



# Aperture 3 AppleScript Reference

November 2009

# Contents

<b>Page 4</b>	<b>Introduction</b>
<b>Page 5</b>	<b>Classes</b>
	Album
	Application
	Container
	Custom Tag
	EXIF Tag
	Export Setting
	File Naming Policy
	Folder
	Folder Naming Policy
	Image Adjustment Preset
	Image Version
	IPTC Tag
	Keyword
	Library
	Other Tag
	Project
	Subfolder
	Trash-Folder
<b>Page 17</b>	<b>Commands</b>
	Adjust Image Date
	Apply Image Adjustment
	Duplicate
	Empty
	Export
	Import
	Move
	Refresh Metadata
	Reset All Image Adjustments
	Restore
	Reveal
<b>Page 27</b>	<b>Enumerations</b>
	Color Labels
	Import Options
	Metadata Types

<b>Page 29</b>	<b>Events</b> ImportActionForVersions
<b>Page 31</b>	<b>Appendix A</b> Containment Diagram Inheritance Diagram
<b>Page 33</b>	<b>Appendix B</b> EXIF Tag Name Mapping IPTC Tag Name Mapping Other Tag Name Mapping

# Introduction

This reference document is a guide to using AppleScript with Aperture 3. This document describes classes, commands, and other AppleScript-specific features found in the Aperture 3 AppleScript dictionary.

## Who Should Read This Document?

You should read this document if you want to automate tasks in Aperture 3.

This document assumes you are familiar with information found in the *AppleScript Language Guide* and the *Aperture 3 User Manual*. Also, the AppleScript examples included in this guide apply only to Aperture 3.

## See Also

These Apple documents provide additional information for working with AppleScript:

- See *Getting Started with AppleScript* for a guided introduction to AppleScript.
- See the *AppleScript Language Guide* for a comprehensive guide to AppleScript.

For information about using Aperture 3, refer to the *Aperture 3 User Manual* and visit <http://www.apple.com/aperture>.

# Classes

This chapter contains information about classes found in the Aperture 3 AppleScript dictionary. Each class is presented with information about the class, properties and elements of the class, sample code if appropriate, and other information.

## Album

**album** (noun, pl. **albums**): This class represents an album, Smart Album, Light Table album, web gallery album, web journal album, slideshow album, or book album within an Aperture library. Although only regular albums can be created using AppleScript, other types of albums can be accessed using AppleScript.

### Properties

Property	Access	Type	Description
id	get	Unicode text	Unique identifier of the album.
name	get/set	Unicode text	Name of the album.
parent	get	container	Parent container of the album.

### Elements

Element	Access	Refer to...
container	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
image version	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.

### Example

The following script creates a new album, "my album", at the library level.

```
tell application "Aperture"
  tell library 1
    make new album with properties {name:"my album"}
  end tell
end tell
```

## Application

**application** (noun): This class represents the Aperture application.

### Properties

Property	Access	Type	Description
frontmost	get	Boolean	Indicates whether the application is the frontmost application.
fullscreen	get/set	Boolean	Indicates whether Aperture is in Full Screen view. Also, instructs Aperture to enter or leave Full Screen view.
name	get	Unicode text	Name of the application.
selection	get	list of image versions	The image versions currently selected in the Aperture Browser.
tasks	get	list of Unicode text	List of application tasks.
trash	get	trash-folder	The Aperture Trash.
version	get	Unicode text	The version of Aperture.

### Elements

Element	Access	Refer to...
album	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
export setting	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
file naming policy	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
folder	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
folder naming policy	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
image adjustment preset	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
image version	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
library	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
project	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
subfolder	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
window	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.

## Example

The following script directs Aperture to enter Full Screen view, and then it reveals each image version in the selection, with a 5-second delay between each image. When all images have been revealed, the script directs Aperture to exit Full Screen view.

```
tell application "Aperture"
    activate
    set imageSel to (get selection)
    if imageSel is {} then
        error "Please select an image."
    else
        set fullscreen to true
        repeat with i from 1 to count of imageSel
            reveal {item i of imageSel}
            delay 5
        end repeat
        set fullscreen to false
    end if
end tell
```

## Container

**container** (noun, pl. **containers**): An item that contains other items. In Aperture, this class refers generically to various containers: album, folder, library, project, subfolder, and trash-folder.

### Properties

Property	Access	Type	Description
id	get	Unicode text	Unique identifier of the container.
name	get/set	Unicode text	Name of the container.
parent	get	container	Parent container of the container.

### Elements

Element	Access	Refer to...
container	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
image version	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.

## Custom Tag

**custom tag** (noun, pl. **custom tags**): This class represents a custom metadata tag of an image version.

### Properties

Property	Access	Type	Description
name	get	Unicode text	Name of the custom tag.
value	get/set	Unicode text	Value of the custom tag.

## Example

The following script directs Aperture to create a custom tag, “custom name”, for each of the selected images and provides a value for that custom tag. If a value for a particular tag does not already exist for an image version, use the `make` command. If you want to change an existing tag’s value, use the `set` command.

```
tell application "Aperture"
    set imageSel to (get selection)
    if imageSel is {} then
        error "Please select an image."
    else
        repeat with i from 1 to count of imageSel
            tell library 1
                tell item i of imageSel
                    make new custom tag with properties {name:"custom
name", value:"custom value"}
                end tell
            end tell
        end repeat
    end if
end tell
```

## EXIF Tag

**EXIF tag** (noun, pl. **EXIF tags**): This class represents an EXIF tag for an image version.

### Properties

Property	Access	Type	Description
name	get	Unicode text	Name of the EXIF tag.
value	get	real   integer   Unicode text   date   Boolean   anything	Value of the EXIF tag.

## Example

The following script returns a list of available EXIF tags for a particular image version

```
tell application "Aperture"
    tell library 1
        tell project 1
            tell image version 1
                get name of every EXIF tag
            end tell
        end tell
    end tell
end tell
```

## Export Setting

**export setting** (noun, pl. **export settings**): A collection of settings that describe an export operation for an image. Export settings can be created and managed in the Aperture interface.

### Properties

Property	Access	Type	Description
name	get	Unicode text	Name of the export setting.

## File Naming Policy

**file naming policy** (noun, pl. **file naming policies**): This class represents a collection of settings that describe the filename format for an exported image. Filenaming policies can be created and managed in the Aperture interface.

### Properties

Property	Access	Type	Description
name	get	Unicode text	Name of the filenaming policy.

## Folder

**folder** (noun, pl. **folders**): This class represents a folder within an Aperture library.

### Properties

Property	Access	Type	Description
id	get	Unicode text	Unique identifier of the folder.
name	get/set	Unicode text	Name of the folder.
parent	get	container	Parent container of the folder.

### Elements

Element	Access	Refer to...
album	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
container	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
folder	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
image version	get/get/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
project	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.

## Example

The following script creates a new folder, "my folder", at the library level,

```
tell application "Aperture"
  tell library 1
    make new folder with properties {name:"my folder"}
  end tell
end tell
```

## Folder Naming Policy

**folder naming policy** (noun, pl. **folder naming policies**): This class represents a collection of settings that describe the folder name format for an exported image. Folder naming policies can be created and managed in the Aperture interface.

### Properties

Property	Access	Type	Description
name	get	Unicode text	Name of the folder naming policy.

## Image Adjustment Preset

**image adjustment preset** (noun, pl. **image adjustment presets**): This class represents an image adjustment preset. Image adjustment presets can be created and managed in the Aperture interface. See the information about the `Apply Image Adjustment` command for an example that illustrates the usage of this class.

### Properties

Property	Access	Type	Description
container	get	Unicode text	Preset group name.
id	get	Unicode text	Unique identifier of the image adjustment preset.
name	get	Unicode text	Name of the image adjustment preset.

## Image Version

**image version** (noun, pl. **image versions**): This class represents an Aperture image version.

## Properties

Property	Access	Type	Description
color label	get/set	color labels	The color label of the image version.
flagged	get/set	Boolean	Indicates whether an image version has its flag set.
height	get	integer	Height of the image version in pixels.
id	get	Unicode text	Unique identifier of the image version.
latitude	get/set	real	The GPS latitude in the range of -90.0 to 90.0.
longitude	get/set	real	The GPS longitude in the range of -180.0 to 180.0. Measurement is taken from the prime meridian. "West" longitudes are negative numbers.
main rating	get/set	integer	Image version's rating.
name	get/set	Unicode text	Name of the image version.
online	get	Boolean	Indicates whether an image version is online or offline.
parent	get	container	Parent container of the image version.
picked	get	Boolean	Indicates whether an image version is the stack pick for its stack.
referenced	get	Boolean	Indicates whether an image version's master file is referenced or managed within Aperture.
selected	get	Boolean	Indicates whether an image version is selected in the Aperture Browser.
width	get	integer	Width of the image version in pixels.

## Elements

Element	Access	Refer to...
custom tag	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
EXIF tag	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
IPTC tag	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
keyword	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
other tag	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.

## Example

The following script directs Aperture to set the main rating of every image within a project to five stars.

```
tell application "Aperture"
  tell library 1
    tell project "my project"
      tell every image version to set main rating to 5
    end tell
  end tell
end tell
```

## IPTC Tag

**IPTC tag** (noun, pl. **IPTC tags**): This class represents an IPTC tag for an image version.

### Notes

Aperture stores an image version's ID into the value for IPTC Special Instructions for preview files and no other file. This allows a complete roundtrip for images between Aperture and other applications such as Pages and Keynote.

### Properties

Property	Access	Type	Description
name	get	Unicode text	Name of the IPTC tag.
value	get/set	Unicode text	Value of the IPTC tag.

### Example

The following script directs Aperture to create an IPTC tag, "DateCreated", for each of the selected images and provides a value for that IPTC tag. If a value for a particular tag does not already exist for an image, use the `make` command. If you want to change an existing tag's value, use the `set` command.

```
tell application "Aperture"
    set imageSel to (get selection)
    if imageSel is {} then
        error "Please select an image."
    else
        repeat with i from 1 to count of imageSel
            tell library 1
                tell item i of imageSel
                    make new IPTC tag with properties ↵
                    {name:"DateCreated", value:"20090819T194049-0700"}
                end tell
            end tell
        end repeat
    end if
end tell
```

## Keyword

**keyword** (noun, pl. **keywords**): This class represents a keyword in Aperture and is the same as an IPTC keyword.

### Properties

Property	Access	Type	Description
id	get	Unicode text	Unique identifier of the keyword.
name	get	Unicode text	Name of the keyword.
parents	get	Unicode text	Tab-separated list of parent keywords.

### Example

The following script directs Aperture to create and assign a keyword, “Graduation”, to an image version. Once assigned to an image version, the “Graduation” keyword appears under the parent keyword, “Personal”.

```
tell application "Aperture"
  tell library 1
    tell project 1
      tell image version 1
        make new keyword with properties {name:"Graduation",
          parents:{"Personal"}}
      end tell
    end tell
  end tell
end tell
```

## Library

**library** (noun, pl. **libraries**): This class represents an Aperture library.

### Properties

Property	Access	Type	Description
id	get	Unicode text	Unique identifier of the library.
name	get/set	Unicode text	Name of the Aperture library.
parent	get	container	Parent container of the library.

### Elements

Element	Access	Refer to...
album	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
container	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
folder	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
image version	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
project	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.

## Other Tag

**other tag** (noun, pl. **other tags**): This class represents metadata unique to Aperture for an image version. The Aperture interface refers to these tags as “Aperture” metadata.

### Properties

Property	Access	Type	Description
name	get	Unicode text	Name of the other tag.
value	get	integer   Unicode text   date   Boolean   anything	Value of the other tag.

### Example

The following script returns a list of available other `tags` for a particular image version. If a value for a particular tag does not already exist for an image, use the `make` command. If you want to change an existing tag’s value, use the `set` command.

```
tell application "Aperture"
  tell library 1
    tell project 1
      tell image version 1
        get name of every other tag
      end tell
    end tell
  end tell
end tell
```

## Project

**project** (noun, pl. **projects**): This class represents a project within Aperture.

### Properties

Property	Access	Type	Description
id	get	Unicode text	Unique identifier of the project.
name	get/set	Unicode text	Name of the project.
parent	get	container	Parent container of the project.

### Elements

Element	Access	Refer to...
album	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
container	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
image version	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
subfolder	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.

## Example

The following script creates a new project, “my project”, at the library level.

```
tell application "Aperture"
    tell library 1
        make new project with properties {name:"my project"}
    end tell
end tell
```

## Subfolder

**subfolder** (noun, pl. **subfolders**): This class represents a subfolder within an Aperture project or subfolder.

## Notes

Once a subfolder is moved to the root level or becomes a subitem of a folder, its class changes to that of a folder.

## Properties

Property	Access	Type	Description
id	get	Unicode text	Unique identifier of the subfolder.
name	get/set	Unicode text	Name of the subfolder.
parent	get	container	Parent container of the subfolder.

## Elements

Element	Access	Refer to...
album	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
container	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
subfolder	get/make/delete	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.

## Example

The following script creates a new subfolder, “my subfolder”, as a child item to a project.

```
tell application "Aperture"
    tell library 1
        set myProj to make new project with properties {name:"my project"}
        tell myProj to make new subfolder with properties {name:"my subfolder"}
    end tell
end tell
```

## Trash-Folder

**trash-folder** (noun): This class represents the Aperture Trash within an Aperture library.

### Properties

Property	Access	Type	Description
id	get	Unicode text	Unique identifier of the Aperture Trash.
name	get/set	Unicode text	Name of the Aperture Trash.
parent	get	container	Parent container of the Aperture Trash.

### Elements

Element	Access	Refer to...
container	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.
image version	get	By name, by index, by range, relative to others, by whose/where filter, and by unique ID.

# Commands

This chapter provides a reference for the commands found in the Aperture 3 AppleScript dictionary. This chapter lists each command alphabetically, along with a brief description of the command, parameters, syntax, and, if appropriate, a sample script illustrating the command's usage.

## Adjust Image Date

**adjust image date** (verb): This command adjusts an image version's EXIF Image Date value. The command also provides the option to write the date value to the master file.

### Syntax

```
set theResult to adjust image date date of images {image version,  
...} masters include Boolean
```

### Result

Boolean

### Parameters

Parameter	Required	Type	Description
direct parameter	required	date	Specified date.
masters included	optional	Boolean	Additionally, change the master date while updating image versions.
of images	required	list of image versions	List of image versions to adjust.

## Example

The following script adjusts the EXIF image date of a selection of image versions by decreasing the hour by one.

```
tell application "Aperture"
    set imageSel to (get selection)
    repeat with i from 1 to count of imageSel
        tell item i of imageSel
            set imageDate to value of EXIF tag "ImageDate"
        end tell
        set imageDate to imageDate - (1 * hours)
        adjust image date imageDate of images {item i of imageSel}
    end repeat
end tell
```

## Apply Image Adjustment

**apply image adjustment** (verb): This command applies an image adjustment preset to an image version.

### Syntax

```
apply image adjustment image version preset image adjustment  
preset replacing existing adjustments Boolean
```

### Parameters

Parameter	Required	Type	Description
direct parameter	required		
preset	required	image adjustment preset	An image adjustment preset.
replacing existing adjustments	optional	Boolean	Replaces existing adjustments with the adjustments in the specified preset. By default, adjustments are added to the existing set.

## Example

The following script applies an image adjustment preset to all images in a selection that have an EXIF ISOSpeedRating value greater than 800.

```
tell application "Aperture"
    set imageSel to (get selection)
    set adjustPreset to every image adjustment preset

    if imageSel is {} then
        error "Please select an image."
    else if adjustPreset is {} then
        error "Please create an adjustment preset."
    else
        repeat with i from 1 to count of imageSel
            tell (item i of imageSel)
                if value of EXIF tag "ISOSpeedRating" is greater than
200 then
                    apply image adjustment preset (item 1 of
adjustPreset)
                end if
            end tell
        end repeat
    end if
end tell
```

## Duplicate

**duplicate** (verb): This copies image versions from an album or project to an album or project.

### Syntax

`duplicate` *reference* to *location specifier*

### Parameters

Parameter	Required	Type	Description
direct parameter	required	reference	The image versions to copy.
to	required	location specifier	The destination of the copy command. Acceptable destinations are albums and projects within the current Aperture library.

## Example

The following script directs Aperture to duplicate all selected image versions to a project, “my project”. The script assumes a project, “my project”, already exists in the current Aperture library.

```
tell application "Aperture"
    set imageSel to (get selection)
    if imageSel is {} then
        error "Please select an image."
    else
        repeat with i from 1 to count of imageSel
            duplicate item i of imageSel to project "my project"
        end repeat
    end if
end tell
```

## Empty

**empty** (verb): This command empties the Aperture Trash.

### Syntax

```
empty reference
```

### Parameters

Parameter	Required	Type	Description
direct parameter	required	reference	Aperture Trash.

## Example

The following script directs Aperture to empty the Aperture Trash.

```
tell application "Aperture"
    empty trash
end tell
```

## Export

**export** (verb): This command exports image versions, masters, or projects to the specified location.

### Syntax

```
set theResult to export {anything, ...} naming folders with folder naming policy ↵
    using export setting metadata metadata types to anything ↵
    naming files with file naming policy consolidating images Boolean
```

### Result

list of exported image versions | list of exported projects

## Parameters

Parameter	Required	Type	Description
direct parameter	required	list of anything	List of image versions or projects to be exported.
consolidating images	optional	Boolean	The export command consolidates images into the exported project if set to true. By default, images are not consolidated.
metadata	optional	metadata types	Exports masters with either XMP sidecars or embedded IPTC metadata. See the metadata types enumeration for more information. This parameter does not apply to exporting versions.
naming files with	optional	filenaming policy	Specifies which filenaming policy to use when exporting images.
naming folders with	optional	folder naming policy	Specifies which folder naming policy to use when exporting images.
to	required	anything	The destination of the exported images or projects.
using	optional	export setting	Specifies which export setting to use when exporting images. Masters are exported when this parameter is omitted.

## Example

The following script exports selected images as JPEG versions and original masters with embedded IPTC data.

```
tell application "System Events"
    set exportFolder to (choose folder with prompt "Choose an export folder")
end tell

tell application "Aperture"
    set imageSel to (get selection)
    if imageSel is {} then
        error "Please select an image."
    else
        export imageSel naming folders with folder naming policy ~
            "Project Name" using export setting ~
            "JPEG - Original Size" to exportFolder
        export imageSel naming folders with folder naming policy ~
            "Project Name" to exportFolder metadata embedded
    end if
end tell
```

## Import

**import** (verb): This command imports images, projects, or libraries into the specified location within the current Aperture library.

### Syntax

```
set theResult to import {anything, ...} by import options into  
project | folder | library
```

### Result

list of image versions | list of Aperture 3 libraries | list of Aperture 1 or Aperture 2 projects

### Parameters

Parameter	Required	Type	Description
direct parameter	required	list of anything	The file system objects that will be imported. Only supported file formats are imported into the Aperture library. The file path to the file system objects can be posix paths, file aliases, or HFS Plus paths.
by	optional	import options	Specifies mode of import when importing images.
into	required	project   folder   library	The location within the current Aperture library into which to import. Images can only be imported into projects. Libraries can be imported into the currently open Aperture library at the root level or within a folder.

## Example

The following script prompts the user for a directory of images and then directs Aperture to import those images by reference into a project, "my project".

```
--ask user for a directory containing images.
tell application "System Events"
    set importFolder to (choose folder with prompt "choose")
    set importFiles to (get path of every file of importFolder where-
visible is true)
end tell

tell application "Aperture"
    activate
    tell library 1
        if not (exists project "my project") then
            set theProject to make new project with properties {
name:"my project"}
        else
            set theProject to project "my project"
        end if
    end tell

    --import by reference
    with timeout of 600 seconds
        import importFiles by referencing into theProject
    end timeout
end tell
```

## Move

**move** (verb): This command moves image versions from an album or project to an album or project.

### Syntax

*move reference to location specifier*

### Parameters

Parameter	Required	Type	Description
direct parameter	required	reference	The image versions to move.
to	required	location specifier	The destination of the move command.

## Example

The following script directs Aperture to move all selected image versions to a project, “my project.” The script assumes a project, “my project,” already exists in the current Aperture library.

```
tell application "Aperture"
    set imageSel to (get selection)
    if imageSel is {} then
        error "Please select an image."
    else
        repeat with i from 1 to count of imageSel
            move item i of imageSel to project "my project"
        end repeat
    end if
end tell
```

## Refresh Metadata

**refresh metadata** (verb): This command updates the EXIF metadata from the master file.

### Syntax

```
refresh metadata image version | {image versions, ...}
```

This command has three possible variations on its syntax:

1. `tell image version 1 to refresh metadata`
2. `image version 1 refresh metadata`
3. `refresh metadata of every image version of project 1`

### Parameters

Parameter	Required	Type	Description
direct parameter	required	image version   list of image versions	Image version(s).

## Reset All Image Adjustments

**reset all image adjustments** (verb): This command removes all image adjustments from an image version.

### Syntax

```
reset all image adjustments image version
```

### Parameters

Parameter	Required	Type	Description
direct parameter	required	image version	Image version.

### Example

The following script directs Aperture to remove all image adjustments from image version 1.

```
tell application "Aperture"  
    reset all image adjustments image version 1  
end tell
```

## Restore

**restore** (verb): This command restores a container from the Aperture Trash.

### Syntax

```
restore container | {containers, ...}
```

### Parameters

Parameter	Required	Type	Description
direct parameter	required	container   list of containers	The container(s) to restore from the Aperture Trash.

### Example

The following script demonstrates the usage of the `reveal` command.

```
tell application "Aperture"  
    restore every container in trash  
end tell
```

## Reveal

**reveal** (verb): This command reveals image versions or containers in the Aperture interface.

### Syntax

```
reveal {containers, ...} | {image versions, ...}
```

### Parameters

Parameter	Required	Type	Description
direct parameter	required	list of projects   list of albums   list of folders   list of subfolders   list of image versions	The list of image versions or containers.

### Example

The following script demonstrates the usage of the `reveal` command. Note that the `reveal` command responds only to lists, not single items.

```
tell application "Aperture"
    activate
    set imageSel to (get selection)
    if imageSel is {} then
        error "Please select an image."
    else
        repeat with i from 1 to count of imageSel
            reveal {item i of imageSel}
            delay 5
        end repeat
    end if
end tell
```

# Enumerations

This chapter provides a reference for the enumerations found in the Aperture 3 AppleScript dictionary. Each enumeration presented in this chapter is accompanied by a list of the available constants, a brief description of each constant, and a brief description how the enumeration is used.

## Color Labels

**color labels** (enumeration)

### Constants

Constant	Description
blue	Blue color label.
gray	Gray color label.
green	Green color label.
no color label	No color label.
orange	Orange color label.
purple	Purple color label.
red	Red color label.
yellow	Yellow color label.

### Where Used

The `color labels` enumeration is used in the following way:

color label property of the `image version` class

## Import Options

**import options** (enumeration)

### Constants

Constant	Description
copying	Import by copying files.
moving	Import by moving files.
referencing	Import by referencing files.

### Where Used

The `import options` enumeration is used in the following way:  
by parameter of the `import` command

## Metadata Types

**metadata types** (enumeration)

### Constants

Constant	Description
embedded	Embed IPTC metadata in exported masters.
sidecar	Create XMP sidecar files when exporting masters.

### Where Used

The `metadata types` enumeration is used in the following way:  
by parameter of the `export` command

# Events

This chapter provides a reference for the events found in the Aperture 3 AppleScript dictionary. This chapter provides a brief description of each event, the event's syntax, and a sample script illustrating how the event is used.

## ImportActionForVersions

**ImportActionForVersions** (event handler): This event is called at the end of an import using the Aperture Import browser. Scripts that contain this event handler can be chosen using the Actions controls in the Import Settings area of the Import browser. For more information about the Actions controls in Import Settings, see the *Aperture 3 User Manual*.

### Syntax

```
on importActionForVersions (image versions)  
    [ statement ]...  
end importActionForVersions
```

### Placeholders

*image versions*: A list of image versions that have just been imported into Aperture.

*statement*: Any AppleScript statement.

## Example

The following script demonstrates the use of the `ImportActionForVersions` event handler. Once executed, the script attempts to get the value of EXIF `ISOSpeedRating` from every image imported into Aperture and apply an adjustment preset depending on the value of EXIF `ISOSpeedRating`.

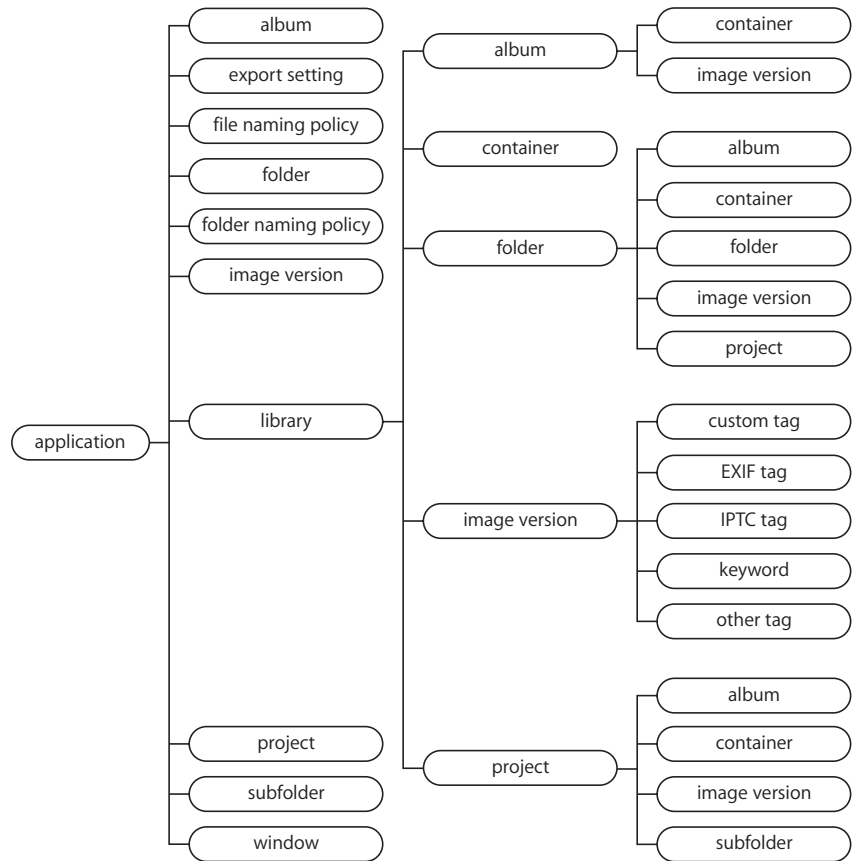
```
on ImportActionForVersions(input)
    tell application "Aperture"
        set adjustPreset to every image adjustment preset

        tell library 1
            repeat with i from 1 to count of input
                tell (item i of input)
                    try
                        if value of EXIF tag "ISOSpeedRating" is
greater than 800 then
                            apply image adjustment preset
(item 1 of adjustPreset)
                        else if value of EXIF tag "ISOSpeedRating" is
800 then
                            apply image adjustment preset
(item 2 of adjustPreset)
                        else if value of EXIF tag "ISOSpeedRating" is
400 then
                            apply image adjustment preset
(item 3 of adjustPreset)
                        else
                            apply image adjustment preset
(item 4 of adjustPreset)
                        end if
                    on error
                        log "image does not have a value for
ISOSpeedRating"
                    end try
                end tell
            end repeat
        end tell
    end tell
end ImportActionForVersions
```

# Appendix A

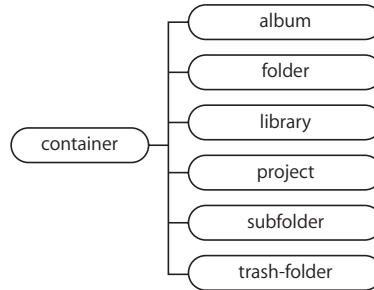
## Containment Diagram

The following diagram reveals the containment hierarchy found in the Aperture Suite of the Aperture 3 AppleScript dictionary.



## Inheritance Diagram

The following diagram reveals the inheritance hierarchy found in the Aperture Suite of the Aperture 3 AppleScript dictionary.



# Appendix B

## EXIF Tag Name Mapping

The following table provides a mapping between EXIF tag names found in the Aperture 3 interface and EXIF tag names that appear in AppleScript.

AppleScript tag name	Aperture tag name	Notes
Altitude	Altitude	Negative numbers represent values below sea level. Positive numbers represent values above sea level.
AltitudeRef		Combined with Altitude.
ApertureValue	Aperture	
Artist	Artist	
BrightnessValue	Brightness	
CameraSerialNumber	Serial Number	
CaptureDayOfMonth	Date	The day of the month separated from Date.
CaptureDayOfWeek	Date	The day of the week separated from Date.
CaptureHourOfDay	Date	The hour of the day separated from Date.
CaptureMinuteOfHour	Date	The minute of the hour separated from Date.
CaptureMonthOfYear	Date	The month of the year separated from Date.
CaptureSecondOfMinute	Date	The second of the minute separated from Date.
CaptureYear	Date	The year separated from Date.
ColorModel	Color Model	
ColorSpace	Color Space	
Contrast	Contrast	
Copyright	Copyright	
Depth	Depth	
ExifVersion	EXIF Version	
ExposureBiasValue	Exposure Bias	
ExposureMode	Exposure Mode	
ExposureProgram	Exposure Program	
Firmware	Firmware	
Flash	Flash	
FlashExposureComp	Flash Exposure Compensation	
FlashPixVersion	Flash Pixel Version	
FocalLength	Focal Length	
FocalLenIn35mmFilm	Focal Length (35mm)	

AppleScript tag name	Aperture tag name	Notes
FocusDistance	Focus Distance	
FocusMode	Focus Mode	
GPSTime	GPS Time	
GPSTimeZone	GPS Time Zone	
ImageDate	Date	
ImageDescription	Image Description	
ISOSpeedRating	ISO	
Latitude	Latitude	
LensMaxMM	Lens Maximum (mm)	
LensMinMM	Lens Minimum (mm)	
LensModel	Lens	
LightSource		Displayed in White Balance.
Longitude	Longitude	
Make	Camera Make	
MapDatum	Map Datum	
MaxApertureValue	Max. Lens Aperture	
MeasuredEV	Measured EV	
MeteringMode	Metering Mode	
Model	Camera Model	
NikonFlashSetting	Nikon Flash Setting	
NikonFocusMode	Nikon Focus Mode	
NikonQuality	Nikon Quality	
NikonSharpenMode	Nikon Sharpen Mode	
NikonWhiteBalanceMode	Nikon White Balance	
OwnerName	Owner Name	
PixelHeight	Pixel Height	
PixelWidth	Pixel Width	
ProfileName	Profile Name	
Saturation	Saturation	
SceneCaptureType	Scene Capture Type	
Sharpness	Sharpness	
ShutterSpeed	Shutter Speed	
Software	Software	
SubjectDistance	Subject Distance	
UserComment	User Comment	
WhiteBalance	White Balance	
WhiteBalanceIndex		Displayed in White Balance.

## IPTC Tag Name Mapping

The following table provides a mapping between IPTC tag names found in the Aperture 3 interface and IPTC tag names that appear in AppleScript.

AppleScript tag name	Aperture tag name	Notes
ActionAdvised	Action Advised	
AudioDuration	Audio Duration	
AudioOutcue	Audio Outcue	
AudioSamplingRate	Audio Sampling Rate	
AudioSamplingRes	Audio Sampling Resolution	
AudioType	Audio Type	
Byline	Creator	

AppleScript tag name	Aperture tag name	Notes
BylineTitle	Creator's Job Title (Job Title)	
Caption/Abstract	Caption	
Category	Category	
Contact	Contact	
Contact Address	Contact Address (Address)	
Contact City	Contact City (City)	
Contact Country	Contact Country (Country)	
Contact Email(s)	Contact Email(s) (Email)	
Contact Phone(s)	Contact Phone(s) (Phone)	
Contact Postal Code	Contact Postal Code (Postal Code)	
Contact State/Province	Contact State/Province (State/Province)	
Contact Website(s)	Contact Website(s) (Website)	
ContentLocationCode	Content Location Code	
ContentLocationName	Content Location Name	
CopyrightNotice	Copyright Notice	
Country/PrimaryLocationCode	Image Country Code (ISO Country Code)	
Country/PrimaryLocationName	Image Country (Country)	
Credit	Provider	
DateCreated	Date Created	
DigitalCreationDate	Digital Creation Date	
DigitalCreationTime		Combined with Digital Creation Date.
EditorialUpdate	Editorial Update	
EditStatus	Edit Status	
ExpirationDate	Expiration Date	
ExpirationTime		Combined with Expiration Date.
FixtureIdentifier	Fixture Identifier	
Headline	Content Headline (Headline)	
Image City	Image City (City)	In previous versions of Aperture, this tag was named "City".
ImageOrientation	Image Orientation	
ImageType	Image Type	
Keywords	Keywords	
LanguageIdentifier	Language Identifier	
ObjectAttributeReference	Intellectual Genre	
ObjectCycle	Object Cycle	
ObjectName	Title	
ObjectTransmissionReference	Job Identifier	
ObjectTypeReference	Object Type Reference	
OriginatingProgram	Originating Program	
ProgramVersion	Program Version	
Province/State	State/Province	
ReferenceDate	Reference Date	
ReferenceNumber	Reference Number	
ReferenceService	Reference Service	
ReleaseDate	Release Date	
ReleaseTime		Combined with Release Date.
Scene	IPTC Scene	
Source	Source	
SpecialInstructions	Special Instructions	

AppleScript tag name	Aperture tag name	Notes
SubjectReference	IPTC Subject Code	
SubLocation	Image Location (Location)	
SupplementalCategory	Supplemental Category	
TimeCreated		Combined with Date Created.
Urgency	Urgency	
UsageTerms	Usage Terms	
Writer/Editor	Caption Writer	

## Other Tag Name Mapping

The following table provides a mapping between Other tag names found in the Aperture 3 interface and Other tag names that appear in AppleScript. The Aperture interface refers to these tags as “Aperture” metadata.

AppleScript tag name	Aperture tag name	Notes
AspectRatio	Aspect Ratio	
ColorLabel	Label	
FileName	File Name	
FileSize	File Size	
Flagged	Flag	
ImportGroup	Import Session	
LastModifiedDate	Last Modified Date	
MainRating	Rating	
MasterLocation	Project Path	
MasterProject		Available only in AppleScript.
Orientation	Orientation	
PixelSize	Master Pixel Size	Pixel size of the master image.
ProcessedPixelSize	Pixel Size	Pixel size of the cropped image version.
ProjectName	Project Name	
VersionName	Version Name	

Copyright © 2009 Apple Inc. All rights reserved.

Apple, the Apple logo, AppleScript, Keynote, and Pages are trademarks of Apple Inc., registered in the U.S. and other countries. Aperture is a trademark of Apple Inc. Other product and company names mentioned herein may be trademarks of their respective companies. Product specifications are subject to change without notice.

November 2009 019-1580