

EasyPlay's Apple Events

EasyPlay™ can be fully controlled from another application by having that application send high level events. This document describes the events which EasyPlay 1.0 supports.

EasyPlay also has an internal 'aete' resource inside it which describes the supported events. Some other applications may be able to read this resource to determine the events that EasyPlay supports, and the format for each event.

The paragraphs below describe the supported events. At the right side of the first line, the four letter class and ID codes for the event appears, along with the type of the direct parameter and result, if any. At the left side, the UserLand "Frontier" scripting verb appears. Navigator also supports Frontier's shared menu system. If you have Frontier, double-click the "EasyPlay.Frontier" document to install the EasyPlay verbs.

In the descriptions, some keywords describe the data type for arguments and results: "bool" means a true/false value. "item" means a long integer value. "string" means a string of no more than 255 characters. "path" means a text string of no more than 255 characters which specifies a file and its location. "rect" means a rectangle. 'pict' means a standard picture in 'PICT' format. "appID" means a four-letter application creator code, which for EasyPlay is always 'EZpa'. "files" means an object which is a list of file specifications. "fixed" means a fixed-point decimal value, which need not be an integer.

Required Events

required.openApplication (appID,path)

aevt.oapp()

Sent by the Finder when an application is opened by direct double clicking, it has no effect. AppID is the four letter ID for EasyPlay, 'EZpa'.

required.openDocument (appID,path)

aevt.odoc(files)

Opens the specified file(s). AppID is the four letter ID for EasyPlay, 'EZpa'. The path is a pathname specifying the file to open. The event direct parameter is a list of files to open. If a specified file is not a movie or catalog file, it is ignored. If a movie is opened, it may begin playing, and may automatically close when finished, depending on the preferences. Since only one movie file and one catalog file may be open at a time, opening a movie closes any previously open movie, and opening a catalog closes any previously open catalog.

required.printDocument (applID,path)

aevt.pdoc(files)

Asks to print the specified file(s). Since EasyPlay does no support printing, this event is ignored. AppID is the four letter ID for EasyPlay, 'EZpa'.

required.quitApplication (applID)

aevt.quit()

Quits EasyPlay. AppID is the four letter ID for EasyPlay, 'EZpa'.

Dialogs & Errors

app.alertDialog (string)

app1.alrt (string)

Displays the string in a modal dialog box, waiting for the user to click on OK before returning true. This verb allows script writers to deliver a message to the user without forcing the scripting system to the front.

Note: if the application isn't the frontmost application it brings itself to the front before displaying the dialog.

bool app.confirmDialog (string)

bool app1.cnfm (string)

Displays the string in a modal dialog box, waiting for the user to click on OK or Cancel. Returns true if the user clicked on OK, false if Cancel. This verb allows script writers to confirm an action with the user without forcing the scripting system to the front.

Note: if the application isn't the frontmost application it brings itself to the front before displaying the dialog.

bool app.enableDialogs (bool)

bool app1.enbd (bool)

Dialogs are normally enabled, call this verb to disable them, or re-enable them after disabling. The verb takes a boolean parameter, if true, dialogs will appear when an error happens when opening or saving a file, for example.

If the parameter is false, subsequent dialogs will be disabled, verbs will return with no interaction required by the user.

In either case, the script writer can call the `getErrorString` verb to get a string describing any error condition.

Returns a boolean, the value of the enable-dialogs flag before it was set. You can use this value to restore the value of the flag after setting it.

`string app.getErrorString ()`

`string app1.gers ()`

Returns the error string, the result of the last error-prone operation. It's empty if the operation was successful. This verb is useful if you've turned alerts off using the `enableDialogs` verb. Call this verb if a `save` or `open` returns `false`, for example; the string can be displayed in a call to the `Frontier dialog.alert` verb.

Target Windows

`string app.getTargetWindow ()`

`app1.gtrg (string)`

Returns a string containing the title of the target window.

Returns the title of the frontmost window if the target window hasn't been set.

Returns the empty string if no windows are open.

`string app.setTargetWindow (string)`

`string app1.strg (string)`

Takes one parameter, a string indicating the title of the target window. All subsequent verbs apply to the target window.

Special case: if the title is empty, the target is set to the name of the frontmost window.

Returns the title of the target window. If it failed, the empty string is returned.

Basic Window Operations

`string app.newWindow (string)`

`string app1.nwin (string)`

Creates a new catalog window with the indicated title, and an empty catalog file on disk. If a catalog file is already open, it is automatically closed.

Takes one parameter, a string indicating the title of the new window. If the string is empty, an untitled window is created.

Returns the actual title of the new window.

Note: If a window with the title is already open, the application comes to the front and displays an alert dialog box; and the empty string is returned. This alert dialog can be enabled or disabled using the `enableDialogs` verb.

bool app.openWindow (path)

bool app1.owin (path)

Opens the file indicated by the path. Returns true if it worked, and sets the target window to the newly opened window. The file must be a movie or catalog file for the open to succeed. If a movie is opened, it may begin playing, and automatically close when finished, depending on the preference settings. Since only one movie file and one catalog file may be open at a time, opening a movie closes any previously open movie, and opening a catalog closes any previously open catalog.

If the path matches the name of a window title, the verb just brings the window to the front and returns true.

The title of the window may be the name of the file, or for a movie opened by opening a catalog entry, the "title" field in the catalog entry for the movie.

Returns false if there wasn't enough memory to create a new window or if there was an error loading the file.

bool app.closeWindow ()

bool app1.cwin ()

Close the target window. Returns True, unless there is no target window.

bool app.moveWindow (rect)

bool app1.mwin (rect)

Moves and resizes the target window so that it covers the rectangle.

Returns false if there is no target window.

You can determine a window's position by calling getWindowPosition.

bool app.selectWindow (string)

bool app1.xwin (string)

Brings the window with the indicated title to the front, and sets the target window to that window. Returns true if it worked, false otherwise.

Getting Info about a Window

string app.getFilePath ()

string app1.gpth ()

Returns the file name the file being displayed in the target window. Note the usual function of this verb is to return the full path, but in EasyPlay only the file name is returned.

Returns the empty string if the target window is a new window that wasn't loaded from a file and hasn't been saved.

`rect app.getWindowPosition ()`

`rect app1.gwps ()`

Returns a rectangle describing the position of the target window on the desktop.

`bool app.madeChanges ()`

`bool app1.chgs ()`

Returns true if the target window has been changed since the last save, false otherwise. Since EasyPlay automatically saves changes to a catalog as they are made, this verb always returns False.

Moving Text & Graphics

`pict app.getPicture ()`

`pict app1.gpic ()`

Returns the selected items in the target window as a picture. The type of the returned value is 'PICT'. If the target is a movie, it returns a picture of the currently displayed frame. If the target is a catalog window, it returns the thumbnail picture for the currently selected entry.

`bool app.selectAll ()`

`bool app1.sela ()`

Selects all the items in the target window. Has the same effect as the "Select All" command in the "Edit" menu. For example if a movie window is selected, the entire movie is selected.

Returns true if something is selected, false otherwise.

`bool app.somethingSelected ()`

`bool app1.hsel ()`

Returns true if something is selected in the target window. For a movie or catalog window, something is always selected so it returns True.

Controlling the Environment

`item app.countWindows ()`

`item app1.twin ()`

Returns the number of open windows.

`string app.nthWindow (item)`

`string app1.nthw (item)`

Returns a string, the title of the nth window. Windows are ordered from front to back, so the first window is the frontmost one. If n is greater than the number of windows, the empty string is returned.

`bool app.quit ()`

`bool app1.quit ()`

The application quits to the operating system if no changed windows are open. Since changes are automatically saved as they are made, changed windows are never open, so the Quit succeeds.

Returns true if it quit, false if it didn't.

Testing

`rect app.performanceTest (rect,rect)`

`rect app1.perf (rect,rect)`

This verb allows any application to participate in a performance benchmark of interapplication communication. It performs rectangle subtraction on two rectangle params, r1 and r2. It returns a rectangle that's the difference between each of the four fields (top, left, bottom, right) of the two rectangles.

It does no screen updating or disk operations, so if you call it repeatedly you'll get a pretty good measure of IAC performance.

EasyPlay Control

`EasyPlay.doMenu (item,item)`

`EZpa.menu (item)`

This verb can perform any menu command in EasyPlay. The first item is a number indicating the menu, and item is a number indicating the item in the menu, the first item being item 1. The possible values for the ID are:

- 128 Apple menu**
- 129 File menu**
- 130 Edit menu**
- 131 Movie menu**
- 132 Window menu**
- 133 Catalog menu**

In an event, the direct parameter is a longword with the ID in the high word and the item in the low word.

Movie Events

EasyPlay only allows one movie to be open at a time. These events operate on the movie file that is currently open.

fixed EasyPlay.getMovieDuration ()

fixed EZpa.mgdu ()

Returns the duration of the currently open movie in seconds, as a fixed point decimal value.

fixed EasyPlay.getMoviePosition ()

fixed EZpa.mgtm ()

Returns the position of the currently open movie. The position is expressed as a fixed point value in seconds, with 0 being the start of the movie.

EasyPlay.setMoviePosition (fixed)

EZpa.mstm (fixed)

Sets the position of the currently open movie. Pos is expressed as a fixed point value in seconds, with 0 being the start of the movie. If the position is out of range, it is set to the start or end of the movie as appropriate.

EasyPlay.playMovie (bool)

EZpa.mply (bool)

Play or stop the currently open movie. go is a boolean value. If it is true, the movie begins (or continues) playing at its current position. If it is false, the movie stops playing.

EasyPlay.isPlaying ()

EZpa.mgip ()

Returns the playing status of the currently open movie. If the movie played to its finish, the result is false. If the movie has not finished, it is either paused (by user action or the playMovie event), or it is still playing, and the result is true. If there is no movie open at all, the result is false.

EasyPlay.stepMovie (item)

EZpa.mstp (item)

Steps the movie forward or back the specified number of frames. If frames is positive, it moves forward the specified number of frames and then stops. If negative, it moves backward. If zero, it moves forward one frame. The direct parameter is a fixed point value specifying the number of frames to move.

pict EasyPlay.getMoviePoster ()

pict EZpa.mgmp ()

Returns a picture containing the poster frame for the currently open movie. The returned type is 'PICT. To get a picture of a particular movie frame, use the app1.getPicture event.

Catalog Events

Catalog files are kept up to date as changes are made - EasyPlay has no "Save" command. If you change a catalog entry, the change is immediately saved to the disk, if the catalog file can be written to. If not, the appropriate error is returned.

Most of these events work on the currently selected catalog entry in the currently open catalog file.

EasyPlay.selectEntry (item)

EZpa.cstg (item)

Sets the selection in the catalog. The catalog window is redrawn to reflect the new selected entry. The item is an integer specifying the entry to be selected, 0 being the first.

item EasyPlay.selectedEntry ()

item EZpa.cgtg ()

Returns an integer describing the selected catalog entry, 0 being the first.

item EasyPlay.entryCount ()

item EZpa.cgec ()

Returns an integer with the number of entries in the currently open catalog.

bool EasyPlay.getEntryLooping ()

bool EZpa.eglp ()

Returns a boolean value containing the looping property of the currently selected entry.

EasyPlay.setEntryLooping (bool)

EZpa.eslp (bool)

Sets the looping property of the currently selected catalog entry. If true, the movie will initially loop if opened through this catalog entry.

item EasyPlay.getEntryDepth ()

item EZpa.egmd ()

Returns an integer containing the monitor depth property of the currently selected entry.

EasyPlay.setEntryDepth (item)

EZpa.esmd (item)

Sets the monitor depth property of the currently selected catalog entry. The parameter is an integer specifying the depth property. This controls the color depth the screen is set to when the movie is played through this catalog entry. Since the user should always be in control of the depth through the “Monitors” control panel, the setting should be “No Change” unless the movie requires a specific depth. Valid values are:

- 1 No change - the suggested setting.
- 3 Black/White
- 4 4 grays
- 5 16 grays
- 6 256 grays
- 8 4 colors
- 9 16 colors
- 10 256 colors
- 11 Thousands
- 12 Millions

Preference Events

EasyPlay preferences are saved in a file in the System Folder. If you change a preference, the new setting is saved until it is changed again, by the user or another event.

bool EasyPlay.getMenubarPref ()

bool EZpa.pgmb ()

Returns a boolean value containing the current “Always use screen with menu bar” preference.

EasyPlay.setMenubarPref (bool)

EZpa.psmb (bool)

Sets the “Always use screen with menu bar” preference. If true, movies are always opened on the screen with the menu bar, even if it is not the one set to the most colors.

bool EasyPlay.getNoAutoplayPref ()

bool EZpa.pgap ()

Returns a boolean value containing the current “Don’t automatically play movies when opened” preference.

EasyPlay.setNoAutoplayPref (bool)

EZpa.psap (bool)

Sets the “Don’t automatically play movies when opened” preference. If true, movies do not automatically start playing when they are opened.

bool EasyPlay.getHideCtrlPref ()

bool EZpa.pghc ()

Returns a boolean value containing the current “Hide controller for movies when opened” preference.

EasyPlay.setHideCtrlPref (bool)

EZpa.pshc (bool)

Sets the “Hide controller for movies when opened” preference. If true, a movie controller does not initially appear at the bottom of movie windows when they are opened.

bool EasyPlay.getNoAutoclosePref ()

bool EZpa.pgcf ()

Returns a boolean value containing the current “Don’t close movies when finished” preference.

EasyPlay.setNoAutoclosePref (bool)

EZpa.pscf (bool)

Sets the “Don’t close movies when finished” preference. If true, movie windows are not automatically closed when the movie finishes playing

bool EasyPlay.getCatalogPref ()

bol EZpa.pgcm ()

Returns a boolean value containing the current “Show catalog menu” preference.

EasyPlay.setCatalogPref (bool)

EZpa.pscm (bool)

Sets the “Show Catalog menu” preference. If true, the “Catalog” menu appears on the menu bar.

item EasyPlay.getBackgroundPref ()

item EZpa.pgbg ()

Returns an integer describing the current window background preference.

EasyPlay.setBackgroundPref (item)

EZpa.psbg (item)

Sets the window background preference. The item is an integer with the preference setting. This controls the presence and color of the background around a movie in movie windows. The valid values are:

- 1 None. Open a window the same size as the movie image.
- 3 Black background.
- 4 Gray background.
- 5 White background.

item EasyPlay.getSizePref ()

item EZpa.pgds ()

Returns an integer describing the current default movie size preference.

EasyPlay.setSizePref (item)

EZpa.psds (item)

Sets the movie default size preference. The item is an integer with the preference setting. This controls the default size of movie images when movies are opened. Note that if a movie is opened by opening a catalog entry, the catalog entry may override the default. The valid values are:

- 1 Half normal
- 2 Normal
- 3 Double normal
- 4 Largest that maintains movie aspect ratio
- 5 Full screen