



Group membership expansion: guidelines for deployment

Executive summary.....	2
Groups background.....	2
Group membership expansion.....	2
ngroups_max tunable.....	2
Dynamic tuning.....	2
Deployment scenario.....	3
Installing group membership expansion.....	3
Updating the HP-UX core.....	3
Updating system utilities.....	4
Updating applications.....	4
Known limitations.....	4
Performance considerations.....	5
Summary.....	5
Appendix: Enabling applications for groups expansion.....	5
Terminology.....	5
Correct handling of supplementary group ID lists.....	6
Incorrect handling of supplementary group ID lists.....	7
For more information.....	7

Executive summary

The maximum number of supplementary groups that can be associated with a user or process has been made a tunable parameter in Update 3 to HP-UX 11i v3. Previously, the maximum limit was fixed by the constant `NGROUPS`, whose value is 20. The system administrator can adjust the limit to values higher than the previous maximum by invoking the `kctune` command to increase the tunable `ngroups_max`. This enhancement allows users—and processes running on their behalf—to gain access permissions based on their membership in an expanded number of groups.

Groups background

Every HP-UX user belongs to one primary group and a variable number of supplementary groups. Group membership can be established by the `useradd`, `usermod`, and `groupmod` commands, and is reflected in the files `/etc/passwd` and `/etc/group`. A user's login process and its child processes have the same group membership as the user.

Every HP-UX file is owned by a group and has group-level permissions associated with it. Users and processes running on behalf of users can be granted access to files based on their membership in the various groups.

The HP-UX kernel makes information about group membership available to applications, which can then use that information in a variety of ways—for example, to implement their own security and access policies.

Group membership expansion

While the number of groups listed for a user in `/etc/group` is limited only by the size of that file, the HP-UX kernel has a maximum limit on the number of those groups that are actually used to determine effective group membership. Historically, the maximum limit on effective group membership has been determined by the constant `NGROUPS` (for BSD) or `NGROUPS_MAX` (for POSIX). On HP-UX, both constants have a value of 20, and that has not been changed by this enhancement. The POSIX standard permits the limit to be increased at run time, so the group membership expansion enhancement created the tunable `ngroups_max` to establish the maximum limit on the effective group membership for a user or process.

The effective group list is used to determine file access permissions and is made available to applications through the groups APIs `getgroups` and `setgroups`.

`ngroups_max` tunable

The default value of the `ngroups_max` tunable is 20. The group membership expansion enhancement has absolutely no effect as long as `ngroups_max` remains at its default value.

The minimum value to which `ngroups_max` can be tuned is also 20. This guarantees backward compatibility with all previous applications and system configurations.

The maximum value to which `ngroups_max` can be tuned is 65536.

Dynamic tuning

The `ngroups_max` tunable is dynamic: changes do not require a reboot. Any change takes effect immediately for new user logins. However, standards conformance requires that a running process not experience a change in the maximum number of groups value during its lifetime. Specifically, this means that when a process is made aware of the value of the maximum number of groups, that value

is then frozen for the lifetime of the process. This value is inherited across `exec()` calls, but not across `fork()` calls.

Deployment scenario

The following example shows how group membership expansion is useful in a business context. Suppose that a company has many sales offices, each having a number of sales representatives. The representatives within an office share data among themselves and protect it from other users based on group membership. Each of the sales representatives has a user account that is a member of a group named after the representatives' sales office.

If Jones, Lewis, and Dhali worked out of a Milwaukee sales office of a company with additional offices in Chicago and Madison, a portion of the `/etc/group` file might look like this:

Figure 1. Excerpt from the `/etc/group` file

```
chicago::185:smith,rolls,chang,burke,kafer,zenda,jbkaiser
milwaukee::186:jones,lewis,dhali,jbkaiser
madison::187:bucky,lenin,fiske,carty,jbkaiser
```

File permissions are established so that the file owner and members of the group can read and write the files, but others are excluded from access. A listing of the files in a directory following this scheme might look like this:

Figure 2. Excerpt from a directory listing

```
-rw-rw---- 1 smith chicago 92160 May 1 11:00 plist.chi
-rw-rw---- 1 bucky madison 51200 May 2 13:18 plist.mad
-rw-rw---- 1 jones milwaukee 181280 May 2 13:07 plist.mil
```

The vice president, Kaiser, could gain access to all of the files belonging to all of the sales offices by becoming a member of all of the groups. In a large company, the vice president might need to manage several dozen sales offices.

There are other file-sharing and protection models along this line where the limit of 20 groups per user is too restrictive.

Installing group membership expansion

The code to enable the group membership expansion enhancement is available as a set of patches in Update 3 to HP-UX 11i v3.

Updating the HP-UX core

All of the files needed to update the HP-UX core for groups expansion are installed as a consequence of installing the patch `PHKL_38095`. A reboot is required following installation. Installing the

enhancement does not change system behavior in any way. It is necessary to increase the `ngroups_max` tunable above its default value of 20 to change the system behavior.

Updating system utilities

A number of system utilities are sensitive to the maximum number of groups per user, and so must be updated before they will recognize that a user or process is a member of more than 20 groups. Because these utilities are not part of the core, they are not installed along with `PHKL_38095`. Table 1 shows the utilities that must be updated if they are to be used in an expanded groups environment:

Table 1. Utilities requiring update before use with expanded groups

Utility	Minimum required version	Included in Update 3?
sendmail	C.8.13.3.2	YES
FTP	C.2.6.1.4.0	YES
ONC+	B.11.31.04	YES
LDAP-UX	B.04.20	NO; available via Web release (Software Depot) by December 2008

The utilities RCS and SCCS have not yet been updated to work with group membership expansion. They will not work properly for users belonging to more than 20 groups until such time as patches are created by HP.

Updating applications

Most applications are not sensitive to the maximum number of groups per user, but some specialized applications may track the group membership of users and processes. If such applications assume that the maximum number of groups is fixed at 20, they may not perform properly in an expanded groups environment: they may terminate unexpectedly, or fail to recognize that a user has any supplementary groups at all. In the appendix are some details that will help application developers to write programs that work with an expanded number of groups.

HP has not validated that all third-party applications work properly with more than 20 groups. For that reason, the `kctune` command displays a warning message when `ngroups_max` is tuned to a value greater than 20.

Known limitations

The ONC RPC over-the-wire protocol used by all ONC RPC applications, including NFS, limits the number of supplementary group IDs recognized across NFS filesystems to 16.

Network Information Service (NIS) builds a database that contains information about user group membership, called `netid`. NIS will include only the first 20 supplementary user groups in this database. NIS is usually used in conjunction with ONC RPC, where only the first 16 groups are significant, so this is of little practical consequence.

The records in the netid database are limited to 1,024 characters. The use of extremely long group names can cause the 1,024-character limit to be exceeded even with just 20 groups, and in that case, such long records will not be included in the database.

Performance considerations

Some HP-UX commands—for example, `groups` and `id`—complete in an amount of time proportional to the number of groups to which a user or process belongs. The additional time required to complete such commands can be noticeable if a user is a member of an extremely large number of groups. While HP-UX has the flexibility to allow customers to place a user in as many as 65,536 groups, such a high limit may make some operations take longer than is acceptable to interactive users.

The performance of these commands depends on the actual number of groups to which the user or process belongs, not the value of the `ngroups_max` tunable. In this sense, there is no penalty to setting the tunable to its maximum value. However, the login process can require memory space proportional to `ngroups_max`, so the tunable should not be set to an unnecessarily large value. The guideline is to choose a value modestly higher than the planned greatest number of groups to which any user will belong.

Some clustered applications are sensitive to the group membership of users and processes. A clustered filesystem fits into this category. In such cases, it is best if each node in the cluster has the same value for the maximum number of supplementary groups per user.

Summary

The group membership expansion enhancement to HP-UX 11i v3 Update 3 changes the maximum number of groups for a user or process, making it a tunable parameter. This change allows customers to implement models for file access and protection that were not possible under the previous limit of 20 groups per user. Before deploying such models, customers must update the HP-UX kernel and utilities, and must verify that their local applications are compatible with the expanded group limit.

Appendix: Enabling applications for groups expansion

Terminology

NGROUPS

`NGROUPS` is an obsolete compile-time constant that may still be in use by some older applications. Its value (20) is the same as the `NGROUPS_MAX` compile-time constant described below.

NGROUPS_MAX (compile-time limit)

The `<limits.h>` header file includes a compile time definition for `NGROUPS_MAX` whose value remains 20. The Unix2003 standard (and its predecessors) describe `NGROUPS_MAX` as a “Runtime Increaseable Value”—that is, the actual limit on an instance of HP-UX may be greater than the value of the compile-time constant. All instances of HP-UX support membership in at least 20 supplementary groups. With group membership expansion, HP-UX instances may support more than 20.

ngroups_max

`ngroups_max` is a new kernel tunable parameter specifying the maximum number of supplementary group IDs that may be associated with a user or process. Prior to the group membership expansion, this maximum number was fixed at the value of the compile-time `NGROUPS_MAX` constant. With the group membership expansion, the maximum can now be tuned as high as 65,536. The default value for `ngroups_max` is 20, and so is the minimum value.

Note 1

Setting the `ngroups_max` tunable to a value higher than 20 can cause applications that assume the old fixed limit to fail.

Note 2

The `ngroups_max` tunable is dynamic. Increases to the tunable take effect immediately for new logins, but do not affect users who are already logged in or processes that are already running. It is also possible to decrease the tunable to values lower than the number of groups associated with existing users. Users associated with more groups than are permitted by a new lower limit will note, on their next login, that some of those groups are missing from their set of supplementary group IDs. File accesses and other operations that depend on the missing groups will no longer work.

NGROUPS_MAX, _SC_NGROUPS_MAX (run-time limit)

The `NGROUPS_MAX` system variable is the same value as that of the `ngroups_max` system tunable parameter, but it will be frozen for the life of a process or session once read. The value may be obtained via the function `sysconf(_SC_NGROUPS_MAX)` or the command `getconf NGROUPS_MAX`. The act of reading the value freezes the value associated with the calling process if it was not already frozen. This value is copied on `exec(2)` but not `fork(2)`. Children of `fork` will be limited to the greater of the `ngroups_max` tunable or the number of groups to which they currently belong.

Correct handling of supplementary group ID lists

This section describes how correct programs handle supplementary group ID lists, from the most common to the least common use cases.

Use `getgroups(0, NULL)`

Programs that need the number of supplementary group IDs currently associated with the calling process should use `getgroups(0, NULL)` to determine the answer. Most processes in a typical HP-UX environment will belong to only a few groups.

Use dynamic allocation for supplementary group ID lists

Once the number of supplementary group IDs associated with a process has been determined, space to store that many `gid_t` values must be dynamically allocated with `malloc()` or a similar facility. The number of groups you have space for and the pointer to the dynamically allocated space must be passed to the `getgroups(2)` call so that it can fill in the list of supplementary group IDs currently associated with the calling process. `getgroups(2)` will return `EINVAL` if the space allocated is insufficient to hold all the supplementary group IDs associated with the calling process.

Use `initgroups(3)`

Privileged applications that need to initialize the list of supplementary group IDs associated with the calling process should use `initgroups(3)`.

Use `sysconf(_SC_NGROUPS_MAX)`

The `sysconf(_SC_NGROUPS_MAX)` call should be used to determine the maximum number of supplementary group IDs that may be associated with the current process. The value returned will never be smaller than the `NGROUPS_MAX` compile-time constant, but it may be larger.

Privileged applications that must directly set the supplementary group IDs associated with a process must be designed to work within the limit returned by this call. Such applications may rely on this value to be at least `NGROUPS_MAX(20)` on any HP-UX 11i system.

Use `NGROUPS_MAX` compile-time constant (only when portability is required)

A program that must run on any instance of HP-UX, even one without the group membership expansion enhancement, must be designed such that it does not rely on the ability to associate more than 20 supplementary groups with a given user or process.

Incorrect handling of supplementary group ID lists

This section describes how some programs' built-in assumptions about supplementary group ID limits can prevent them from working with expanded groups.

Use of `NGROUPS` compile-time constant

The `NGROUPS` compile-time constant is obsolete and should never be used.

Use of `NGROUPS_MAX` compile-time constant when retrieving supplementary group IDs

A process may have more supplementary group IDs associated with it than are specified by the compile-time constant `NGROUPS_MAX`. The `NGROUPS_MAX` compile-time constant should not be used to size storage to be filled in by the `getgroups(2)` system call, or as the iteration limit when searching the array of supplementary group IDs filled in by the `getgroups(2)` system call.

Use of `NGROUPS_MAX` compile-time constant when setting supplementary group IDs

Privileged applications setting the list of supplementary group IDs with `setgroups(2)` must not use `NGROUPS_MAX` to limit the number of supplementary group IDs they set.

For more information

For additional information see the following HP-UX man pages: `getgroups(2)`, `setgroups(2)`, `initgroups(3)`, `ngroups_max(5)`, `limits(5)`.

To help us improve our documents, please provide feedback at: www.hp.com/solutions/feedback

Technology for better business outcomes

© Copyright 2008 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

UNIX is a registered trademark of The Open Group.

4AA2-1738ENW, August 2008

