases enable the program to operate correctly on any HP-UX version whether or not expanded names are assigned.

NOTE: Enabling the expanded name capability and creating long user or group names may result in incorrect operation of some existing applications. It is important to read and understand all the information in this document before the use of long user and group name is attempted.

1.0 Introduction

The user and group names have default length limits of 8 and 16 bytes[1], respectively. However, the system administrator can configure the system to expand both these limits to 255 bytes. AIX (255 bytes) and Windows (20 bytes) support longer user and group names. Hence, it is necessary to support longer user and group names in a heterogeneous environment.

Programs using the C library APIs (getpw* and getgr*) which return the groupname and username information in a structure, do not require any modification because the structure element is a character pointer (char *). The structure can scale to any size. Thus, the FLV mechanism is not required to get enhanced version of the C library APIs to work under the expanded user/group name. However, APIs and applications that assume the current maximum length are impacted due to this change. Therefore, the HP-UX default configuration ensures that the runtime environment is compatible for existing application binaries. This paper describes how to enable the expanded user and group name capability, the risks to applications, limitations in HP product support of long user and group names, how to determine whether software code has dependencies upon the default name size limits, and how to enhance programs to accommodate expanded name sizes.

This paper is organized as follows:

Enabling the system for expanded User and group names: Describes, for system administrators, how to enable and use the expanded name capability.

Output formats: Describes, for system administrators and programmers, how to format the output of utilities/commands, when the user and group names are longer.

HP Product Limitations: Describes the limitations in some HP products with respect to the size of user and group names.

Programmer Notes: Describes briefly how software developers can enhance their applications to accommodate expanded user and group name sizes.

[1] HP-UX does not document 16byte limit on groupnames. However, SAM allows 16bytes groupnames.

Testing Guidelines: Describes briefly how to test an application to ensure that it works well in an expanded user and group name enabled environment.

2.0 Enabling the System for expanded User and Group names

This section describes how an administrator enables the capability to set longer user and group names.

WARNINGS

- Verify whether the versions of all applications which use user and group names are validated for expanded user and group name sizes before setting the names to longer values
- HP recommends that, when longer user or group names are assigned, the proper operation of all applications be validated in a test environment before they are used in a production environment.
- Some HP products offer limited in their support to long user and group names. (See HP Product Limitations).
- Some optional HP software products, if present on the system, must be updated to support expanded names.
- Third party software products may be limited in their support for long user or group names. Associated documentation should be consulted for products which may use user or group names for acknowledgement that expanded names are supported.
- Locally developed programs (For example, from an in-house software development organization) may offer limited support for long user or group names.
- Any program which uses LOGIN_NAME_MAX Macro may get truncated usernames.
- Any program that uses the obsolescent API cuser ()and L_cuserid Macro can misbehave.
- Refer to the programmer-oriented sections of this document for details about application issues.

ENABLING EXPANDED USER AND GROUP NAMES

The default operating system configuration does not enable the setting of user name in excess of 8 bytes or a group name in excess of 16 bytes. The system administrator must explicitly enable the system for expanded username/group name by using the lugadmin command. The details of using lugadmin command are documented in its man page. The lugadmin –e option enables long username. Once the system is enabled for long user/group name, it cannot be disabled. A disable option is not provided due to the impracticality of automatically finding all instances of stable storage that may contain names in excess of the default limits.

Once the expanded user/group name feature is enabled, all the user and group management commands (useradd, usermod, userdel, groupadd, groupmod and groupdel) will allow the user to create/update users with expanded user/group names. All other core components of HP-UX will work fine with expanded user/group names except for the ones listed in HP Product Limitations.

3.0 Output Formats

With user/group names increasing to 255 bytes, display of long user/group names on terminals can have issues. Formatting guidelines for displaying the user/group names are followed across all HP supplied applications/utilities that support long user/group names. The same guidelines must be followed by all application developers as well.

3.1 Issues due to Display of Long User/Group Names

- Extending user/group names beyond 8 bytes disrupts the tabular output of commands such as ps that do not truncate user/group names.
- Commands such as ls that truncate user/group names provide incomplete/erroneous data to callers.
- Script writers often use the knowledge of the column widths of different fields in the output by commands such as ls. This gathered data is then used to perform some other action. Changes in column width of user/group name fields break these scripts.
- Truncation of user/group names can result in script errors.
- Truncation of user/group names may result in misidentification if two user/group names having the same prefix.

3.2 Display Guidelines Followed by HP Supplied Commands and Utilities

Under an expanded user/group name enabled environment following display guidelines are followed by HP supplied commands and utilities. Some of the commands/utilities may not follow the display guidelines due to the limitations posed by the output of those commands. Exceptions to these guidelines are documented in the man page of the command/utility.

3.2.1 Default Display Mode

Default display width of zero is set when lugadmin -e is executed without specifying the -d option. It can be changed at any time by executing lugadmin with the -d option. For user and group names longer (in bytes) than the default length for that command/utility, the name is truncated to the maximum default length and the last printed character is replaced with a plus sign (+). Consider an example of a command which uses a field width of eight characters for the display of username in a system where long user/group name is not enabled. When the long user/group name is enabled and default display mode is set, the command continues to use the field width of eight characters for displaying the user name. Thus if the username is nine characters, then the first seven characters of the username are printed followed by '+'.

3.2.2 System-wide or user specific configuration of display width.

The lugadmin command with –d option can be used to change the system-wide configuration.
For example, when lugadmin –d 64 is issued, the display width is set to 64, thus the user/group name field width of the applications is set to 64. For all the user/group names that are longer than 64 bytes, the name is truncated to 63 characters, and last character in the 64th column is replaced with "+" character to indicate the truncation.
If an individual user wants a value for the display width that is different from the value set with the system-wide configuration, the user can set the UG_DISPLAY_WIDTH environment variable.
The environment variable overrides the system-wide configuration.
Man page of lugadmin provides the details of the display width configuration.

3.3 API for Display Width

In order to maintain the consistency of the display of user/group names and ease of future modifications, a new API, ug_display_width () is introduced to query the display width to be used based on the formatting rules described in section 3.2. The man page of ug_display_width () describes how to use it.

3.4 Types of Applications That Need to Consider the Display Guidelines

The following are examples of applications which need to consider the display guidelines:

- Applications that provide output formatted in columns, such as who, ls, ps, etc.
- Applications that display output in fields, such as finger.
- Applications that attempt to limit output to the conventional terminal display width of 80 characters.

- Applications that may have their output parsed by other applications, but do not clearly define rules for parsing the output.

3.5 Type of Applications that need not consider the Display Guidelines

The following applications need not consider the formatting rules:

- Applications that do not display user or group names.
- Applications that output user/group names, but their output is never parsed by other applications.
- Applications that have consistent and well defined rules for determining beginning and end of user/group name fields when parsed by other applications.
- Applications that are not of the type listed in section 3.4

4.0 HP Product Limitations

This section describes HP software product limitations in the support of user names more than eight bytes in length and/or group names that are more than 16 bytes in length. Some of these limitations may be removed in product or OS updates made available after the publication of this document. Note that third party and in-house software providers may also have some limitations. While this list is intended to describe all the HP product limitations, it may not be complete. Any other important limitations encountered by customers should be reported to HP.

4.1 SNA user and administration

The SNA administration tools snapadmin and xsnapadmin allow 32-character user/group names. These tools are not enhanced for long user/group names.

4.2 Trusted mode

Long user/group name functionality cannot be enabled under trusted mode. If the system is long user/group name enabled, the system cannot be switched to trusted mode.
Although HP-UX Trusted Mode does not support the long user and group name functionality, standard mode security extensions in the base OS, including /etc/shadow and the audit subsystem, do support long user names. Please refer to http://software.hp.com/portal/swdepot/displayProductInfo.do?productNumber=StdModSecExt for more information.

5.0 Programmer Notes - Making Programs Expanded User/group Name Clean

Function Level Versioning (FLV) mechanism does not exist because it is not required by any of the APIs which return username/groupname. The APIs getpw* and getgr* are scalable to handle the expanded username/groupname. Moreover, the value for the symbolic constant _POSIX_LOGIN NAME MAX remains unchanged in 11i v3. It continues to be nine. As per the standards, this symbolic constant has to be the most restrictive (minimum) value a conforming implementation should support, which is specified to be nine.

All scalable applications are unaffected by the user and group name expansion. However, there may be some applications that are not scalable, in which case their code needs modification.

This section describes how software developers can determine whether their product needs any consideration and, if so, how to enhance the product to accommodate expanded user and group names.

Enhanced programs also operate correctly in systems where the names are not long. Programs that have no dependencies on the user and group name size, or which have been enhanced for the expanded sizes, are said to be expanded user and group name clean.

Developers should read this section in its entirety, along with the related appendices, to ensure all issues are understood.

Developers should also become familiar with the administrative issues described in this paper.

5.1 Checklist to Find out Whether Your Program is Long User/Group Name Clean and Methods to Make Them Clean

Please note, this checklist, assumes that "name" means user name or group name.

1. Does your program depend on LOGIN_NAME_MAX /_POSIX_LOGIN_NAME_MAX to identify the maximum supported size of username/group name on the system?
    - Ø If so, use sysconf (_SC_LOGIN_NAME_MAX) to query the maximum supported size of the user/group names. LOGIN_NAME_MAX is a constant and does not represent the run time value. _POSIX_LOGIN_NAME_MAX is a constant which represents only the minimum supported value.
    - Ø Because _SC_LOGIN_NAME_MAX system configuration variable is not available on earlier versions of HP-UX, users may use conditional compilation in the following manner:

```
          #include <unistd.h>
          …….
          …….
          user_buf_size = 0;
#ifdef _SC_LOGIN_NAME_MAX
user_buf_size = sysconf (_SC_LOGIN_NAME_MAX) + 1;
#endif
if (user_buf_size <= 0) user_buf_size = 256;
   userbuf = malloc(user_buf_size);
```

    - Ø If an array of size LOGIN_NAME_MAX/_POSIX_LOGIN_NAME_MAX is used, then either the array size needs to be changed to 256 or it can be changed to a pointer and the memory for it can be later on dynamically allocated based on the sysconf (_SC_LOGIN_NAME_MAX).
2. Does your program use the obsolescent interface cuserid () and the macro L_cuserid?
    - Ø This interface is not enhanced for long user/group name. Use the appropriate replacement interface by referring to the man page of cuserid ().
3. Insufficient storage
    - Are any array buffers allocated to store a name, or a string that might contain a name? Are they large enough? Trace through the code and identify where that buffer is used and how it might impact the code.
    - Are there any structures defined that contain an array for a name? Are they large enough? Are they passed back to a user application?
    - Are there any malloced memory regions used to store a name (or names)? Are they large enough? Where is that memory accessed in other locations of the code?
        - Ø Allocate sufficient storage, and make sure the right pointer is passed.
4. Are there any message strings (such as in a message catalog) that contain a name? Does the code assume the output does not exceed 80 bytes (to fit in a terminal window.)?
    - Ø Review your messages and message catalogs.
    - Ø Verify whether the change in output can cause readability or other display problems.
5. Are any file names created using a name that could exceed the limit of a legal file name length on the file system?
    - Note that different file systems types may have different limits (HFS supports 14 and 250 bytes.)
    - Are any characters appended to a name, for example to create a temporary file name by appending a random numeric value at the end?

        - Ø Make sure the file system limits does not create problems with longer user /group name, by either truncating the name if it is OK, or designing a new hashing algorithm to generate a unique file name, instead of using the name itself.

6. Are there places where names are collected into a fixed size buffer? An example would be such as members defined in /etc/group are limited to LINE_MAX bytes long. If so, how does increasing the size of a name impact the code?
    - Ø If there is an impact, and the buffer cannot be increased, document the limitation. Else use a different algorithm to handle longer buffer.
7. Does the code call any of getpw* or getgr* routines? Where is the name portion of the passwd_t structure used throughout the code? Does the code handle any size name that is passed back from those routines?
    - Ø Make sure sufficient storage is allocated to handle any size name.
8. Does the code call any of the utmp or utmpx routines?
    - Ø They must be changed to call utmps.
9. Does the code read directly from any file that contains a user name? For example, instead of using getuts* APIs directly reading /etc/utmps file
    - Ø If so, switch to APIs if possible. If not, bring this to the notice of HP.
10. Does your code parse output from other commands, such as ls -l , ps –ef , who? Does it assume that fields in the output will appear in certain character (column) locations?
    - Ø If so, set the appropriate display width to get the required output from the command.

5.2 Other Considerations

5.2.1 Dependency on the system state: Expanded user/group name enabled or not

Ideally, all the APIs and utilities must be independent of the length of user/group names. However some applications may need to understand the state of the system – (whether it is expanded user/group name enabled or not)

- To allocate the required storage at run time based on the length.
- To operate correctly. For example, the useradd should not allow long user/group names unless system is enabled for it.

You must determine whether your application needs to know the state of the system.
If your application needs to know the state of the system then sysconf(_SC_LOGIN_NAME_MAX) can be used. If it returns -1 or 64, then the system is not enabled with long user/group name. If it returns 255, then the system is enabled with long user/group name.

6 Test Guidelines

In addition to the existing tests that exercise the user/group name functionality, testing should be primarily focused on the maximum capacity situations. HP recommends that the passwd/group setup is done as part of installing the test tools using a chrooted environment and not to modify the system files. This helps in avoiding removal/copy of these files during the test run, and reduces the risk of leaving the system in an unstable state if it crashes in the middle of a test run.

6.1 User/Group Database Configuration

Configure system to enable expanded user/group name and test with a new passwd database. This can be integrated into the test environments (potentially under a chrooted environment. In general, it is sufficient for application developers to test the code with just one backend database because the code path exercised in the application is the same irrespective of the backend database. However, some of the applications may have the need to get tested with some other backend database as well (for example, NSS password backend).

6.2 Display Functionality

In a long user/group environment, the output of some of the commands and utilities change according to the display guidelines. These utilities require the following three types of testing:

- No configurations, default mode

- System-wide configuration (lugadmin)
- User level configuration(environment variable)

6.3 Buffer Overflow tests

Ensure that there are test cases that validate buffer overflow conditions. This ensures that there are no security vulnerabilities, especially in setuid programs.

7.0 Summary

HP has provided expanded user and group names in response to the needs expressed by customers. This feature increases the maximum user and group name lengths from eight and 16 bytes (respectively) to 255. The system commands and utilities are enhanced to handle long user/group names. However, longer names must be used with caution. Recompiling the programs alone does not make the program expanded user/group name clean .No FLV mechanism exists because it is not required for any of the APIs that return username/groupname. The APIs are scalable to handle the expanded username/groupname. The programs that are not enhanced to accommodate the expanded name interfaces truncate them. This can cause programs to exhibit ambiguous output or incorrect behavior (including unexpected and undetermined aborts). The default system and product configuration settings limit names to the original, compatible, maximums in order to avoid these problems. The system administrator must explicitly enable the expanded name capability through a command.

This paper has described how to enable the capability, limitations of specific HP products, how software developers can determine whether their products have dependencies on user and group name sizes and how to enhance the programs to accommodate long names, and how to test them.

Glossary of Terms

Application Programming Interface (API)

An application interface is a data structure definition or type, a program function or procedure, a data variable, or a constant (symbolic or literal) which is exported by a component (for example, a library) for use by application code. Interfaces appear in header files and are (usually) documented in the API specification (for example, the man pages).

Byte

A byte is an eight-bit quantity. In the ASCII character set, each character can be represented in a single byte. While HP-UX currently supports only ASCII characters for user and group names the word byte (rather than character) is used in reference to those values.

Long User/ Group Name

This refers to user or group names that are longer (i.e., have a greater string length) than the default maximums of eight or 16 bytes, respectively.

LOGIN_NAME_MAX

A symbolic constant parameter that defines the maximum length of user name strings. Its value is 64 bytes. This should no longer be used. This symbolic constant will be obsoleted in future releases.

_POSIX_LOGIN_NAME_MAX

This is a symbolic name for the most restrictive (minimum value) value a conforming implementation should support, which is specified to be nine bytes.

Manufacturing Part Number : 5992-4118

4AA0-XXXXENW, May 2007