

USERS AND GROUPS PROGRAMMING INTERFACE

UGLibrary is a set of functions that lets you implement users and groups capabilities for your shared applications in a consistent way. Thus, you can take advantage of existing system-software users and groups components to authenticate users of your shared applications.

Additionally, your applications can use UGLibrary to manipulate the Users & Groups Data File for the AppleShare Server 3.0, or for any Macintosh that is using Macintosh File Sharing. These functions provide a simple programming interface that allows your applications to add and delete users and groups, and to set information on users, groups, the AppleShare File Server 3.0, or Macintosh File Sharing. (Throughout the rest of this document, the term file server is used to refer to both the AppleShare File Server 3.0 and Macintosh File Sharing.)

Here are some examples of how UGLibrary can help you make use of users and groups files from within your applications:

- The UGAuthenticateUser function can let server applications other than the file server use the users and groups file to authenticate the applications' remote users. (The password must be a Pascal string containing the Cleartext value of the password. the application must use the appropriate encryption methods if the password will be sent over the network.)

- A database application can use the UGGetUserInfo function to see if a remote user is a member of a specific group before allowing the user access to files.

- Classroom software can generate users and groups files for file servers, using its student and class database.

- An application, running on a server Macintosh, that is communicating with a remote application can be used to maintain the users and groups files remotely. Apple events or the PPC Toolbox can be used for authenticated communications between the applications.

The users and groups file

A users and groups file contains records that define file server characteristics, the list of users, the list of groups, the mapping of users into groups, and other information about users. Users and groups files can be opened and closed. When a users and groups file is opened, a file reference number is assigned to the access path and a program access ID is assigned to the application that opened the file. The file reference number and program access ID are passed to all UGLibrary functions that manipulate the open users and groups file.

When your program creates a users and groups file, the file's server version number is Sys7SrvrVer. (This is the value that the UGGetULInfo function returns in the ugSrvrVersion field.) This version number indicates that the file can be used only by Macintosh File Sharing. If the file is to be used with the AppleShare File Server 3.0, the ugSrvrVersion field must be set to AS3SrvrVer by means of the UGSetULInfo function.

The following UGLibrary functions are relevant to users and groups files and are described in detail later in this document:

- UGOpenFile
- UGCloseFile
- UGCreateFile

Users

The following attributes are assigned to each user (except the guest user): a user ID, a user name, a password, and user flags. These attributes are assigned when the user is created with the UGNewUser function. The user ID cannot be changed. The user name can be changed with the UGRenameUser function.

Note UGLibrary does not allow user or group names to contain the colon (:) character. User or group names can contain the slash (/) character, but you should avoid using such names because they are not compatible with the Apple II family of computers.

The password and user flags can be read or changed with the UGGetUserInfo or UGSetUserInfo function, respectively. There are five user flags:

- | | |
|--------------|---|
| IACEnabled | Set if interapplication communication (IAC) is enabled for the user. |
| LoginEnabled | Set if file server login is enabled for the user. |
| SuperUser | Set if the user is a system administrator user (that is, the system owner, known in AppleShare terminology as the superuser). Only one superuser is allowed under Macintosh File Sharing. |
| NoChangePwd | Set if the user cannot change his or her password from a remote workstation. |
| GetNewPwd | Set if the file server should demand a new password from the user at the next login. This flag is available only on the AppleShare File Server 3.0. This flag must be clear for Macintosh File Sharing. |

Two kinds of users are treated differently than other users: the guest user and the system administrator user. The Guest attributes are maintained as part of the server characteristics and can be read or set with the UGGetULInfo or UGSetULInfo functions, respectively. The record with ID 0 contains this information and is automatically created along with a new users and groups file by the UGCreateFile function.

The administrator user's user ID is always 1. You can use either of two methods to set up the administrator user: You can have your application create the administrator user with the UGNewUser function, or you can let the file server create the administrator user for you. The first time either Macintosh File Sharing or the AppleShare Admin program accesses a users and groups file, it checks for the administrator user. If the administrator user is not found, it is created and given the "owner name" found in the 'STR ' resource ID -16096 (the name of the person who "owns" the computer), which resides in the System file. If the administrator user is found and its name does not match the owner name in the System file, then the name of the administrator user is put into the 'STR ' resource ID -16096 so that all

parts of the system use the same administrator name. Once the administrator user is created, it cannot be deleted or renamed by means of UGLibrary; it can only be renamed through the System 7 Sharing Setup control panel or by means of AppleShare Admin.

In addition to returning attribute information on users, the UGGetUserInfo function can be used to enumerate all users in the users and groups file or all users in a group, to provide a list of all groups of which a user is a member, and to get a user's primary group.

The UGDeleteUser function can be used to delete all users except the guest user and the administrator user. However, the guest user and administrator user can be disabled with UGSetULInfo and UGSetUserInfo, respectively.

The UGAuthenticateUser function provides a way to authenticate a user's password. UGLibrary does not provide a way to read user passwords.

The following UGLibrary functions are relevant to users and are described in detail later in this document:

- UGNewUser
- UGDeleteUser
- UGRenameUser
- UGGetUserInfo
- UGSetUserInfo
- UGAuthenticateUser

Groups

Each group has two attributes, its group ID and its name, which are assigned when the group is created with UGNewGroup. The group ID cannot be changed. The group name can be changed with UGRenameGroup.

The UGGetGroupInfo function can be used to enumerate all groups in the users and groups file, and to map a group name to a group ID or map a group ID to a group name. Groups can be deleted with the UGDeleteGroup function, and renamed with the UGRenameGroup function.

You use the UGAssignUserToGroup function to make a user a member of a group. The UGAssignUserToGroup function also lets you assign a user's primary group. (The primary group is used by the AppleShare File Server 3.0 when Apple II mode is on. The primary group is also the group assigned access privileges to a folder when a folder is created remotely.) The UGDeleteUserFromGroup function is used to remove a user from a group.

Note UGLibrary does not allow user or group names to contain the colon (:) character. User or group names can contain the slash (/) character, but you should avoid using such names because they are not compatible with the Apple II family of computers.

The following UGLibrary functions are relevant to groups and are described in detail later in this document:

- UGNewGroup
- UGDeleteGroup
- UGRenameGroup
- UGGetGroupInfo
- UGAssignUserToGroup
- UGDeleteUserFromGroup

The file server

The UGGetULInfo function is used to retrieve six file server attributes: the file server name, the file server version, the minimum password length, the maximum number of failed login attempts, the number of days before user passwords expire, and server flags. The file server name cannot be changed by means of UGLibrary. The name comes from the 'STR ' resource ID -16413 (the computer name), located in the System file, and is initialized by the System 7 Sharing Setup control panel or by the AppleShare Admin program. The file server version is the version number of the users and groups file; a value of Sys7SrvrVer (3) indicates Macintosh File Sharing and a value of AS3SrvrVer (4) indicates the AppleShare File Server 3.0. The minimum password length, the maximum number of failed login attempts, the number of days before user passwords expire, the server flags, and the server version can all be changed with the UGSetULInfo function.

Note When your program creates a new users and groups file by using the UGCreateFile function, the file does not contain a server name. However, when this file is made the current users and groups file (that is, when the old users and groups file is dragged out of the Preferences folder, the new file is moved in, and the computer is restarted), the Macintosh name is automatically inserted into the file as the server name.

There are five server flags:

Apple2	Set if the server is in Apple II mode. This flag is used for the AppleShare File Server 3.0 only; it must be clear for Macintosh File Sharing.
NoGuests	Set if guests cannot log on to this server.
IACGuestEnabled	Set if IAC is turned on for guest.
IACGlobalEnabled	Set if IAC is turned on for this computer.
DontSavePwd	Set if the AppleShare workstation should not allow the user to save his or her password by means of the Chooser. This flag is available on the AppleShare File Server 3.0 only, it must be clear for Macintosh File Sharing.

The UGSetULInfo function is used to set six file server attributes: the administrator password, the minimum password length, the maximum number of failed login attempts, the number of days before user passwords expire, the server flags, and the server version. All server flags set with UGSetULInfo are effective as soon as the file server is started, except for IACGlobalEnabled, which is effective only after the computer is restarted.

Note For Apple II users, AppleShare Admin creates a personal folder for each user inside the Users folder. Thus, even if the Apple2 bit is set, the Startup menu (in AppleShare Admin) must be used for all Apple II users. (For more information about the contents of the Users folder, see the AppleShare Programmer's Guide for the Apple II.)

The following UGLibrary functions are relevant to file servers and are described in detail later in this document:

- UGGetULInfo
- UGSetULInfo

Using UGLibrary

Four files are provided to allow programs to use the library functions:

- UGLibrary.o contains the object code for all functions and must be linked with programs that call the library functions.
- UGLibrary.p contains the data structures, function declarations, and error codes returned by the functions. This file must be included in Pascal programs.
- UGLibrary.h contains the data structures and function declarations and must be included in C programs.
- UGLibErrors.h contains the error codes returned by the functions and must be included in C programs.

All functions are passed two parameters: UserGroupPBPtr, which is used for passing all the input values for the function and also to pass back the output values, and async, a Boolean value used to indicate whether the function is called synchronously or asynchronously. All functions in version 1.0 of UGLibrary must be called synchronously. Calling a function asynchronously will result in a ugBadParamErr error.

You should be aware of the following restrictions in the use of UGLibrary version 1.0:

- The UGLibrary functions work only with users and groups files on local volumes.
- Functions must be called synchronously.
- While the file server or AppleShare Admin program is active, your program cannot use UGLibrary functions that write to the users and groups file used by the file server or the AppleShare Admin program. For your program to write to a users and groups file, either the file server (or AppleShare Admin) must be shut down, or a copy of the file must be made before the file server (or AppleShare Admin) is started, so that your program can work with the copy.

!! IMPORTANT Use of the UGLibrary functions in any way other than that described in this document will produce undefined results. You should take every precaution to avoid corrupting the users and groups file. It is highly recommended that you make a copy of the users and groups file, and call the functions only on the copy. !!

Macintosh File Sharing supports a combined total of 100 users and groups. Several features supported by the AppleShare File Server 3.0 are not available under Macintosh File Sharing:

- minimum password length
- password expiration
- maximum failed login attempts
- administrator password
- multiple superusers
- the GetNewPwd user flag
- the Apple2 server flag
- the DontSavePwd server flag
- the primary group flag

When using UGLibrary functions that set these features with a Macintosh File Sharing users and groups file, you should clear the unused flags; use a value of 0 for minimum password length, password expiration, and maximum failed login attempts; and use a zero-length Str8 for the administrator password.

UGLibrary functions

This section describes each of the UGLibrary functions in detail. For each function, the Pascal and C declarations are shown, the parameter block for the function is described, the function is described, and all possible function result codes are listed. The section "Data Structures," later in this document, defines all parameter blocks used by these functions. The subsequent section, "Function Summary," specifies the parameter block used by each function.

UGOpenFile

```
FUNCTION UGOpenFile (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;
```

```
pascal OSErr UGOpenFile (UserGroupPBPtr thePB, Boolean async);
```

Parameter Block	18	ugNamePtr	long	pointer to filename
	22	ugVRefNum	word	volume specification
	24	ugRefNum	word	file reference number
	26	ugDirID	long	parent directory ID
	30	ugAccessID	long	program access ID
	42	ugAdminKey	long	pointer to administrator password

UGOpenFile opens the users and groups file specified in ugNamePtr (a pointer

to a Str31) within the folder specified in ugDirID on the volume specified in ugVRefNum. If ugNamePtr points to a zero-length Str31, the localized name of the current users and groups file is used. (For example, the localized name of the current users and groups file under U.S. system software version 7.0 is "Users & Groups Data File.") Otherwise, the name of the file to which ugNamePtr points is used. The administrator password is passed in ugAdminKey (a pointer to a Str8); the function verifies that the password is correct before returning the values of the ugRefNum and ugAccessID fields. The ugRefNum field identifies the file being used, and the ugAccessID field identifies the program using the file. These two fields are used by all other UGLibrary functions except UGCloseFile and UGCreateFile. The value of the ugAdminKey field should be a pointer to a zero-length Str8 if the file is used by Macintosh File Sharing.

Result Codes

noErr	0	No error.
ugULInfoRecNotCreated	3	There is a problem with the users and groups file. Delete it and re-create it.
ugBadParamErr	13	async is TRUE, the length of the filename is greater than 31, or the length of the administrator password is greater than 8.
ugNotBtree	18	Users and groups file is not a B-Tree file.
ugInvalidPasswd	22	ugAdminKey is not the correct password.
ugDefNameErr	98	Default name of users and groups file could not be found.
ugFatalError	99	There is a problem with the users and it.

UGCloseFile

FUNCTION UGCloseFile (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

pascal OSErr UGCloseFile (UserGroupPBPtr thePB, Boolean async);

Parameter Block 24 ugRefNum word file reference number

UGCloseFile closes the open users and groups file identified by ugRefNum.

Result Codes

noErr	0	No error.
ugBadParamErr	13	async is TRUE.
ugNotBtree	18	Users and groups file is not a B-Tree file.
ugFatalError	99	There is a problem with the users and groups file. Delete it and re-create it.

UGCreateFile

FUNCTION UGCreateFile (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

pascal OSErr UGCreateFile (UserGroupPBPtr thePB, Boolean async);

Parameter Block	18	ugNamePtr	long	pointer to filename
	22	ugVRefNum	word	volume specification
	26	ugDirID	long	parent directory ID

UGCreateFile creates an empty users and groups file with the name specified

by ugNamePtr (a pointer to a Str31) within the folder specified in ugDirID on the volume specified in ugVRefNum. If ugNamePtr points to a zero-length Str31, the localized name of the current users and groups file is used. (For example, the localized name of the current users and groups file under U.S. system software version 7.0 is

"Users & Groups Data File.") Otherwise, the name of the file to which ugNamePtr points is used.

Result Codes

noErr	0	No error.
ugBadParamErr	13	async is TRUE or the length of the filename is greater than 31.
ugDskFull	20	No space on disk.
ugDefNameErr	98	Default name of users and groups file could not be found.
ugFatalError	99	There is a problem with the users and groups file. Delete it and re-create it.

UGNewUser

FUNCTION UGNewUser (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

pascal OSErr UGNewUser (UserGroupPBPtr thePB, Boolean async);

Parameter Block	18	ugNamePtr	long	pointer to user name
	24	ugRefNum	word	file reference number
	30	ugAccessID	long	program access ID
	34	ugUserID	long	user ID
	50	ugPassword	long	pointer to user password
	54	ugUserFlags	word	user flags

UGNewUser creates a new user. The pointer ugNamePtr (a pointer to a Str31) contains the name of the user and the pointer ugPassword (a pointer to a Str8) contains the password. The new user is created with the characteristics passed in ugUserFlags (a long; see "Data Structures," later in this document, for details). To create the administrator user, set ugUserID to 1. To create a regular user, set ugUserID to 0. The user ID of the user created is passed back in ugUserID.

Result Codes

noErr	0	No error.
ugUserAlreadyExists	1	User with the same name exists or an attempt was made to create the administrator user and the administrator user already exists.
ugGroupAlreadyExists	2	Group with the same name exists.
ugNoMoreUGs	4	The maximum value (\$FFFFFFFF) has been reached.
ugBadParamErr	13	async is TRUE, the length of the user name is less than 1 or greater than 31, the user name contains a colon, the length of the user password is greater than 8, or ugUserID is not 0 or 1.
ugServerRunning	16	Server is running.
ugAdminRunning	17	Admin is running.
ugNotBtree	18	Users and groups file is not a B-Tree file.
ugDskFull	20	No space on disk.
ugBadUID	21	ugAccessID is wrong. The users and groups file may not be open.
ugFatalError	99	There is a problem with the users and groups file. Delete it and re-create it.

UGDeleteUser

FUNCTION UGDeleteUser (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

pascal OSErr UGDeleteUser (UserGroupPBPtr thePB, Boolean async);

Parameter Block number	24	ugRefNum	word	file reference
	30	ugAccessID	long	program access ID
	34	ugUserID	long	user ID

UGDeleteUser deletes the user specified by ugUserID. The administrator user

(user ID 1) cannot be deleted.

Result Codes

noErr	0	No error.
ugUserNotFound	5	User is not in the users and groups file.
ugNotAUser	7	ugUserID is not the ID of a user.
ugBadParamErr	13	async is TRUE.
ugServerRunning	16	Server is running.
ugAdminRunning	17	Admin is running.
ugNotBtree	18	Users and groups file is not a B-Tree file
ugBadUID	21	ugAccessID is wrong. The users and groups file may not be open.
ugFatalError	99	There is a problem with the users and groups file. Delete it and re-create it.

UGRenameUser

```
FUNCTION UGRenameUser (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;
```

```
pascal OSErr UGRenameUser (UserGroupPBPtr thePB, Boolean async);
```

Parameter Block	18	ugNamePtr	long	pointer to current user name
	24	ugRefNum	word	file reference number
	30	ugAccessID	long	program access ID
	34	ugUserID	long	user ID
	44	ugMisc	long	pointer to new user name

UGRenameUser can be used to change the name of any user except the administrator user. The user can be specified either by the user's ID in ugUserID or by the name

in ugNamePtr (a pointer to a Str31). If ugUserID is used, then ugNamePtr should be a pointer to a zero-length Str31. If ugNamePtr is used, then ugUserID should be 0. If both fields contain values, then ugUserID takes precedence. The new name is specified by ugMisc (a pointer to a Str31).

Result Codes

noErr	0	No error.
ugUserAlreadyExists	1	User with the same name exists.

ugGroupAlreadyExists	2	Group with the same name exists.
ugUserNotFound	5	The user is not in the users and groups file.
ugNotAUser	7	ugUserID is not the ID of a user.
ugBadParamErr	13	async is TRUE, the length of the old user name is greater than 31, the length of the new user name is less than 1 or greater than 31, the new user name contains a colon, or the old user name is the owner's name.
ugServerRunning	16	Server is running.
ugAdminRunning	17	Admin is running.
ugNotBtree	18	Users and groups file is not a B-Tree file.
ugBadUID	21	The ID in ugAccessID is wrong. The users and groups file may not be open.
ugFatalError	99	There is a problem with the users and groups file. Delete it and re-create it.

UGGetUserInfo

FUNCTION UGGetUserInfo (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

pascal OSErr UGGetUserInfo (UserGroupPBPtr thePB, Boolean async);

Parameter Block	18	ugNamePtr	long	pointer to user name
	24	ugRefNum	word	file reference number
	30	ugAccessID	long	program access ID
	34	ugUserID	long	user ID
	38	ugGroupID	long	group ID or 0 for all
	42	ugPosOffset	word	position offset
	54	ugUserFlags	long	user flags
	58	ugNumBadLogins	word	number of failed login attempts
	60	ugPwdDate	long	date of last password change
	64	ugNumGroups	word	number of groups

66 ugGroups 168 bytes array of group IDs
(42 longs)

UGGetUserInfo can be used to provide information about a specific user, enumerate all users in the file, or enumerate all users in a group.

In all three cases, the following information about the user is returned: ugUserFlags, the user's settings; ugNumBadLogins, the number of failed login attempts; ugPwdDate, the password creation date in seconds; ugNumGroups, the number of groups the user belongs to; ugGroups (an array of 42 longwords), the group IDs of the groups to which the user belongs; ugUserID; and ugNamePtr (a pointer to a Str31).

If ugNumGroups is 1, then either the first or the second element in ugGroups contains a group ID. The first element in the ugGroups field is reserved for the primary group and contains the group ID if the user has a primary group; otherwise it contains zero.

In the case of a specific user, the user is specified either by ugUserID or by ugNamePtr. If ugUserID is used, then ugNamePtr should be a pointer to a zero-length Str31. If ugNamePtr is used, then ugUserID should be 0. If both fields have values, then ugUserID takes precedence. ugGroupID should always be set to 0.

If ugPosOffset is 0, then the information passed back will be that of the user specified. If ugPosOffset is 1, then the information passed back will be that of the user following the one specified. If ugPosOffset is neither 0 nor 1, then the ugBadParamErr error will be returned.

To have UGGetUserInfo enumerate all the users, your program should set ugGroupID, ugUserID, and ugPosOffset to 0, and ugNamePtr should be a pointer to a zero-length Str31. When the function is called, the first user record will be returned (if present). Then ugPosOffset should be set to 1 and the function called repeatedly. Each time it is called, the function will return information on the next user, until a ugNoMoreUsers or ugEOF error is returned.

To have UGGetUserInfo enumerate all users in a group, your program should set ugGroupID to the group ID, and ugUserID and ugPosOffset to 0; ugNamePtr should be a pointer to a zero-length Str31. When the function is called, the first user record within that group will be returned, if present. If ugPosOffset is set to 1, the record of the user after ugUserID within that group is returned. This process is repeated until a ugNoMoreUsers or ugEOF error is returned.

If ugGroupID contains a group ID and a user is specified by ugUserID or ugNamePtr, then the user has to be a member of that group. Otherwise, a ugNotaMember error is returned.

Result Codes

noErr	0	No error.
ugULInfoRecNotCreated groups	3	There is a problem with the users and file. Delete it and re-create it.

ugUserNotFound	5	The user is not in the users and groups file.
ugGroupNotFound	6	The group is not in the users and groups file.
ugNotAUser	7	ugUserID is not the ID of a user.
ugNotAGroup	8	ugGroupID is not the ID of a group.
ugNotaMember	10	User is not a member of the group specified by ugGroupID.
ugNoMoreUsers	11	There are no more users in the users and groups file, or there are no more users in the group.
ugBadParamErr	13	async is TRUE, the length of the user name is greater than 31, or ugPosOffset is not 0 or 1.
ugEOF	23	End of file.
ugFatalError	99	There is a problem with the users and groups file. Delete it and re-create it.

UGSetUserInfo

```
FUNCTION UGSetUserInfo (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;
```

```
pascal OSErr UGSetUserInfo (UserGroupPBPtr thePB, Boolean async);
```

Parameter Block	18	ugNamePtr	long	pointer to user name
	24	ugRefNum	word	file reference number
	30	ugAccessID	long	program access ID
	34	ugUserID	long	user ID
	48	ugChngFlags	word	change flag codes
	50	ugPassword	long	pointer to user password
	54	ugUserFlags	long	user flags

UGSetUserInfo can be used to set ugPassword (a pointer to a Str8), the user's password, and the user flags specified by ugUserFlags. (Refer to "Data Structures," later in this document, for details.) The user is specified either by ugUserID or

by ugNamePtr (a pointer to a Str31). If ugUserID is used, then ugNamePtr should be a pointer to a zero-length Str31. If ugNamePtr is used, then ugUserID should be 0. If both fields have values, then ugUserID takes precedence. ugChngFlags should be set to ChngUserPasswd (1) if the password is to be changed, to ChngUserFlags (2) if the user's flags are to be changed, and to ChngUAll (3) if both are to be changed.

Result Codes

noErr	0	No error.
ugUserNotFound	5	The user is not in the users and groups file.
ugNotAUser	7	ugUserID is not the ID of a user.
ugBadParamErr	13	async is TRUE, the length of the password is greater than 8, or the length of the user name is greater than 31.
ugServerRunning	16	Server is running.
ugAdminRunning	17	Admin is running.
ugNotBtree	18	Users and groups file is not a B-Tree file.
ugBadUID	21	The ID in ugAccessID is wrong. The users and groups file may not be open.
ugFatalError	99	There is a problem with the users and groups file. Delete it and re-create it.

UGAuthenticateUser

FUNCTION UGAuthenticateUser (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

pascal OSErr UGAuthenticateUser (UserGroupPBPtr thePB, Boolean async);

Parameter Block	18	ugNamePtr	long	pointer to user name
	24	ugRefNum	word	file reference number
	30	ugAccessID	long	program access ID
	34	ugUserID	long	user ID
	50	ugPassword	long	pointer to user password

UGAuthenticateUser authenticates the user's password. The user is specified either by ugUserID or by ugNamePtr (a pointer to a Str31), and the password is specified in ugPassword (a pointer to a Str8). If ugUserID is used, then ugNamePtr should be a pointer to a zero-length Str31. If ugNamePtr is used, then ugUserID should be 0. If both fields have values, then ugUserID takes precedence.

Result Codes

noErr	0	No error.
ugUserNotFound	5	The user is not in the users and groups file.
ugNotAUser	7	ugUserID is not the ID of a user.

ugBadParamErr	13	async is TRUE, the length of the password is greater than 8, or the length of the user name is greater than 31.
ugInvalidPasswd	22	ugPassword is not the correct password.
ugFatalError	99	There is a problem with the users and groups file. Delete it and re-create it.

UGNewGroup

FUNCTION UGNewGroup (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

pascal OSErr UGNewGroup (UserGroupPBPtr thePB, Boolean async);

Parameter Block	18	ugNamePtr	long	pointer to group name
	24	ugRefNum	word	file reference number
	30	ugAccessID	long	program access ID
	38	ugGroupID	long	group ID

UGNewGroup creates a new group. The name of the group is specified in ugNamePtr (a pointer to a Str31) and the group ID is returned in ugGroupID.

Result Codes

noErr	0	No error.
ugUserAlreadyExists	1	User with that name already exists.
ugGroupAlreadyExists	2	Group with that name already exists.
ugNoMoreUGs	4	The maximum value (\$FFFFFFFF) has been reached.
ugBadParamErr	13	async is TRUE, ugNamePtr contains a colon, or ugNamePtr points to a string longer than 31 characters or shorter than 1 character.
ugServerRunning	16	Server is running.
ugAdminRunning	17	Admin is running.
ugNotBtree	18	Users and groups file is not a B-Tree file.
ugBadUID	21	The ID in ugAccessID is wrong. The users and groups file may not be open.
ugFatalError	99	There is a problem with the users and groups file. Delete it and re-create it.

UGDeleteGroup

FUNCTION UGDeleteGroup (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

pascal OSErr UGDeleteGroup (UserGroupPBPtr thePB, Boolean async);

Parameter Block number	24	ugRefNum	word	file reference number
	30	ugAccessID	long	program access ID
	38	ugGroupID	long	group ID

UGDeleteGroup deletes the group specified by ugGroupID.

Result Codes

noErr	0	No error.
ugGroupNotFound	6	The group is not in the users and groups file.
ugNotAGroup	8	ugGroupID is not the ID of a group.
ugBadParamErr	13	async is TRUE.
ugServerRunning	16	Server is running.
ugAdminRunning	17	Admin is running.
ugNotBtree	18	Users and groups file is not a B-Tree file.
ugBadUID	21	The ID in ugAccessID is wrong. The users and groups file may not be open.
ugFatalError	99	There is a problem with the users and groups file. Delete it and re-create it

UGRenameGroup

FUNCTION UGRenameGroup (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

pascal OSErr UGRenameGroup (UserGroupPBPtr thePB, Boolean async);

Parameter Block	18	ugNamePtr	long	pointer to current group name
	24	ugRefNum	word	file reference number
	30	ugAccessID	long	program access ID
	38	ugGroupID	long	group ID
	44	ugMisc	long	pointer to new group name

UGRenameGroup renames the group specified either by ugGroupID or by ugNamePtr (a pointer to a Str31). If ugGroupID is used, then ugNamePtr

should be a pointer to a zero-length Str31. If ugNamePtr is used, then ugGroupID should be 0. If both fields have values, then ugGroupID takes precedence. The new name is specified by ugMisc (a pointer to a Str31).

Result Codes

noErr	0	No error.
ugUserAlreadyExists	1	User with the same name exists.
ugGroupAlreadyExists	2	Group with the same name exists.
ugGroupNotFound	6	The group is not in the users and groups file.
ugNotAGroup	8	ugGroupID is not the ID of a group.
ugBadParamErr	13	async is TRUE, the new group name contains a colon, the group name points to a string longer than 31 characters, or ugMisc points to a string shorter than 1 character or longer than 31 characters.
ugServerRunning	16	Server is running.
ugAdminRunning	17	Admin is running.
ugNotBtree	18	Users and groups file is not a B-Tree file.
ugBadUID	21	The ID in ugAccessID is wrong. The users and groups file may not be open.
ugFatalError	99	There is a problem with the users and groups file. Delete it and re-create it

UGGetGroupInfo

FUNCTION UGGetGroupInfo (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

pascal OSErr UGGetGroupInfo (UserGroupPBPtr thePB, Boolean async);

Parameter Block	18	ugNamePtr	long	pointer to group name
	24	ugRefNum	word	file reference number
	30	ugAccessID	long	program access ID
	38	ugGroupID	long	group ID
	42	ugPosOffset	word	position offset

UGGetGroupInfo can be used to get information about a specific group or to enumerate all of the groups in the file. In both cases, the ugGroupID and the ugNamePtr (a pointer to a Str31) are returned.

When UGGetGroupInfo is used to get information about a specific group, the group is specified either by ugGroupID or by ugNamePtr. If ugGroupID is used,

then ugNamePtr should be a pointer to a zero-length Str31. If ugNamePtr is used, then ugGroupID should be 0. If both fields have values, then ugGroupID takes precedence. If ugPosOffset is 0, then the information passed back will be that of the group specified. If ugPosOffset is 1, then the information passed back will be that of the group following the one specified. If ugPosOffset is neither 0 nor 1, then a ugBadParamErr error will be returned.

To have UGGetGroupInfo enumerate all groups in the users and groups file, your program should set ugGroupID and ugPosOffset to 0, and ugNamePtr should be a pointer to a zero-length Str31. When the function is called, the first group record will be returned, if present. If ugPosOffset is set to 1, the information of the group after ugGroupID is returned. This process is repeated until a ugNoMoreGroups or ugEOF error is returned.

Result Codes

noErr	0	No error.
ugGroupNotFound	6	The group is not in the users and groups file.
ugNotAGroup	8	ugGroupID is not the ID of a group.
ugNoMoreGroups	12	No more groups exist in the users and groups file.
ugBadParamErr	13	async is TRUE, or ugPosOffset is neither 0 nor 1.
ugEOF	23	End of file.
ugFatalError	99	There is a problem with the users and groups file. Delete and re-create it

UGAssignUserToGroup

FUNCTION UGAssignUserToGroup (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

pascal OSErr UGAssignUserToGroup (UserGroupPBPtr thePB, Boolean async);

Parameter Block	24	ugRefNum	word	file reference number
ID	30	ugAccessID	long	program access
	34	ugUserID	long	user ID
	38	ugGroupID	long	group ID
	48	ugPrimaryFlag	Boolean	primary group flag

UGAssignUserToGroup allows your program to add a user to a group. The user is specified in ugUserID and the group is specified in ugGroupID. If ugPrimaryFlag is TRUE, then the group specified becomes the user's primary

group. (The primary group is used by AppleShare File Servers 2.0.1 and 3.0 when Apple II mode is on. Refer to "Data Structures," later in this document, for details.)

Result Codes

noErr	0	No error.
ugUserNotFound	5	The user is not in the users and groups file.
ugGroupNotFound	6	The group is not in the users and groups file.
ugNotAUser	7	ugUserID is not the ID of a user.
ugNotAGroup	8	ugGroupID is not the ID of a group.
ugTooManyGroups	9	User has already been assigned to the maximum number of groups.
ugBadParamErr	13	async is TRUE.
ugUserAlreadyAMember	14	User is already a member of the group.
ugAPrmryGroupExists	15	User already has a primary group.
UGServerRunning	16	Server is running.
ugAdminRunning	17	Admin is running.
ugNotBtree	18	Users and groups file is not a B-Tree file.
ugBadUID	21	The ID in ugAccessID is wrong. The users and groups file may not be open.
ugFatalError	99	There is a problem with the users and groups file. Delete and re-create it.

UGDeleteUserFromGroup

```
FUNCTION UGDeleteUserFromGroup (thePB: UserGroupPBPtr; async: BOOLEAN):
OSErr;
```

```
pascal OSErr UGDeleteUserFromGroup (UserGroupPBPtr thePB, Boolean async)
```

Parameter Block	24	ugRefNum	word	file reference number
	30	ugAccessID	long	program access ID
	34	ugUserID	long	user ID
	38	ugGroupID	long	group ID

UGDeleteUserFromGroup allows your program to remove the user specified in ugUserID from the group specified in ugGroupID.

Result Codes

noErr	0	No error.
ugUserNotFound	5	The user is not in the users and groups file.
ugGroupNotFound	6	The group is not in the users and groups file.
ugNotAUser	7	ugUserID is not the ID of a user.
ugNotAGroup	8	ugGroupID is not the ID of a group.
ugNotAMember	10	User is not a member of the group specified by ugGroupID.
ugBadParamErr	13	async is TRUE.
ugServerRunning	16	Server is running.
ugAdminRunning	17	Admin is running.
ugNotBtree	18	Users and groups file is not a B-Tree file.
ugBadUID open.	21	The ID in ugAccessID is wrong. The users and groups file may not be open.
ugFatalError	99	There is a problem with the users and groups file. Delete and re-create it.

UGGetULInfo

FUNCTION UGGetULInfo (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

pascal OSErr UGGetULInfo(UserGroupPBPtr thePB, Boolean async);

Parameter Block	18	ugNamePtr	long	pointer to server name
	24	ugRefNum	word	file reference number
	30	ugAccessID	long	program access ID
	46	ugSrvrVersion	word	server version number
	50	ugSrvrFlags	8 bytes	server flags (two longs)
	58	ugMinPwdLen	byte	minimum password length
	60	ugMaxBadLogins	word	maximum failed login attempts

62 ugPwdExpDate long days before passwords expire

UGGetULInfo returns the characteristics of the AppleShare File Server 3.0 or the Macintosh File Sharing server. The information returned includes ugSrvrVersion, the version number of the server; ugNamePtr (a pointer to a Str31), the name of the server; ugSrvrFlags (an array of two longs), the server settings (further described in "Data Structures," later in this document); ugMinPwdLen, the minimum password

length; ugMaxBadLogins, the maximum number of failed login attempts that the file server can allow; and ugPwdExpDate, the number of days before user passwords expire.

A ugSrvrVersion value of Sys7SrvrVer (3) indicates Macintosh File Sharing and a value of AS3SrvrVer (4) indicates the AppleShare File Server 3.0.

Result Codes

noErr	0	No error.
ugULInfoRecNotCreated	3	There is a problem with the users and groups file. Delete it and re-create it.
ugBadParamErr	13	async is TRUE.
ugFatalError	99	There is a problem with the users and groups file. Delete it and re-create it.

UGSetULInfo

FUNCTION UGSetULInfo (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

pascal OSErr UGSetULInfo (UserGroupPBPtr thePB, Boolean async)

Parameter Block number	24	ugRefNum	word	file reference
	30	ugAccessID	long	program access ID
	42	ugAdminKey	long	pointer to administrator password
	46	ugSrvrVersion	word	server version number
	48	ugChngFlags	word	change flag codes
	50	ugSrvrFlags	8 bytes	server flags (two longs)
	58	ugMinPwdLen	byte	minimum password length

60	ugMaxBadLogins	word	maximum failed login attempts
62	ugPwdExpDate	long	days before user passwords expire

UGSetULInfo sets the file server characteristics. ugChngFlags should be set to ChngSrvrFlags (1) if ugSrvrFlags (an array of two longs, described further in "Data Structures" later in this document) is to be changed; ChngAdmnKey (2) if ugAdminKey (a pointer to a Str8) is to be changed; ChngMinPwdLen (4)

if ugMinPwdLen (which must be between 0 and 8 inclusive) is to be changed; ChngMaxBadLogins (8) if ugMaxBadLogins (which must be between 0 and 100 inclusive) is to be changed; ChngPwdExpDate (16) if ugPwdExpDate (which must be between 0 and 1000 inclusive) is to be changed; ChngSrvrVersion (32)

if ugSrvrVersion (which must be Sys7SrvrVer (3) or AS3SrvrVer (4)) is to be changed; or ChngULAll (63) if all of these values are to be changed.

Result Codes

noErr	0	No error.
ugULInfoRecNotCreated	3	There is a problem with the users and groups file. Delete it and re-create it.
ugBadParamErr	13	async is TRUE, ugAdminKey is larger than 8, ugMinPwdLen is not between 0 and 8 (inclusive), ugMaxBadLogins is not between 0 and 100 (inclusive), ugPwdExpDate is not between 0 and 1000 (inclusive), or ugSrvrVersion is not Sys7SrvrVer (3) or AS3SrvrVer (4).
ugServerRunning	16	Server is running.
ugAdminRunning	17	Admin is running.
ugNotBtree	18	Users and groups file is not a B-Tree file.
ugBadUID	21	The ID in ugAccessID is wrong. The users and groups file may not be open.
ugInvalidPasswd	22	ugSrvrVersion is Sys7SrvrVer (3) and ugAdminKey is non-nil.
ugFatalError	99	There is a problem with the users and groups file. Delete it and re-create it.

Data structures

This section describes the bits in the ugUserFlags field of UGUserPB and the bits in the ugSrvrFlags field of UGServerPB. It also lists the parameter blocks needed to call the Users and Groups library functions.

Bits in the ugUserFlags field of UGUserPB

Bit

User flag	number	Description
IACEnabled	0	Set if IAC is enabled for this user.
LoginEnabled	8	Set if login is enabled.
SuperUser	9	Set if this user is a superuser. Only one superuser is allowed under Macintosh File Sharing: the system owner (administrator).
NoChangePwd	10	Set if the user cannot change his or her password from a workstation.
GetNewPwd	12	Set if the server should demand a new password from the user at the next login. This flag is available only on the AppleShare File Server 3.0; it must be clear for Macintosh File Sharing.

All other ugUserFlags bits are reserved for the use of Apple Computer. When your program creates a new user with UGNewUser, the reserved bits should be cleared. When your program is setting or clearing ugUserFlags bits for an existing user, the state of the reserved bits must be preserved.

The following figure shows the ugUserFlags bits.

Bits in the ugSrvrFlags field of UGServerPB

Bit

Server flag	number	Description
Apple2	0	Set if server is in Apple II mode. This flag is used for the AppleShare File Server 3.0 only; it must be clear for Macintosh File Sharing.
NoGuests	1	Set if guests cannot log on to this server.
IACGuestEnabled	4	Set if IAC is turned on for guest.
IACGlobalEnabled computer.	5	Set if IAC is turned on for this
DontSavePwd	0	Set if the AppleShare workstation should not allow the user to save his or her password. This flag is available on the AppleShare File Server 3.0 only; it must be clear for Macintosh File Sharing.

All other ugSrvrFlags bits are reserved for the use of Apple Computer. When your program sets or clears ugSrvrFlags bits for an existing Users & Groups

Data File, the state of the reserved bits should be preserved.

The following figure shows the ugSrvrFlags bits.

All of the ugSrvrFlags flags are effective as soon as the file server is started, except for IACGlobalEnabled, which is effective only after the computer is restarted.

Parameter blocks needed to call the Users and Groups library functions

In Pascal:

```
{ Use UGFilePB for UGOpenFile, UGCloseFile and UGCreateFile }
```

```
UGFilePB = RECORD
```

```
  qLink: QElemPtr;
```

```
  qType: INTEGER;
```

```
  ugTrap: INTEGER;
```

```
  ugCmdAddr: Ptr;
```

```
  ugCompletion: ProcPtr;
```

```
  ugResult: OSErr;
```

```
  ugNamePtr: StringPtr;
```

```
  ugVRefNum: INTEGER;
```

```
  ugRefNum: INTEGER;
```

```
  ugDirID: LONGINT;
```

```
  ugAccessID: LONGINT;
```

```
  ugUserID: LONGINT;
```

```
  ugGroupID: LONGINT;
```

```
  ugAdminKey: StringPtr;
```

```
END;
```

```
{ Use UGUserPB for UGNewUser, UGDeleteUser, UGRenameUser, }
```

```
{ UGGetUserInfo, UGSetUserInfo, and UGAuthenticateUser }
```

```
UGUserPB = RECORD
```

```
  qLink: QElemPtr;
```

```
  qType: INTEGER;
```

```
  ugTrap: INTEGER;
```



```
ugCmdAddr: Ptr;
ugCompletion: ProcPtr;
ugResult: OSErr;
ugNamePtr: StringPtr;
ugVRefNum: INTEGER;
ugRefNum: INTEGER;
ugDirID: LONGINT;
ugAccessID: LONGINT;
ugUserID: LONGINT;
ugGroupID: LONGINT;
ugPosOffset: INTEGER;
ugMisc: StringPtr;
ugChngFlags: INTEGER;
ugPassword: StringPtr;
ugUserFlags: LONGINT;
ugNumBadLogins: INTEGER;
ugPwdDate: LONGINT;
ugNumGroups: INTEGER;
ugGroups: ARRAY[0..41] OF LONGINT;
END;

{ Use UGGroupPB for UGNewGroup, UGDeleteGroup, UGRenameGroup, }
{ UGGetGroupInfo, UGAssignUserToGroup, and UGDeleteUserFromGroup }
UGGroupPB = RECORD
qLink: QElemPtr;
qType: INTEGER;
ugTrap: INTEGER;
ugCmdAddr: Ptr;
ugCompletion: ProcPtr;
ugResult: OSErr;
```

```
ugNamePtr: StringPtr;
ugVRefNum: INTEGER;
ugRefNum: INTEGER;
ugDirID: LONGINT;
ugAccessID: LONGINT;
ugUserID: LONGINT;
ugGroupID: LONGINT;
ugPosOffset: INTEGER;
ugMisc: StringPtr;
ugPrimaryFlag: Boolean;
END;
{ Use UGServerPB for UGGetULInfo and UGSetULInfo }
UGServerPB = RECORD
qLink: QElemPtr;
qType: INTEGER;
ugTrap: INTEGER;
ugCmdAddr: Ptr;
ugCompletion: ProcPtr;
ugResult: OSErr;
ugNamePtr: StringPtr;
ugVRefNum: INTEGER;
ugRefNum: INTEGER;
ugDirID: LONGINT;
ugAccessID: LONGINT;
ugUserID: LONGINT;
ugGroupID: LONGINT;
ugAdminKey: StringPtr;
ugSrvrVersion: INTEGER;
ugChngFlags: INTEGER;
```

```
ugSrvrFlags: ARRAY[0..1] OF LONGINT;
ugMinPwdLen: SignedByte;
filler1: SignedByte;
ugMaxBadLogins: INTEGER;
ugPwdExpDate: LONGINT;
END;
UserGroupPBPtr = ^UGParamBlockRec;
UGParamBlockRec = RECORD
CASE Integer OF
0: ( ugFileParam: UGFilePB );
1: ( ugUserParam: UGUserPB );
2: ( ugGroupParam: UGGroupPB );
3: ( ugServerParam: UGServerPB );
END;
In C:
#define  UGPBHeader
QElemPtr  qLink;
short     qType;
short     ugTrap;
Ptr       ugCmdAddr;
ProcPtr   ugCompletion;
OSError   ugResult;
StringPtr ugNamePtr;
short     ugVRefNum;
short     ugRefNum;
long      ugDirID;
long      ugAccessID;
long      ugUserID;
long      ugGroupID;
```

```

struct  UGFilePB
UGPBHeader
StringPtr  ugAdminKey;
typedef  struct      UGFilePB  UGFilePB;
struct  UGUserPB
UGPBHeader
short      ugPosOffset;
StringPtr  ugMisc;
short      ugChngFlags;
StringPtr  ugPassword;
long       ugUserFlags;
short      ugNumBadLogins;
long       ugPwdDate;
short      ugNumGroups;
long       ugGroups[42];
typedef  struct      UGUserPB  UGUserPB;
struct  UGGroupPB
UGPBHeader
short      ugPosOffset;
StringPtr  ugMisc;
Boolean    ugPrimaryFlag;
typedef  struct      UGGroupPB  UGGroupPB;
struct  UGServerPB
UGPBHeader
StringPtr  ugAdminKey;
short      ugSrvrVersion;
short      ugChngFlags;
long       ugSrvrFlags[2];
Byte       ugMinPwdLen;

```

```

short      ugMaxBadLogins;
long       ugPwdExpDate;
typedef struct    UGServerPB UGServerPB;
union    UGParamBlockRec
UGFilePB  ugFilePB;
UGUserPB  ugUserPB;
UGGroupPB ugGroupPB;
UGServerPB ugServerPB;
typedef    UGParamBlockRec *UserGroupPBPtr;

```

Function summary

This section provides a list of all functions currently available in the Users and Groups library. It also specifies the parameter block to be used with each function.

Functions that use the UGFilePB parameter block

In Pascal:

```

FUNCTION UGOpenFile (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;
FUNCTION UGCloseFile (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;
FUNCTION UGCreateFile (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

```

In C:

```

pascal OSErr UGOpenFile (UserGroupPBPtr thePB, Boolean async);
pascal OSErr UGCloseFile (UserGroupPBPtr thePB, Boolean async);
pascal OSErr UGCreateFile (UserGroupPBPtr thePB, Boolean async);

```

Functions that use the UGUserPB parameter block

In Pascal:

```

FUNCTION UGNewUser (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;
FUNCTION UGDeleteUser (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;
FUNCTION UGRenameUser (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;
FUNCTION UGGetUserInfo (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;
FUNCTION UGSetUserInfo (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;
FUNCTION UGAuthenticateUser (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;

```

In C:

```
pascal OSErr UGNewUser (UserGroupPBPtr thePB, Boolean async);  
pascal OSErr UGDeleteUser (UserGroupPBPtr thePB, Boolean async);  
pascal OSErr UGRenameUser (UserGroupPBPtr thePB, Boolean async);  
pascal OSErr UGGetUserInfo (UserGroupPBPtr thePB, Boolean async);  
pascal OSErr UGSetUserInfo (UserGroupPBPtr thePB, Boolean async);  
pascal OSErr UGAuthenticateUser (UserGroupPBPtr thePB, Boolean async);
```

Functions that use the UGGroupPB
parameter block

In Pascal:

```
FUNCTION UGNewGroup (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;  
FUNCTION UGDeleteGroup (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;  
FUNCTION UGRenameGroup (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;  
FUNCTION UGGetGroupInfo (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;  
FUNCTION UGAssignUserToGroup (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;  
FUNCTION UGDeleteUserFromGroup (thePB: UserGroupPBPtr; async: BOOLEAN):  
OSErr;
```

In C:

```
pascal OSErr UGNewGroup (UserGroupPBPtr thePB, Boolean async );  
pascal OSErr UGDeleteGroup (UserGroupPBPtr thePB, Boolean async );  
pascal OSErr UGRenameGroup (UserGroupPBPtr thePB, Boolean async );  
pascal OSErr UGGetGroupInfo (UserGroupPBPtr thePB, Boolean async );  
pascal OSErr UGAssignUserToGroup (UserGroupPBPtr thePB, Boolean async);  
pascal OSErr UGDeleteUserFromGroup (UserGroupPBPtr thePB, Boolean async);
```

Functions that use the UGServerPB
parameter block

In Pascal:

```
FUNCTION UGGetULInfo (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;  
FUNCTION UGSetULInfo (thePB: UserGroupPBPtr; async: BOOLEAN): OSErr;
```

In C:

```
pascal OSErr UGGetULInfo (UserGroupPBPtr thePB, Boolean async);
```

```
pascal OSErr UGSetULInfo (UserGroupPBPtr thePB, Boolean async);
```

Result codes returned by

UGLibrary functions

ugUserAlreadyExists	= 1
ugGroupAlreadyExists	= 2
ugULInfoRecNotCreated	= 3
ugNoMoreUGs	= 4
ugUserNotFound	= 5
ugGroupNotFound	= 6
ugNotAUser	= 7
ugNotAGroup	= 8
ugTooManyGroups	= 9
ugNotaMember	= 10
ugNoMoreUsers	= 11
ugNoMoreGroups	= 12
ugBadParamErr	= 13
ugUserAlreadyAMember	= 14
ugAPrmryGroupExists	= 15
ugServerRunning	= 16
ugAdminRunning	= 17
ugNotBtree	= 18
ugBadKey	= 19
ugDskFull	= 20
ugBadUID	= 21
ugInvalidPasswd	= 22
ugEOF	= 23
ugBOF	= 24
ugDefNameErr	= 98

ugFatalError = 99

For more information

APDA offers convenient worldwide access to over three hundred development tools, resources, and training products, and to information for anyone interested in developing applications on Apple platforms. Customers receive the quarterly APDA Tools Catalog featuring the most current versions of Apple development tools and the the most popular third-party development tools. Ordering is easy; there are no membership fees, and application forms are not required for most of our products.

If you plan to develop Apple-compatible hardware or software products for sale through retail channels, you can get valuable support from the Developer Support Center:

The Apple Publishing System

AppleShare 3.0 Developer's Kit: Users and Groups Programming Interface, Version 1.0, was written, edited, and composed on a desktop publishing system using Apple Macintosh computers, an AppleTalk network system, Microsoft Word, and QuarkXPress. Line art was created with Adobe Illustrator. Proof pages were printed on Apple LaserWriter printers. Final pages were output on a Varsity VT600. PostScript, the LaserWriter page-description language, was developed by Adobe Systems Incorporated.

Text and display type are Apple's corporate font, a condensed version of Garamond. Bullets are ITC Zapf Dingbats. Some elements, such as program listings, are set in Apple Courier, a fixed-width font.