

[Core Spotlight](#) / `CSSearchableItemAttributeSet`

Class

CSSearchableItemAttributeSet

The detailed metadata for a searchable item.

iOS 9.0+ | iPadOS 9.0+ | Mac Catalyst 13.1+ | macOS 10.11+ | visionOS 1.0+

```
class CSSearchableItemAttributeSet
```

Mentioned in

- Adding your app's content to Spotlight indexes
- Searching for information in your app

Overview

A `CSSearchableItemAttributeSet` contains an extensive set of attributes that describe your app's content. Attributes include information such as its title and a brief description. They can also refer to who created the item, what kind of data it represents, when someone created it, and more. During the indexing process, you create [CSSearchableItem](#) objects and use a `CSSearchableItemAttributeSet` to fill in the attributes for that item. During a search, you can query the index for items with attributes that match specific values.

When creating a [CSSearchableItem](#), it's important to fill out as much information in the accompanying `CSSearchableItemAttributeSet` object as possible. You don't have to provide values for every attribute. Instead, choose attributes that match the domain of your content. This type divides attributes into groups such as media, documents, events, places, music, images, and more. You can also add custom attributes to describe new types of content. When defining custom attributes, be as specific as possible in your definition, and provide a value for the [contentTypeTree](#) property so your custom attribute inherits from a known type.

Important

Modify a `CSSearchableItemAttributeSet` object on only one thread at a time. Concurrent access to properties in an attribute set has undefined behavior.

Topics

Getting an attribute set

```
init(contentType: UTType)
```

Creates an attribute set for the specified content type.

Working with custom attributes

```
func setValue((any NSSecureCoding)?, forCustomKey: CSCustomAttributeKey)  
)
```

Sets the value for a custom attribute key.

```
func value(forCustomKey: CSCustomAttributeKey) -> (any NSSecureCoding)?
```

Returns the value associated with the specified custom attribute key.

Describing Apple Intelligence prioritization and summarization

```
var isPriority: NSNumber?
```

A Boolean value that indicates whether the mail or messages content represents a prioritized item.

```
var textContentSummary: String?
```

A string that presents the Apple Intelligence summarization of the item.

```
var transcribedTextContent: String?
```

A string that represents the text the system transcribed.

Describing documents

```
var audiences: [String]?
```

A class of entity for which the item is intended or useful.

`var contentDescription: String?`

A description of the item's content.

`var creator: String?`

The name of the app that created the content.

`var encodingApplications: [String]?`

The name of the apps that converted the original content into a PDF stream.

`var fileSize: NSNumber?`

The size of the document file.

`var fontNames: [String]?`

An array of font names the document uses.

`var identifier: String?`

A formal identifier that references the document the item represents.

`var kind: String?`

A description of the kind of document the item represents.

`var pageCount: NSNumber?`

The number of pages in the document.

`var pageHeight: NSNumber?`

The height of the document page, in points (72 points per inch).

`var pageWidth: NSNumber?`

The width of the document page, in points (72 points per inch).

`var securityMethod: String?`

The security method (a type of encryption) that protects the document file.

`var subject: String?`

The subject of the document.

`var theme: String?`

The theme of the document.

Describing general attributes

`var alternateNames: [String]?`

An array of localized strings that represent alternate display names for the item.

`var contentType: String?`

The uniform type identifier (UTI) of the item.

`var contentTypeTree: [String]?`

An attribute type that identifies a custom hierarchy of types to describe the attributes of your item.

`var contentURL: URL?`

The file URL of the content to index.

`var darkThumbnailURL: URL?`

The local file URL of the thumbnail image for the item when Dark Mode is active.

`var displayName: String?`

A localized string that contains the name of the item, suitable to display in the user interface.

`var keywords: [String]?`

An array of keywords associated with the item, such as work, birthday, important, and so on.

`var metadataModificationDate: Date?`

The date on which the last metadata attribute was changed.

`var path: String?`

The complete path to the item.

`var rankingHint: NSNumber?`

A number that indicates the relative importance of the item among other items from the app.

`var relatedUniqueIdentifier: String?`

The unique identifier for the item to which the activity is related.

`var thumbnailData: Data?`

Image data that represents the thumbnail of the item.

`var thumbnailURL: URL?`

The local file URL of the thumbnail image for the item.

`var title: String?`

The title of the item.

`var domainIdentifier: String?`

An identifier that represents the domain or owner of the item.

```
var weakRelatedUniqueIdentifier: String?
```

The unique identifier for the item to which the activity is related, but not linked.

Describing user involvement

```
var userCreated: NSNumber?
```

A value that indicates the user created the item.

```
var userCurated: NSNumber?
```

A value that indicates the user selected the item.

```
var userOwned: NSNumber?
```

A value that indicates the user purchased or owns the item.

Describing events

```
var allDay: NSNumber?
```

A value that indicates if the event covers an entire day.

```
var completionDate: Date?
```

The date on which the item was completed.

```
var dueDate: Date?
```

The date on which the item is due.

```
var endDate: Date?
```

The end date for the item.

```
var importantDates: [Date]?
```

An array of important dates associated with the item.

```
var startDate: Date?
```

The start date for the item.

Describing places

```
var altitude: NSNumber?
```

The altitude of the item in meters above sea level, expressed using the WGS84 datum.

`var city: String?`

The city of the item's origin according to guidelines that the provider establishes.

`var country: String?`

The full, publishable name of the country or region in which the intellectual property of the item was created, according to guidelines the provider establishes.

`var gpsAreaInformation: String?`

Information about the GPS area.

`var gpsdop: NSNumber?`

The GPS dilution of precision value.

`var gpsDateStamp: Date?`

The date and time related to the GPS value.

`var gpsDestBearing: NSNumber?`

The bearing to the destination point.

`var gpsDestDistance: NSNumber?`

The distance to the destination point.

`var gpsDestLatitude: NSNumber?`

The latitude of the destination point.

`var gpsDestLongitude: NSNumber?`

The longitude of the destination point.

`var gpsDifferential: NSNumber?`

The differential correction applied to the GPS receiver.

`var gpsMapDatum: String?`

The geodetic data that the GPS receiver uses.

`var gpsMeasureMode: String?`

The measurement precision mode in use by the GPS receiver.

`var gpsProcessingMethod: String?`

The location finding method that the GPS receiver uses.

`var gpsStatus: String?`

The status of the GPS receiver.

`var gpsTrack: NSNumber?`

The direction of travel of the item in degrees from true north.

`var headline: String?`

A publishable string that provides a synopsis of the contents of the item.

`var imageDirection: NSNumber?`

The direction of the item's image in degrees from true north.

`var instructions: String?`

Instructions that concern the use of the item, such as an embargo or warning.

`var latitude: NSNumber?`

The latitude of the item, in degrees north of the equator, expressed using the WGS84 datum.

`var longitude: NSNumber?`

The longitude of the item, in degrees east of the prime meridian, expressed using the WGS84 datum.

`var namedLocation: String?`

The name of the location or point of interest associated with the item.

`var speed: NSNumber?`

The speed of the item, in kilometers per hour.

`var stateOrProvince: String?`

The province or state of origin according to guidelines the provider establishes.

`var timestamp: Date?`

The timestamp on the item.

`var fullyFormattedAddress: String?`

The fully formatted address of the item, received from MapKit.

`var postalCode: String?`

The postal code for the item according to guidelines the provider establishes.

`var subThoroughfare: String?`

The sublocation, such as a street number, for the item according to guidelines the provider establishes.

`var thoroughfare: String?`

The thoroughfare, such as a street name, associated with the location for the item according to guidelines the provider establishes.

Describing media

`var comment: String?`

A comment related to the media file.

`var contentCreationDate: Date?`

The creation date of an edited or optimized version of the song or composition.

`var contentModificationDate: Date?`

The date on which the contents of the file was last modified.

`var contentSources: [String]?`

An array of sources from which the media was obtained.

`var copyright: String?`

The copyright date of the content.

`var downloadedDate: Date?`

The most recent date on which the file was downloaded or received.

`var editors: [String]?`

A list of editors who have worked on the file.

`var lastUsedDate: Date?`

The date on which the file was last used.

`var participants: [String]?`

A list of people who are visible in an image or movie or written about in a document.

`var projects: [String]?`

A list of projects of which this file is a part.

`var addedDate: Date?`

The date on which the item was moved into its current location.

`var codecs: [String]?`

The codecs used to encode/decode the media.

`var contactKeywords: [String]?`

A list of contacts who are associated with the content in some way, not including the author.

`var deliveryType: NSNumber?`

The delivery type of the file.

`var duration: NSNumber?`

The duration (if appropriate) of the content of the file, in seconds.

`var mediaTypes: [String]?`

The media types present in the content.

`var organizations: [String]?`

A list of companies or organizations that created the content.

`var streamable: NSNumber?`

A value that indicates if the content is prepared for streaming.

`var totalBitRate: NSNumber?`

The total bit rate of the media, combining audio and video.

`var audioBitRate: NSNumber?`

The audio bit rate of the media.

`var version: String?`

A version string associated with the file.

`var videoBitRate: NSNumber?`

The video bit rate of the media.

`var contributors: [String]?`

A list of people, organizations, or services that made contributions to the media content.

`var languages: [String]?`

A list of the included languages for the intellectual content of the media.

`var publishers: [String]?`

A list of people, organizations, services, or other entities responsible for making the media available.

`var rights: String?`

A link to information about the rights held in and over the media.

`var role: String?`

Indicates the role of the content creator.

`var contentRating: NSNumber?`

A value that indicates if the media contains explicit content.

`var coverage: [String]?`

A list of descriptors that specify the extent or scope of the media.

`var director: String?`

The name of the director of the media (for example, a movie director).

`var genre: String?`

The genre of the media.

`var information: String?`

Information about the media.

`var local: NSNumber?`

A value that indicates if the media is local.

`var originalFormat: String?`

The original format of the media.

`var originalSource: String?`

The original source of the media.

`var performers: [String]?`

A list of performers in the media.

`var playCount: NSNumber?`

A user-supplied play count for the media.

`var producer: String?`

The producer of the content.

`var rating: NSNumber?`

The user-supplied rating of the media.

`var ratingDescription: String?`

A description of the rating.

`var url: URL?`

The URL associated with the media.

Describing music

`var album: String?`

The title for a collection of audio media.

`var artist: String?`

The artist associated with the media.

`var audioChannelCount: NSNumber?`

The number of channels in the audio data that the file contains.

`var audioEncodingApplication: String?`

The name of the application that encoded the data the audio file contains.

`var audioSampleRate: NSNumber?`

The sample rate of the audio data the file contains, as a float value representing Hz (audio frames per second), such as 44100.0 or 22254.54.

`var audioTrackNumber: NSNumber?`

The track number of a song or audio composition when part of an album.

`var composer: String?`

The composer of the song or audio composition that the audio file contains.

`var keySignature: String?`

The musical key of the song or audio composition that the file contains, such as C, Dm, or F#m.

`var lyricist: String?`

The lyricist or text writer for the song or audio composition that the file contains.

`var musicalGenre: String?`

The musical genre of the song or audio composition that the file contains, such as jazz, pop, rock, or classical.

`var recordingDate: Date?`

The recording date of the song or composition.

`var tempo: NSNumber?`

The tempo of the music that the audio file contains, in beats per minute.

`var timeSignature: String?`

The time signature of the musical composition that the audio or MIDI file contains, in a string, such as "4/4" or "7/8".

`var generalMIDISequence: NSNumber?`

A value that indicates whether the MIDI sequence the file contains is set up for use with a general MIDI device.

`var musicalInstrumentCategory: String?`

The category of the instrument associated with the audio file.

`var musicalInstrumentName: String?`

The name of an instrument within the context of an instrument category.

Describing images

`var isoSpeed: NSNumber?`

The ISO speed setting at the time the camera captured the image.

`var acquisitionMake: String?`

The manufacturer of the device that captured the image.

`var acquisitionModel: String?`

The model of the device that captured the image.

`var aperture: NSNumber?`

The size of the lens aperture at the time the camera captured the image, as a log-scale APEX value.

`var bitsPerSample: NSNumber?`

The number of bits per sample.

`var cameraOwner: String?`

The owner of the camera that captured the image.

`var colorSpace: String?`

The color space model the image uses, such as RGB, CMYK, YUV, or YCbCr.

`var flashOn: NSNumber?`

A value that indicates if the camera used a flash to capture the image.

`var focalLength: NSNumber?`

The actual focal length of the lens, in millimeters.

`var focalLength35mm: NSNumber?`

A value that indicates if the focal length is 35mm.

`var layerNames: [String]?`

An array that contains the names of the various layers in the file.

`var lensModel: String?`

The model of the lens that captured the image.

`var orientation: NSNumber?`

The orientation of the data.

`var pixelCount: NSNumber?`

The total number of pixels in the image.

`var pixelHeight: NSNumber?`

The height of the item, such as image or video frame height, in pixels.

`var pixelWidth: NSNumber?`

The width of the item, such as image or video frame width, in pixels.

`var whiteBalance: NSNumber?`

The white balance setting when the camera captured the image.

`var exifgpsVersion: String?`

The version of GPS Info IFD header that was used to generate the metadata for the image.

`var exifVersion: String?`

The version of the EXIF header that was used to generate the metadata for the image.

`var exposureMode: NSNumber?`

The mode the camera used for the exposure of the image.

`var exposureProgram: String?`

The class of the program the camera used to set exposure when capturing the image.

`var exposureTime: NSNumber?`

The time that the lens was open during exposure, in seconds.

`var exposureTimeString: String?`

The time that the lens was open during exposure, in a string, such as "1/250 seconds".

`var fNumber: NSNumber?`

The focal length of the lens, divided by the diameter of the aperture when the camera captured the image.

`var hasAlphaChannel: NSNumber?`

Indicates if the image file has an alpha channel.

`var maxAperture: NSNumber?`

The smallest F number of the lens.

`var meteringMode: String?`

The metering mode.

`var profileName: String?`

The name of the color profile the camera used for the image.

`var redEyeOn: NSNumber?`

A value that indicates if the camera used red-eye reduction when capturing the image.

`var resolutionHeightDPI: NSNumber?`

The resolution height of the image, in DPI.

`var resolutionWidthDPI: NSNumber?`

The resolution width of the image, in DPI.

Describing messages

 Common Mailbox Identifiers

Constants that describe common mailbox names.

`var htmlContentData: Data?`

The HTML content of the document encoded as an NSData object representing a UTF-8 encoded string.

`var accountHandles: [String]?`

An array of the canonical handles for the account with which the message is associated.

`var accountIdentifier: String?`

The unique identifier for the account with which the message is associated, if any.

`var additionalRecipients: [CSPerson]?`

An array of CSPerson objects representing the content of the Cc: field in an email message.

`var authorAddresses: [String]?`

An array of addresses associated with the author of the message.

`var authorEmailAddresses: [String]?`

An array of email addresses associated with the author of the message.

`var authorNames: [String]?`

An array of names representing the authors who have worked on the message.

`var authors: [CSPerson]?`

An array of CSPerson objects representing the content of the From: field in an item.

`var emailAddresses: [String]?`

An array of email addresses associated with the message.

`var emailHeaders: [String : [Any]]?`

A dictionary that contains all the headers of the message.

`var hiddenAdditionalRecipients: [CSPerson]?`

An array of CSPerson objects representing the content of the Bcc: field in an email message.

`var instantMessageAddresses: [String]?`

An array of instant message addresses for the message.

`var likelyJunk: NSNumber`

A value that indicates if the message is likely to be considered junk.

`var mailboxIdentifiers: [String]?`

An array of mailbox identifiers associated with the message.

`var phoneNumbers: [String]?`

An array of phone numbers associated with the message.

`var primaryRecipients: [CSPerson]?`

An array of CSPerson objects representing the content of the To: field in an email message.

`var recipientAddresses: [String]?`

An array of addresses associated with the recipients of the message.

`var recipientEmailAddresses: [String]?`

An array of email addresses associated with the recipient.

`var recipientNames: [String]?`

An array of names representing the recipients of this message.

`var textContent: String?`

The textual content of the message.

Describing containment

`var containerDisplayName: String?`

A localized string that specifies the name of a container to which the item belongs, suitable to display in the user interface.

`var containerIdentifier: String?`

The identifier of the container to which the item belongs.

`var containerOrder: NSNumber?`

The order of the item within the container.

`var containerTitle: String?`

The title of the container to which the item belongs.

Supporting actions

`var actionIdentifiers: [String]`

The identifiers that specify custom actions the app supports for the item.

`var supportsNavigation: NSNumber?`

A value that indicates whether the item contains information sufficient to provide navigation to the location it represents.

`var supportsPhoneCall: NSNumber?`

A value that indicates whether the item contains information sufficient to allow a phone call to a number associated with the item.

`var sharedItemContentType: UTType?`

The file type of the item to enable the user to share items from Spotlight.

`let CSActionIdentifier: String`

A key that specifies the action's identifier in a user activity.

Providing item representations

`var providerDataTypeIdentifiers: [String]?`

An array of identifiers that corresponds to data representations the delegate provides.

`var providerFileTypeIdentifiers: [String]?`

An array of identifiers that corresponds to file representations the delegate provides.

`var providerInPlaceFileTypeIdentifiers: [String]?`

An array of identifiers that corresponds to in-place file representations the delegate provides.

Deprecated

~~`init(itemContentType: String)`~~

Creates an attribute set for the specified content type.

Deprecated

Instance Methods

`func associateAppEntity<Entity>(Entity, priority: Int)`

Associates an app entity with this searchable item.

`func move(from: CSSearchableItemAttributeSet)`

Relationships

Inherits From

NSObject

Conforms To

CVarArg

CustomDebugStringConvertible

CustomStringConvertible

Equatable

Hashable

NSCoding
NSCopying
NSObjectProtocol
NSSecureCoding

See Also

Searchable items

`class` `CSSearchableItem`

The details of your app-specific content that someone might search for on their devices.

`class` `CSCustomAttributeKey`

A key associated with a custom attribute for a searchable item.

`class` `CSLocalizedString`

An object that displays localized text in search results related to your app.

`class` `CSPerson`

An object that represents a person in the context of search results.