

# The Group Edition Suite

This document describes the Apple Events supported by the programs in Group Edition version 3.0, including Node Server, User Manager, Caller Log, and Transfer Log.

## Node Server Events

The primary purpose of the Node Server events is to provide external applications control over the execution of Node Server.

For example, an external application may want to bring Node Server off-line to free memory during maintenance periods, or to free a serial port so that another communications program may use it, or to act as a "Front Door" to the BBS to pre-process incoming connections.

The Get Node Name event provides external applications a way of distinguishing a specific BBS node from the several that may be running on a given Macintosh.

Also refer to the document: "GEHEvents."

## Send User an Alert

This event sends a short ( up to 127 character ) message to the on-line user. The message is buffered ff a file transfer is in progress. Node Server can buffer just one message at a time. The message is discarded if there is not a user on-line, or if a mmessage is already in the buffer. The event is asynchronous.

The user sees this message in the "TeleFinder Message" window.

**Event Class** kGehClass

**Event ID** kGehSendUserAlert

### Parameters

kGehAlertMessage	Description	A string containing the message. The length can be up to 127 characters.
------------------	-------------	--

Descriptor Type:	typeChar
Required or Optional?	Required

**Reply Parameters** None

**Result Codes** Operating system error or zero for no error

**Note** An FSSpec record may be passed into the event instead of an alias record.

## **Cancel any Transfers in Progress**

This event instructs Node Server to cancel any transfers a user may have in progress. Nothing occurs if a user is not transferring a file. The event is asynchronous.

<b>Event Class</b>	kGehClass
<b>Event ID</b>	kGehCancelXfrs
<b>Parameters</b>	None
<b>Reply Parameters</b>	None
<b>Result Codes</b>	always 0

## **Disconnect User**

This event instructs Node Server to disconnect any on-line user. Node Server stops the session in progress, then hangs up the phone. Node Server remains on-line for new calls. The event is synchronous.

<b>Event Class</b>	kGehClass
<b>Event ID</b>	kGehDisconnectUser
<b>Parameters</b>	None
<b>Reply Parameters</b>	None
<b>Result Codes</b>	Always 0

## **Go Off-Line**

This event instructs Node Server to go off-line. Node Server will stop polling for incoming calls, reset the modem, and close it's serial port. The event is asynchronous.

<b>Event Class</b>	kGehClass
<b>Event ID</b>	kGehGoOffline
<b>Parameters</b>	None
<b>Reply Parameters</b>	None
<b>Result Codes</b>	Always 0

## **Go On-Line**

This event instructs Node Server to go on-line. Node Server will open its serial port, wake and initialize the modem, and start polling for incoming calls. The event is asynchronous.

<b>Event Class</b>	kGehClass
<b>Event ID</b>	kGehGoOnline
<b>Parameters</b>	None
<b>Reply Parameters</b>	None
<b>Result Codes</b>	Always 0

## Make Direct Connection

This event instructs Node Server to start the user logon process.

**Event Class** kGehClass

**Event ID** kGehGoDirect

### Parameters

kGehBaudRate	Description	An integer describing the baud rate to start the logon at. Valid values are: 300, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600. Descriptor Type: typeLongInteger Required or Optional? Required
--------------	-------------	---

**Reply Parameters** None

**Result Codes**  
1: Not online  
2: Completed

## Get Node Name

This event queries Node Server for it's Node Name. This is the string the sysop enters in the "BBS Node Options" dialog to identify a node of the BBS.

**Event Class** kGehClass

**Event ID** kGehNodeName

**Parameters** None

### Reply Parameters

kGehBaudRate	Description	A string containing the Node Name. The length will be up to 31 characters.
	Descriptor Type:	typeLongInteger
	Required or Optional?	Required

**Result Codes** 0: Completed.  
<0: System error code.

## User Manager Events

The User Manager events are an integral part of the operation of a Group Edition BBS. They allow Node Server to get the name, password, and privilege information it needs to provide proper security and access to the BBS. They also allow Node Server, and other applications to perform maintenance on the user data base.

Also refer to the documents: "UMAppleEvents.h," "UserFile.h," "UserAccessFile.h," and "UMEvents.c."

## Start User Manager

This event queries User Manager for the active user data base. The receipt of this event triggers a user warning if a user tries to close the active user data base. The reply to the event contains an error code, and aliases of the user database, and folder containing access group files.

**Event Class** umClass

**Event ID** umStartUM

### Parameters

umPassword	Description	A string containing the access password to the user database. The length can be up to 9 characters.
	Descriptor Type:	typeChar
	Required or Optional?	optional, User Manager 3.0 ignores this parameter

### Reply Parameters

umUserFileSpec	Description	An alias of the active TeleUsers file.
	Descriptor Type:	typeAlias
umAGFolderSpec	Description	An alias of the folder containing the access group files.
	Descriptor Type:	typeAlias

### Result Codes

0: Completed.  
 1005: There is not an active user database.  
 <0: System error code.

**Note:** The AppleEvent manager will automatically coerce typeAlias to typeFSSpec so that you do not need to manually resolve the alias.

## Get User

Use this event to fetch the user record of a specific user.

**Event Class**                      umClass

**Event ID**                              umGetUser

### Parameters

umPassword

**Description**              A string containing the access password to the user database. The length can be up to 9 characters.

**Descriptor Type:** typeChar  
**Required or Optional?**      optional,  
 User Manager 3.0 ignores this parameter.

umUserName

**Description**              A string containing a user name. The length can be up to 31 characters.

**Descriptor Type:** typeChar  
**Required or Optional?**      required.

### Reply Parameters

umUserRecord

**Description**              The UserRecord structure for the user. See "UserFile.h" for details of this structure.

**Descriptor Type:** typeWildcard

### Result Codes

0: Completed.

1001: User not found.

1005: There is not an active user database.

<0: System error code.

## Add a User

Use this event to to add a new user record to the user database. This event requires that you create a mailbox and the UserRecord data structure. This event lets you specify the exact contents of the user record. Use the "Create a User" event when you want User Manager to create the mailbox and structure for you.

**Event Class** umClass

**Event ID** umAddUser

### Parameters

umPassword

**Description** A string containing the access password to the user database. The length can be up to 9 characters.

**Descriptor Type:** typeChar  
**Required or Optional?** optional,  
 User Manager 3.0 ignores this parameter.

umUserRecord

**Description** The UserRecord structure for the user. See "UserFile.h" for details of this structure.

**Descriptor Type:** typeWildcard  
**Required or Optional?** required.

**Reply Parameters** None

### Result Codes

0: Completed.  
 1002: Duplicate user name.  
 1005: There is not an active user database.  
 <0: System error code.

## Create a User

Use this event to to add a new user record to the user database. You only need to provide a user name, password, and voice telephone number for this event. User Manager will create a mailbox for the user, and create the UserRecord data structure using its "New User Settings" information.

**Event Class** umClass

**Event ID** umCreateUser

### Parameters

**umPassword**

Description A string containing the access password to the user database. The length can be up to 9 characters.  
Descriptor Type: typeChar  
Required or Optional? optional,  
User Manager 3.0 ignores this parameter.

**umUserName**

Description A string containing the name of the new user. The length can be up to 31 characters.  
Descriptor Type: typeChar  
Required or Optional? required.

**umUserPassword**

Description A string containing the password for the new user. The length can be up to 9 characters.  
Descriptor Type: typeChar  
Required or Optional? required.

**umUserPhone**

Description A string containing the voice phone number for the new user. The length can be up to 19 characters.  
Descriptor Type: typeChar  
Required or Optional? required.

**Reply Parameters** None

**Result Codes**

0: Completed.  
1002: Duplicate user name.  
1005: There is not an active user database.  
<0: System error code.

## Delete a User

Use this event to delete the user record and mailbox of a specific user.

**Event Class** umClass

**Event ID** umDeleteUser

### Parameters

umPassword

**Description** A string containing the access password to the user database. The length can be up to 9 characters.

**Descriptor Type:** typeChar  
**Required or Optional?** optional,  
 User Manager 3.0 ignores this parameter.

umUserName

**Description** A string containing a user name. The length can be up to 31 characters.

**Descriptor Type:** typeChar  
**Required or Optional?** required.

**Reply Parameters** None

### Result Codes

0: Completed.  
 1001: User not found.  
 1005: There is not an active user database.  
 <0: System error code.

## Change a User

Use this event to to add a change the record of a user in the user database. This event requires that you prooperly fill in the UserRecord data structure. This event lets you specify the exact contents of the user record.

**Event Class** umClass

**Event ID** umChangeUser

### Parameters

umPassword

**Description** A string containing the access password to the user database. The length can be up to 9 characters.

**Descriptor Type:** typeChar  
**Required or Optional?** optional,  
 User Manager 3.0 ignores this parameter.

umUserRecord

**Description** The UserRecord structure for the user. See "UserFile.h" for details of this structure.

**Descriptor Type:** typeWildCard  
**Required or Optional?** required.

**Reply Parameters** None

### Result Codes

0: Completed.  
 1001: User not found.  
 1005: There is not an active user database.  
 <0: System error code.

## Update a User's Statistics

Use this event to update a user's last logon time, mail, file transfer, and message statistics.

**Event Class** umClass

**Event ID** umUpdateUser

### Parameters

umPassword	Description A string containing the access password to the user database. The length can be up to 9 characters. Descriptor Type: typeChar Required or Optional? optional, User Manager 3.0 ignores this parameter.
umUserName	Description A string containing a user name. The length can be up to 31 characters. Descriptor Type: typeChar Required or Optional? required.
umUserRecord	Description The UserRecord structure for the user. The fields used are: lastUpdate, totalTime, kb_uploaded, kb_downloaded, email_sent, and msgs_posted.  Send the totals accumulated in the session in the event.  See "UserFile.h" for details of this structure.  Descriptor Type: typeWildcard Required or Optional? required.

### Reply Parameters

None

### Result Codes

0: Completed.  
1001: User not found.  
1005: There is not an active user database.  
<0: System error code.

## Change a User's Password

Use this event to change a user's password.

**Event Class** umClass

**Event ID** umChangePassword

### Parameters

umPassword

**Description** A string containing the access password to the user database. The length can be up to 9 characters.

**Descriptor Type:** typeChar  
**Required or Optional?** optional,  
 User Manager 3.0 ignores this parameter.

umUserName

**Description** A string containing a user name. The length can be up to 31 characters.

**Descriptor Type:** typeChar  
**Required or Optional?** required.

umUserPassword

**Description** A string containing the user's new password. The length can be up to 9 characters.

**Descriptor Type:** typeChar  
**Required or Optional?** required.

**Reply Parameters** None

### Result Codes

0: Completed.  
 1001: User not found.  
 1005: There is not an active user database.  
 <0: System error code.

## Get a User's Mailbox

Use this event to find the location of the folder that is the user's mailbox.

**Event Class** umClass

**Event ID** umGetMailPath

### Parameters

umPassword	Description	A string containing the access password to the user database. The length can be up to 9 characters.
	Descriptor Type:	typeChar
	Required or Optional?	optional, User Manager 3.0 ignores this parameter.

umUserName	Description	A string containing a user name. The length can be up to 31 characters.
	Descriptor Type:	typeChar
	Required or Optional?	required.

### Reply Parameters

umMailPath	Description	An alias of the user's mailbox folder.
	Descriptor Type:	typeAlias

**Result Codes**

- 0: Completed.
- 1001: User not found.
- 1005: There is not an active user database.
- <0: System error code.

**Note:** The AppleEvent manager will automatically coerce typeAlias to typeFSSpec so that you do not need to manually resolve the alias.

## Scan User Name List

Use this event to scan through the list of user names in the active user database.

<b>Event Class</b>	umClass	
<b>Event ID</b>	umScanUsers	
<b>Parameters</b>		
umPassword		<p>Description     A string containing the access password to the user database. The length can be up to 9 characters.</p> <p>Descriptor Type: typeChar          Required or Optional?     optional,          User Manager 3.0 ignores this parameter.</p>
umLastUser		<p>Description     A string containing a reference user name. The length can be up to 31 characters.</p> <p>Descriptor Type: typeChar          Required or Optional?     required.</p>
umGoForward		<p>Description     A value that tells user manager to go forward(1), or backward(0), from the reference user name.</p> <p>Descriptor Type: typeSMInt          Required or Optional?     required.</p>
umDoLoadUser		<p>Description     A value that tells user manager whether or not to return the UserRecord data structure with the name in the AppleEvent reply.</p> <p>Descriptor Type: typeSMInt          Required or Optional?     required.</p>

### Reply Parameters

umUserName

Description A string containing a reference user name. The length can be up to 31 characters.  
Descriptor Type: typeChar

umUserRecord

Description The UserRecord structure for the user. See "UserFile.h" for details of this structure. The data is valid only if the "umDoLoadUser" value is not zero.  
Descriptor Type: typeWildCard

### Result Codes

0: Completed.  
1001: User not found.  
1005: There is not an active user database.  
<0: System error code.

## Transfer Log Events

The Transfer Log events provide Node Server a way to communicate transfer activity to Transfer Log, and learn the location and name of the ZMODEM Partial Files database. The Partial files database contains information that Node Server uses to re-start failed ZMODEM file transfers.

External programs may want to receive these events to pre-process them, or to take special action in response to certain transfer events; then pass them through to Transfer Log so that it may also process them. An external program could also receive and process the event and replace the Transfer Log program altogether.

Also refer to the document: "FLEvents.h" and "SendFLEvent.c"

## Add Activity to Transfer List

Use this event to add transfer or mail activity to Transfer Log's list.

<b>Event Class</b>	flClass	
<b>Event ID</b>	flAddFileToList	
<b>Parameters</b>		
flUserName		<p>Description     A string containing the name of the user responsible for the activity. The length can be up to 31 characters.</p> <p>Descriptor Type: typeChar</p> <p>Required or Optional?     Required.</p>
flBBSName		<p>Description     A string containing the BBS Node Name. The length can be up to 31 characters.</p> <p>Descriptor Type: typeChar</p> <p>Required or Optional?     required.</p>
flFileSpec		<p>Description     An alias describing the file.</p> <p>Descriptor Type: typeAlias</p> <p>Required or Optional?     required.</p>
flXfrCode		<p>Description     A code to describe the type of activity the event is reporting. See "Activity Codes" in the Transfer Log constants defined at the end of this document.</p> <p>Descriptor Type: typeSMInt</p> <p>Required or Optional?     required.</p>
flDirType		<p>Description     A code to describe the type of directory the event took place in. This is always 0 in Group Edition ver 3.0, future versions will pass defined values here.</p> <p>Descriptor Type: typeSMInt</p> <p>Required or Optional?     required.</p>
flElapsedTime		<p>Description     The elapsed time of the file transfer in seconds.</p> <p>Descriptor Type: typeLongInteger</p> <p>Required or Optional?     required.</p>

flBytesSent

Description The number of bytes sent in the file transfer.

Descriptor Type: typeLongInteger  
Required or Optional? required.

flFileSize

Description The size of the file, in bytes.

Descriptor Type: typeLongInteger  
Required or Optional? required.

flRetransmit

Description The number of re-transmits that occurred during the file transfer.

Descriptor Type: typeSMInt  
Required or Optional? required.

flBaudRate

Description The connection baud rate in bits per second, 0 if speed is not applicable.

Descriptor Type: typeLongInteger  
Required or Optional? required.

flThruPut

Description The average transfer thru-put in bits per second.

Descriptor Type: typeLongInteger  
Required or Optional? required.

flTimeComplete

Description The Mac system time at the moment the file transfer completed.

Descriptor Type: typeLongInteger  
Required or Optional? required.

**Reply Parameters**

None

**Result Codes**

0: Completed.  
<0: System error code.



## Caller Log Events

Caller Log responds to events to add users to its list of calls to the BBS. The events takes two forms, one form adds the caller to the list, another form of the event adds call summary information to the list.

External programs may want to receive these event to pre-process them; then pass them through to Caller Log so that it may also process them. An external program could also receive and process the events and replace the Caller Log program altogether.

Also refer to the document: "CLEvents.h" and "CLEvents.c."





Description The number of e-mail mnesasges sent during the call.  
Descriptor Type: typeLongInteger  
Required or Optional? required.

clNumPosted

Description The number of mnesasges posted to topcis during the call.  
Descriptor Type: typeLongInteger  
Required or Optional? required.

clCallErrorCode

Description A code to describe the type of activity the event is reporting. See "Call Codes" in the Caller Log constants defined at the end of this document.  
Descriptor Type: typeSMInt  
Required or Optional? required.

clCallNum

Description The call number that Node Server received from the "Add a New Call to the Caller List" event.  
Descriptor Type: typeLongInteger  
Required or Optional? required.

**Reply Parameters**

None

**Result Codes**

0: Completed.  
<0: System error code.

**Constants defined in the Group Edition Suite**

## Node Server Events

<b>kGehClass</b>	'TFhs'
<b>kGehSendUserAlert</b>	'snda'
<b>kGehCancelXfrs</b>	'cxfr'
<b>kGehDisconnectUser</b>	'disu'
<b>kGehGoOffline</b>	'goff'
<b>kGehGoOnline</b>	'gonl'
<b>kGehGoDirect</b>	'gdir'
<b>kGehGetNodeName</b>	'nodn'
<b>kGehAlertMessage</b>	'galm'
<b>kGehNodeName</b>	'genn'
<b>kGehBaudRate</b>	'gbrt'

## User Manager Events

<b>umClass</b>	'UMan'
<b>umStartUM</b>	'Strt'
<b>umCreateUser</b>	'Crte'
<b>umGetUser</b>	GUsr'
<b>umAddUser</b>	AUsr'
<b>umDeleteUser</b>	'DUsr'
<b>umChangeUser</b>	'CUsr'
<b>umGetMailPath</b>	'GMPt'
<b>umUpdateUser</b>	'Updt'
<b>umChangePassword</b>	'ChPs'
<b>umScanUsers</b>	'Scan'
<b>umUserName</b>	'UNam'
<b>umLastUser</b>	'LNam'
<b>umUserPassword</b>	'UPas'
<b>umUserPhone</b>	'UPho'
<b>umPassword</b>	'Pwrd'
<b>umUserRecord</b>	'URec'
<b>umMailPath</b>	'Mail'
<b>umUserFileSpec</b>	'TUfd'
<b>umAGFolderSpec</b>	'AGfd'
<b>umGoForward</b>	'GoFw'
<b>umDoLoadUser</b>	'DoLU'

## Transfer Log Events

<b>flClass</b>	'TFfl'
<b>flAddFileToList</b>	'tadf'
<b>flGetPartialFile</b>	'brkn'
<b>flUserName</b>	'tusr'
<b>flBBSName</b>	'tbbs'
<b>flFileSize</b>	tfsz'
<b>flFileSpec</b>	'tfss'
<b>flXfrCode</b>	'txcd'
<b>flDirType</b>	'tdtp'
<b>flElapsedTime</b>	'tept'
<b>flBytesSent</b>	'tsnt'
<b>flRetransmit</b>	'trtx'
<b>flBaudRate</b>	'tbps'
<b>flThruPut</b>	'ttrp'
<b>flTimeComplete</b>	'tcmp'

## flXfrCode Activity Codes:

kLogMailReceived = 1  
kLogPostToTopic = 2  
kLogCreateTopic = 3  
kLogXMRcvSuccess = 4  
kLogZMRcvSuccess = 5  
kLogXMSSendSuccess = 6  
kLogZMSSendSuccess = 7  
kLogXMCancel = 8  
kLogZMCancel = 9

## Caller Log Events

<b>clClass</b>	'TFcl'
<b>clNewCaller</b>	'TFnc'
<b>clAddCaller</b>	'TFac'
<b>clUserName</b>	'CLnm'
<b>clBBSName</b>	'CLbb'
<b>clLogonTime</b>	'CLlg'
<b>clElapsedTime</b>	'CLet'
<b>clKbUpload</b>	'CLul'
<b>clKbDownload</b>	'CLdl'
<b>clNumEMail</b>	'CLem'
<b>clNumPosted</b>	'CLnp'
<b>clConnectBPS</b>	'CLbp'

## Call Codes

kNormalCall = 1  
kPasswordError = 2  
kConnectionError = 3  
kNewUserRegistered = 5  
kUserQuitBeforeLogon = 6  
kFidonetMailerScheduleTime = 7  
kFidonetIncomingCrashmail = 8